グラフィクスとCAD　83－7
（1996．12．20）

多重解像度モデルを用いた仮想空間における3次元物体の高速表現

王 生進　　佐藤 誠

東京工業大学 精密工学研究所
〒226 横浜市緑区長津田町 4259

あらまし バーチャルリアリティ（VR）視覚空間内で3次元物体の表現に対して実時間で物体を描画することが重要な課題である．本報告は3次元物体の多重解像度モデルを構築するための階層的八分木キューブ分割法を提案する。視覚空間内でのパラメータとする解像度、視点距離を設定し、形状簡略化に関する階層化八分木キューブ構造を得る。階層化八分木キューブ構造により物体形状を階層的に簡略化する。はじめに、レベル1の解像度に対応するキューブサイズを決め、キューブ分割法による離散点群データから、レベル1解像度のモデルを構成する。次に、八分木キューブ構造により形状を簡略化し、各レベル解像度のモデルを構築する。さらに、3次元物体の局所構造から大域構造までの形状を表す多重解像度モデルを構築し、本アルゴリズムの有効性を確認する VR 空間で3次元物体を高速に表示する実験を行い、VR 空間での高速表示への多重解像度モデルの応用を実現し、VR 空間で大量かつ高速にの物体を表現するで有効であることを明らかにした。

キーワード：3次元形状表現, 階層的キューブ分割, 形状簡略化, 多重解像度表現, 八分木キューブ構造, 3次元物体高速表現.

High-Speed Display of 3-D Objects in Virtual Reality Systems Based on Multiresolution Models

Shengjin WANG and Makoto SATO

Precision and Intelligence Laboratory, Tokyo Institute of Technology.
4259 *Nagatsuta – cho, Midori – ku, Yokohama – shi,* 226 *Japan,*
E-mail: wsj@pi.titech.ac.jp

Abstract High-speed display of a 3-D object in virtual reality environments is one of the currently important subjects. Shape simplification is considered an efficient method. We present a new approach called hierarchical cube-based segmentation to shape simplification of arbitrary 3-D objects for multiresolution model construction. The relationship among shape simplification, resolution and visual distance are derived. The idea of the algorithm is to construct the shape of a 3-D object by cube-based segmentation and further to adjust the cube sizes hierarchically using octree hierarchical structure for multiresolution description. First level model is constructed from scattered range data with the first level cube size. Multiresolution models are then generated by re-sampling the polygonal vertices of each former level patches. The results show that the algorithm can simply and quickly generate multiresolution models from multiple view range images and rendering in an interactive display for a real-time viewing with a designated image resolution during the whole visualization.

Key words: *3-D shape representation, Hierarchical cube-based segmentation, Shape simplification, Multiresolution description, Octree cube structure, High-speed display.*

1 Introduction

Real-time viewing of a 3-D object in virtual reality environments is one of the currently important subjects. The interactive computer graphics systems for visualization of realistic looking, 3-D objects are useful for evaluation, design and training in virtual environments, such as those found in mechanical CAD, scenery simulation, and virtual reality.

Modern graphics workstation allow the display of thousands of shaded or textured polygons at interactive rates. However, many applications contain graphical models with geometric complexity still greatly exceeding the capabilities of typical graphics hardware. This problem is particularly prebvalent in applications dealing with large polygonal surface models, such as 3-D object modeling and visual simulation.

In order to accommodate complex surface models while still maintaining real-time display rates, methods for approximating the polygonal surfaces and using multiresolution models have been proposed[1]. Shape simplification can be used to generate multilevel models at varying levels of detail, and techniques are employed by the display systems to select and render the appropriate level of detail model.

Multiresolution algorithms have been mainly described as two approaches. One of them is space subdivision by octree-related data structures on ray tracing[2]. Although they are effective, they have no concerned with the characters of human visual sensation of near and far viewpoint. Another approach is described as the techniques for choosing a level of detail at which to render each visible object which is called detail elision or shape simplification[3]. However, the objects they concerned with are represented only by the sets of plane polyhedrons. The shape simplification of arbitrary shapes of objects is remained as a significant subject. Recently, excellent results are reported on planar polyhedrons made by computer graphics, such as cylinder, cube and sphere. However, the difficulty of dealing with arbitrary shapes is evidenced by the extensive recent work on the topic. The problems of simplification and detail level control have been addressed by Turk[4], Schroeder et al.[5], Hoppe et al.[6], Rossignac and Borrel[7], Varsney[8], and Eck[9]. However, the relationship between models and resolution are not given out and the complex algorithms are take a lot of times.

