VARIABLE RESOLUTION LEVEL-OF-DETAIL OF MULTIRESOLUTION ORDERED MESHES

J. Ribelles, A. López, O. Belmonte, I. Remolar, M. Chover [1]

Departamento de Informática
Universitat Jaume I
12080 Castellón
Spain
{ribelles, lopeza, belfern, remolar, chover}@uji.es

ABSTRACT

Variable resolution level-of-detail is an important application of multiresolution modeling, since te use of different resolutions across the surface allows interactive visualization of highly detailed objects. Multiresolution Ordered Meshes (MOM) was first introduced as a model that achieves the efficient management of an ample range of uniform resolution levels-of-detail and presents reduced storage space requirements. In this paper, we introduce an algorithm capable of retrieving variable resolution levels-of-detail from a multiresolution MOM representation without having to reorganize data in new structures or store new information. The proposed algorithm starts from the level of highest detail and simplifies it adaptively to reach the required resolution in each area of the surface. Experiments with data sets of varying complexity demonstrate that the new algorithm obtains variable resolution levels of detail while retaining the advantages of MOM.

Keywords: multiresolution modeling, level of detail, dynamic simplification, interactive visualization.

1 INTRODUCTION

Multiresolution modeling of polygonal surface meshes has been presented as a solution for the interactive visualization of objects formed by hundreds of thousands of polygons [Garla99b]. A multiresolution model represents an object by means of multiple resolutions, of varying complexities, called levels-of-detail (lods) (see Fig. 1), so that the application can visualize the object using the most suitable level-of-detail (lod) and thus avoid, for instance, wasting time on visualizing imperceptible details [Heckb94]. These models provide two possible solutions: levels-of-detail of uniform resolution [Hoppe96, Popov97, Ciamp97, Guezi98, Ribel98, Abadj99, Guezi99] and levels-of-detail of variable resolution [Flori97a, Xia97, Hoppe97, Luebk97, Guezi99, Elsan99]. For example, the visualization of a table in a room is performed using uniform resolution lods, while the terrain for a flight simulator must be represented with lods where resolution varies across the surface depending on visibility and distance in relation to the pilot.

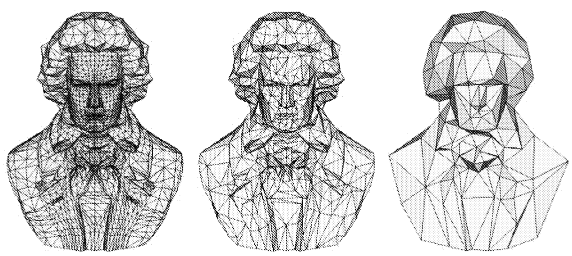


Figure 1: Three levels of detail of the *Ludwig* model. Left, 4998 faces; center, 1250 faces; right, 250 faces.

In this paper an algorithm capable of retrieving variable resolution lods from a *Multiresolution Ordered Meshes* (MOM)[Ribel98] representation

[1] This research work has been supported by grants TIC1999-0510-C02-02 and TIC2000-1131 (CICYT, Ministerio de Educación y Ciencia)

is presented. MOM is a multiresolution model that achieves the efficient management of an ample range of uniform resolution lods and presents reduced storage space requirements. This model is characterized by its generality: it can represent manifold and non-manifold surfaces and any simplification method can be used in the construction process. Consequently, it can be used in a wider range of applications. The goal is to fit MOM with the possibility of retrieving variable resolution lods while retaining these properties.

Algorithms designed to retrieve variable resolution lods traverse the data structure to reach the required lod either by starting from the lod of lowest resolution and refining [Hoppe97, Luebk97, Xia97, Puppo98], or by starting from the lod of highest resolution and simplifying [Linds96]. The algorithm proposed here is based on the second strategy. In both cases, simplification or refinement is made at run time, based on criteria such as view location, illumination, speed and motion, or the interactive selection by a user of the area to be represented in high detail (see Fig. 2).