In this report we present a new approach called hierarchical cube-based segmentation to shape simplification of arbitrary 3-D objects for multiresolution model construction. The relationship among shape simplification, resolution and visual distance are derived. The idea of the algorithm is to construct the shape of a 3-D object by cube-based segmentation and further to adjust the cube sizes hierarchically using octree hierarchical structure for multiresolution description. First level model is constructed from scattered range data with the first level cube size. Multiresolution models are then generated by re-sampling the polygonal vertices of each former level patches. The results show that the algorithm can simply and quickly generate multiresolution models from multiple view range images and rendering in an interactive display for a real-time viewing with a designated image resolution during the whole visualization.

2 Multilevel Representation

2.1 Parameter Definition

For a 3-D object, since it is difficult to generate simplification data according with visual distance in real-time, we quantize the distance previously and keep the simplified data in computer memory previously. Assuming that the objects space is limited, L distant divisions are arranged and L step simplified data are prepared. According to the distance from viewpoint to object, the simplified level is selected and corresponded shape of objects data are rendered. We determine four parameters of multilevel representation in visual space.

· Number of simplified level L

· Threshold of distance from view point for selecting simplified level $D_i (1 \leq i \leq L)$

· Distance of rendering limitation D_{max}

· Resolution d_i in response to D_i

The larger the number of level L is, the more smooth change and nature rendering images can be obtained by way of gradually simplifying of the objects. However, more level can not always save display time practically. The method of balancing image quality and speed is necessary in practical display.

2.2 Image Resolution

Multiresolution description that the objects appear larger to the viewer described more to the image is shown in Fig. 1. Let θ be the visual angle and D_p be the distance between object and viewer. Supposing the visual range at D_p is W, the total pixels of horizontal(or vertical) direction of screen is N, and the number of horizontal pixels which the object projects image on screen are n, We can obtain

$$x = \frac{2n sin\frac{\theta}{2}}{N} D_p, \tag{1}$$

where x is the length of the object along horizontal direction and $sin\frac{\theta}{2} = (\frac{W}{2})/D_p$.

Equation (1) shows the relationship of the length x and the distance D_p under the certain conditions of N and θ, where the pixels of x projects on screen are n.

We now define a parameter n_0. The significance of n_0 is a threshold of image deterioration. That means when an object is represented on screen, the object is not displayed when $n < n_0$. Obviously, if $n = 1$, the image is no deterioration. Supposing that θ and N are constants in a visual environment and the deterioration threshold n_0 is given, we can, from Eq. (1), find that x has in direct ratio with D_p. This property enables us to realize multiresolution description in visual space.

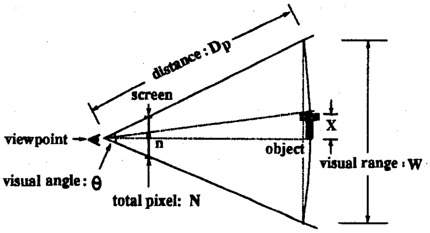


Fig. 1 Multiresolution determination.

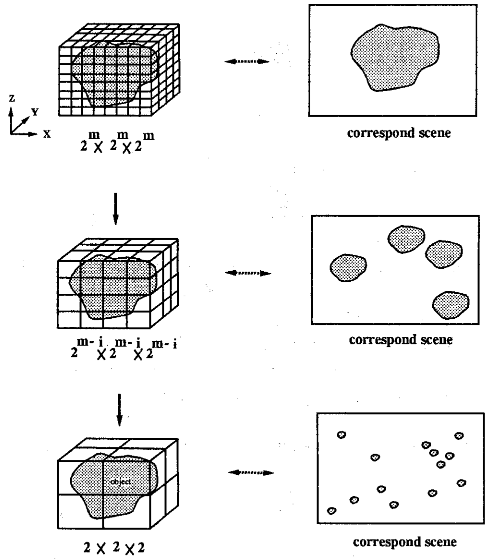


Fig. 2 Hierarchical cube-based segmentation and multiresolution models.

3 Multiresolution Model Construction

3.1 Hierarchical Cube-Based Segmentation

The basic idea of construction of multiresolution models, as illustrated in Fig. 2 is to construct a group of cubes over the determined domain space. The sizes of cubes are hierarchical ones corresponding to image resolutions with octree structures.

Now we derive the cube size of each level from Eq. (1) and regard the size as the resolution of objects. In a practical manner, the variable D_p in Eq. (1) is quantized into L domains as we mentioned above. Therefore, the size of cube can be obtained by:

$$d_i = \frac{2n_0 sin\frac{\theta}{2}}{N} D_i, \qquad i = 1, ..., L \tag{2}$$

where the resolution d_i is the size of cube and D_i is a distance between viewpoint and bound of *ith* domain. The cube size d_1 in first level of view space can be determined by D_1. User can designate D_1 according to the range of first level of view space.

3.2 First Level Model Construction

The shape information of a 3-D object can be obtained from multiple view range images. The data are delivered from an optical measuring device in unorganized forms. We express raw data as $\mathbf{R}_i(x_i, y_i, z_i)|i = 1, ..., M$. M indicates the total number of scattered measurement data. Let $(x_{cen}, y_{cen}, z_{cen})$ be the center of gravity of the object which is described by the scattered range data. Let $(x_{min}, y_{min}, z_{min})$ be the minimum vertex of outline box which encloses the object and $(x_{max}, y_{max}, z_{max})$ be the maximum vertex of the outline box.

Supposing the number of level is L, so the cube size of the *Lth* level is

$$d_L = 2^{L-1} \cdot d_1, \tag{3}$$

where d_1 is the cube size of the first level. Let $(x_{start}, y_{start}, z_{start})$ be the starting point and $(x_{stop}, y_{stop}, z_{stop})$ be the stopping point of cube-based segmentation. They now can be determined as follows:

$$x_{start} = x_{cen} - d_L \cdot int(\frac{x_{cen} - x_{min}}{d_L} + 1), \tag{4}$$

$$y_{start} = y_{cen} - d_L \cdot int(\frac{y_{cen} - y_{min}}{d_L} + 1), \tag{5}$$

$$z_{start} = z_{cen} - d_L \cdot int(\frac{z_{cen} - z_{min}}{d_L} + 1). \tag{6}$$

$$x_{stop} = x_{cen} + d_L \cdot int(\frac{x_{max} - x_{cen}}{d_L} + 1), \quad (7)$$

$$y_{stop} = y_{cen} + d_L \cdot int(\frac{y_{max} - y_{cen}}{d_L} + 1), \quad (8)$$

$$z_{stop} = z_{cen} + d_L \cdot int(\frac{z_{max} - z_{cen}}{d_L} + 1). \quad (9)$$

We start cube-based segmentation along the directions of x, y, z axes from point $(x_{start}, y_{start}, z_{start})$ with the interval d_1. The segmentation will stop when the cubes exceed the bounds of x_{stop}, y_{stop}, z_{stop} along the direction of x, y, z axes[10].

We assume that the cube size of d_1 is fine enough, so the patches inside each cube are approximately planes. Figure 3 shows all patterns of polygonal patches segmented inside a cube. The circle marks indicate the segmented cube apexes and the dot marks are polygonal vertices.

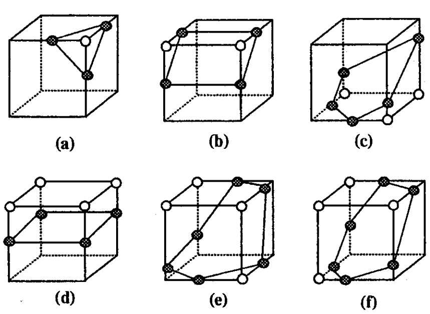


Fig. 3 Patterns of patches inside cubes: (a) triangle, (b) quadrangle,(c) pentagon, (d) quadrangle, (e) hexagon, (f) hexagon.

3.3 Shape Simplification with Octree Cubes

An octree is a solid model that controls a set of cubic regions in a 3-D space shown in Fig. 4. In an octree structure, this set is represented by apexes arranged in a hierarchical structure. A d_i cubic region corresponding to the resolution level i merges eight equal cubic subregions, which correspond to the resolution level $(i-1)$. The octree structure shown in Figure 4 shows that the new apexes of cube d_i are one of the apexes of each cube d_{i-1}, respectively.

Since all polygonal vertices are located on edges of each cube, only three edges of cube d_{i-1}, not twelve edges, appear in the edges of cube d_i. The polygonal vertices of former level patches can be reduced. We now make polygonal vertices re-sampling on the new cube by picking out the vertices only located on the edges of new cube.

Our re-sampling algorithm consists of three steps, as illustrated as follows:

1) First, settling the eight apexes of new cube and searching for all former level patches being inside the new cube. It is easy to determine the cube size of the resolution level d_i with $d_i = 2^{i-1} \cdot d_1$, where $i = 1, ..., L$. From starting point $(x_{min}, y_{min}, z_{min})$, the apexes of new cube can be obtained.

2) Grouping the patches of which their boundaries connect each other, since there are possibly several kinds of patch patterns inside a cube simultaneously. The grouped patches are continuous patches.

3) For the grouped patches, searching for the vertices which locate on the twelve edges of new cube and according to their series order which is shown in Fig. 5(a) an enclosed loop routine of new patch is formed.

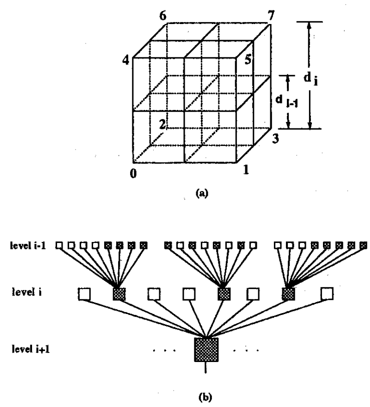


Fig. 4 Octree structure: (a) an octree cube, (b) octree cube structure. indicate the cubes which have patches.

Now we use the obtained new vertices to form a new patch and to construct the new level model. Since there are complex shapes of some objects, the distribution of the new vertices locating on new cube edges has various forms. According to the concept of detail elision, the new patches at present level should be approximately planes. To judge the patch form inside a cube, we use a criterion of six pattern patches. That is, after doing the re-sampling of former level vertices, only the vertices which can form the one of the six pattern patches are remained at present level.

In accordance with this criterion, we form the loop routine of new patch boundary by searching for the new vertices and picking out the vertices which can form one of the six pattern patches. The processing steps is stated as follows:

1) Forming the loop routine of new patch boundary with the obtained vertices.

2) Assuming the order of a vertex in loop routine is i, then the adjacent two vertices are $i-1$ and $i+1$. Checking whether the vertices $i-1$, i, $i+1$ locate on the same cube face simultaneously or not. If they do not locate on the same cube face, the vertex i is remained as a vertex of new patch. If they locate on the same cube face, the vertex i will be eliminated from the loop routine, where $i = 1, ..., M$. M is the total number of the obtained vertices.

Assuming m vertices are remained, if $m = 0$, there is no patch is formed inside the cube; if $m \neq 0$, one of the six pattern patches will be formed and the number of m is corresponding to the patterns. For example, $m = 3$ is corresponding to a triangle, and $m = 4$ is corresponding to a quadrangle.

Figure 5 gives the two examples of patch merging and new patch generation. Figure 5(a) shows four grouped quadrangles inside a new cube. Figure 5(c) shows two quadrangles and two pentagons inside a new cube. The loop routine of new patch boundary is formed as shown in Fig. 5(a) and 5(c), respectively. Figure 5(b) and 5(d) shows the generated new patches.

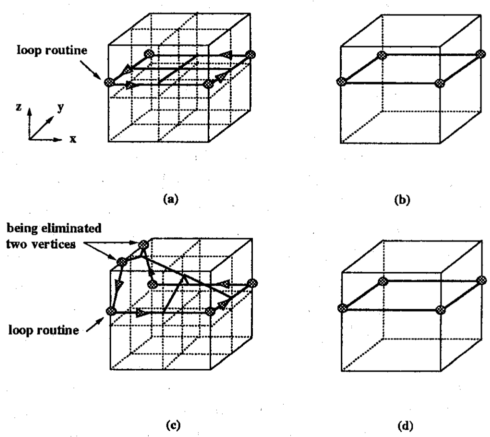


Fig. 5 The patch merging and new patch generation: (a) merging four quadrangles inside new cube, (b) the generated new quadrangular patch, (c) merging two quadrangles and two pentagons inside new cube in a case of two vertices locating on a cube edge, (d) the generated new quadrangular patch.

3.4 Border Patch Generation

The principal aim of shape simplification for multiresolution representation is detail elision. That means some detail volume parts of an object in former resolution level will be eliminated in the present resolution level. If there are two vertices locating on a new cube edge simultaneously, as we discussed above, such two kind of vertices will be abandoned. It means this detail part of the object is eliminated. As a result, a hole will occur certainly on the object surface.

To mend the hole, a special patch, called border patch, is needed. When a border patch is generated, the hole can be mended with the border patch.

In order to generate a border patch, the patches which interrelate the hole should be found. An interrelated patch has a feature that its boundary, which forms the hole, does not connect with other patches, or can say disconnecting. According to this feature, we can find all patches round the hole and generate a border patch with the interrelated vertices of the patches.

Since such two vertices which locate on a cube edge simultaneously are abandoned within each cube face, a hole occurs certainly on the cube face. Therefore, the generation of a border patch is also done on each corresponding cube face. Figure 6 shows the algorithm of a border patch generation, as illustrated as follows:

1) Figure 6(a) shows the cube-based segmentation with the cube size of level $i-1$ and Fig. 6(b) is the object shape represented by the generated patches in level $i-1$. Figure 6(c) shows the hierarchical cube-based segmentation with the cube size of level i and Fig. 6(d) is the object shape represented by the generated patches in level i. Here a hole occurs on the object surface.

Now for each cube face, searching for the patches whose one boundary locates on the cube face. For each found patch, checking whether there is a connected patch with it or not. If the connected patch exists, this patch is not considered. Finally, the remained patches are all the disconnected patches. If there is no remained patch, then going to next cube face.

2) For the obtained disconnected patches on the cube face, finding all the vertices of the disconnected patches and forming an enclosed loop routine of a new patch boundary on the cube face.

3) Using the obtained vertices in the loop routine to generate a new patch. The new patch boundary surrounds the hole and the new patch is taken as a border patch to mend the hole. The shading facets shown in Fig. 6(e) indicate the generated border patches. Figure 6(f) is the object shape in resolution level i after mending the hole.

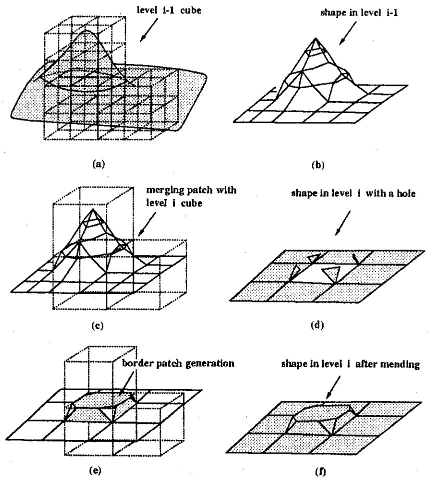


Fig. 6 The illustration of hole happening and border patch generation on object surface: (a) cube-based segmentation in resolution level $i-1$, (b) the shape in level $i-1$, (c) hierarchical cube-based segmentation in resolution level i, (d) a hole occurs on object surface in level i, (e) border patch generation, (f) the shape after mending the hole with the border patches.

4 Experimental Results

4.1 Multiresolution Models

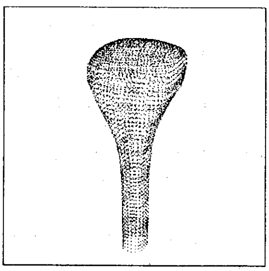


Fig. 7 The raw range data of a car's knob model.

We have experimented with the multiresolution description method on data sets obtained from a multiple view range images. The algorithm's parameters are listed in the Table 1 for the car's knob model. The figures are shown from Fig. 7 to Fig. 8. Figure 7 shows the three-dimensional display of a car model's range data of which the total number of raw data is 68844.

Figure	D_i (m)	d_i (cm)	patch num.	compr. ratio	time (sec.)
8(a) L1	2	0.6	1612	2.34%	1770
8(d) L2	4	1.2	372	0.54%	447
8(g) L3	8	2.4	102	0.15%	118
8(j) L4	16	4.8	28	0.04%	62
8(m) L5	32	9.6	12	0.02%	48

Table 1: Parameters of multiresolution description with $\theta = 60°$, $N = 1280$ pixels, $n_0 = 4$ pixels, and the results of multiresolution description with distances, patch numbers, compress ratio, and processing time.

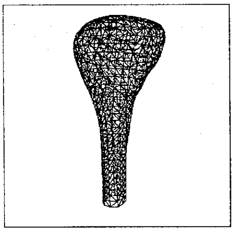
Table 1 summarize the results of multiresolution descriptions applied to the models. Each line of the table gives the number of parameter D_i and d_i, triangular patches, compression ratio between original data and re-sampling points of multiresolution models.

Figure 9 shows the knob number in each volume space under the condition of having approximate total patch number. That is 1, 4, 16, 56, 132 in each volume space, respectively. Figure 10 shows the distribution of knob patch number in each volume space with a certain knob number and different multiresolution models.

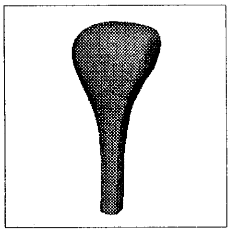
4.2 Interactive Display

Figure 8(b) is the reconstructed models in the first level, which have 1612 triangular patches. Figure 8(d) and Fig. 8(e) are the reconstructed models in the second level, which have 372 triangular patches. Figure 8(f) is the corresponding display of Fig. 8(e) with two times visual distance of Fig. 8(c). Figure 8(g) and Fig. 8(h) are the reconstructed models in the third level, which have 50 triangular patches. Figure 8(i) is the corresponding display of Fig. 8(h) with four times visual distance of Fig. 8(c).

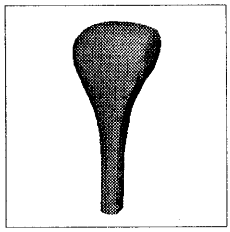
Figure 11 shows the display time of multiresolution representation in each volume space with a certain knob number and different multiresolution models. In Fig. 11, it can be observed that the multiresolution display with five knob models is done with a linear time function, and becoming more and more nonlinear time function when the number of knob model types is reduced.



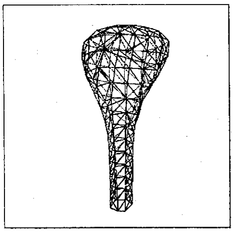
(a) Wireframe of 1st level model.



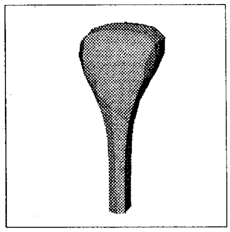
(b) Shading representation.



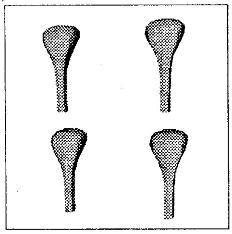
(c) Display in 1st level space.



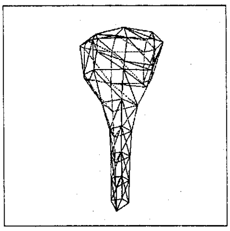
(d) Wireframe of 2nd level model.



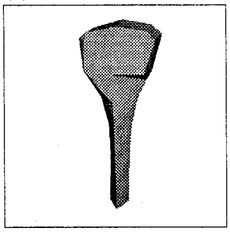
(e) Shading representation.



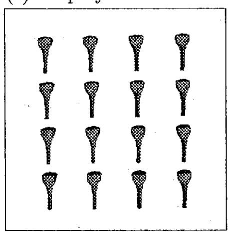
(f) Display in 2nd level space.



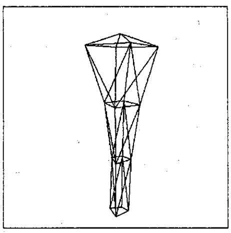
(g) Wireframe of 3rd level model.



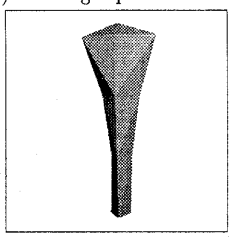
(h) Shading representation.



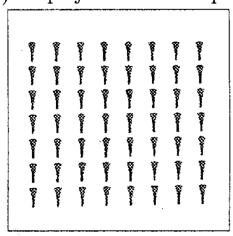
(i) Display in 3rd level space.



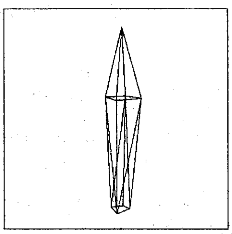
(j) Wireframe of 4th level model.



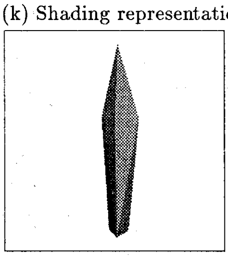
(k) Shading representation.



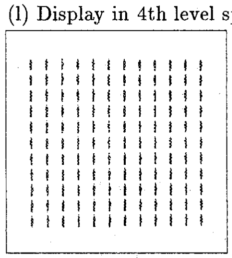
(l) Display in 4th level space.



(m) Wireframe of 5th level model.



(n) Shading representation.



(o) Display in 5th level space.

Fig. 8 The experimental results of a car's knob model. The display of each level space is under the appreciate total patch number.

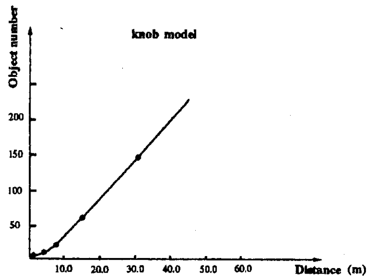


Fig. 9 The knob number in each volume space under the condition of having approximate total patch number.

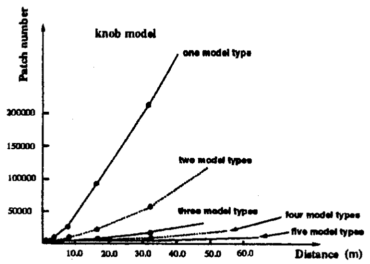


Fig. 10 The distribution of knob patch number in each volume space with a certain knob number and different multiresolution models.

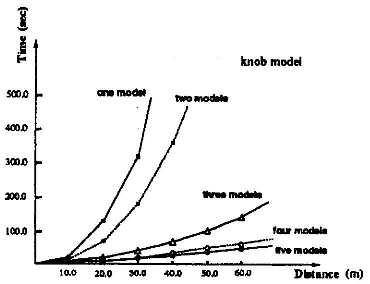


Fig. 11 The display time of multiresolution representation in each volume space with a certain knob number and different multiresolution models.

5 Conclusions

We have presented a new approach to the shape simplification of arbitrary shape 3-D objects for generating multiresolution models for high-speed display. The algorithm can simply, quickly and automatically generate the multiresolution models in corresponding to a certain resolution by only changing the cube sizes. Our algorithm provides guaranteed visual resolution bounds, and corresponds to the concept of fine to coarse with detail elision. The algorithm can be used as a tool of 3-D shape construction in visual environment of virtual reality system.

References

[1] Heckbert, P. S. and Garland, M. : "Multiresolution Modeling for Fast Rendering", In Proceedings of Graphics Interaface'94, 1994, pp. 1-8.

[2] Yamaguchi K., Kunii T. and Fujimura K.: "Octree-Related Data Structure and Algorithms ", IEEE Computer Graphics & Applications, vol.4, no.1, pp.53-59(1984).

[3] Kitajima K. and Yusa Y.: "Study on Saving Display Time for a Real-Time Visual Simulator Based on the Methods of Grouping and Multi-Representation of Shapes", IEICE Trans., vol.J77-D-II, no.2, pp.311-320(1994).

[4] Greg Turk: "Re-tiling polygonal surface", Computer Graphics (SIGGRAPH'92 Proceedings), pp. 55-64 (July 1992)

[5] William Schroeder, Jonathan Zarge, and William Lorensen: "Decimation of triangle meshes", Computer Graphics (SIGGRAPH'92 Proceedings), pp. 65-70 (July 1992)

[6] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle: "Mesh optimization", Computer Graphics (SIGGRAPH'93 Proceedings), pp. 19-26 (August 1993)

[7] J. Rossignac and P. Borrel: "Multi-resolution 3D approximations for rendering", In B. Falcidieno and T.L. Kunii, editors, Modeling in Computer Graphics, pp. 455-465. Spring-Verlag (June-July 1993)

[8] Amitabh Varshney: "Hierarchical Geometric Approximations", PhD thesis, Department of Computer Science, University of North Carolina at Chapel Hill (1994)

[9] Matthias Eck, Tony DeRose, Tom Duchamp, Hugues Hoppe, Michael Lounsbery, and Werner Stuetzle: "Multiresolution analysis of arbitrary meshes", Computer Graphics (SIGGRAPH'95 Proceedings), pp. 173-182 (August 1995)

[10] S. Wang, Y. Cai, and M. Sato: "Scattered Range Data Re-sampling for High Speed Display of 3-D Objects," The Journal of The Institute of Television Engineers of Japan (Journal of ITEJ.), vol. 49, no. 10, pp. 1246-1251, 1995.