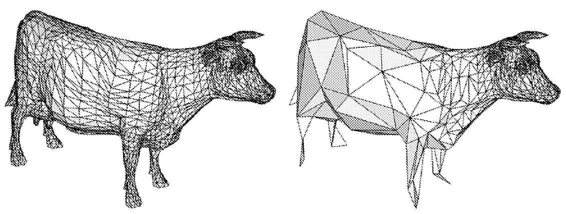


Figure 2: Example of variable resolution lod. Left, the original model (5804 faces); right, the model is represented with highest resolution in the head and resolution decreases towards the tail (2074 faces).

Generally, multiresolution models that allow the retrieval of variable resolution lods rely on hierarchical data structures, usually based on trees [Xia96, Hoppe97, Luebk97] or directed acyclic graphs [Flori98], which typically have a higher storage cost than incremental representations. For instance, Hoppe changed the incremental representation [Hoppe96] of Progressive Meshes for a hierarchical representation [Hoppe97] in order to retrieve variable resolution lods in real time. An important advantage of the algorithm proposed in this paper is the use of the original MOM incremental representation, without having to reorganize data in new structures or store new information, thus maintaining a reduced storage cost. Consequently, the MOM construction process creates an incremental representation from which both uniform and variable resolution lods can be retrieved. There is no need for an additional process which could increment the cost and complexity of the construction process.

When extracting variable resolution lods, if the possible dependencies between triangulations are not taken into account foldovers can appear in the mesh. Xia et al. [Xia97] introduced dependencies lists in order to prevent foldovers at run time. Although dependencies lists were improved later by Hoppe [Hoppe97], they are expensive to store and to test at run time due to the memory overhead and several non-local accesses [Elsan99]. Another advantage of our algorithm is that it does not need to store new information on dependencies between updates, given that MOM encodes them implicitly.

Notation. The geometry of a triangulated model, M, is denoted as a tuple $\{V, F\}$, where V is a set of N positions $v_i = (x_i, y_i, z_i) \in \mathbb{R}$, $1 \leq i \leq N$, and F is a set of triples $\{j, k, l\}$, $j, k, l \in V$, specifying positions of triangles faces.

2 RELATED WORK

2.1 Variable Resolution Level of Detail

Several authors [Hoppe98b, Garla99b] classify multiresolution models according to the surfaces to which they are applied: models for *height fields* and models for *arbitrary surfaces*. Height fields are among the simplest types of surfaces. Many methods [Flori96, Heckb97] have been developed for this special case of surfaces, mainly due to their application to geographic information systems and flight simulators [Puppo97]. Some representative references of the work on height fields are [Scarl90, Berg95, Linds96, Puppo96, Cigno97, Hoppe98b].

However, our interest focuses on multiresolution modeling of arbitrary triangle surfaces. Floriani et al. [Flori97a, Puppo98] presented a general framework called *Multi-Triangulation* where any multiresolution representation is described. In [Flori97b, Flori98] the same authors describe an efficient implementation of a Multi-Triangulation and several alternative data structures for its codification. A particular representation which has received special attention are models based on vertex hierarchies, such as the models proposed by Xia et al. [Xia97], Hoppe [Hoppe97], Luebke and Erikson [Luebk97], Guéziec et al. [Guezi99] and El-Sana and Varshney [Elsan99]. They are mainly motivated by the simplification methods used to construct the multiresolution representation, which are based on edge collapse [Garla99a] or vertex clustering [Rossi96].

2.2 Review of Multiresolution Ordered Meshes

Multitresolution Ordered Meshes was presented with the idea of improving the interactive visualization of complex polygonal surfaces [Ribel98, Ribel99]. To build M^r, the MOM representation of an arbitrary mesh M, with n levels of detail, we apply $n-1$ iterations of a simplification method:

$$(M = M_0) \stackrel{S_0}{\to} M_1 \stackrel{S_1}{\to} M_2 \stackrel{S_2}{\to} \dots \stackrel{S_{n-2}}{\to} M_{n-1} \quad (1)$$

Therefore, the tuple $(\{S_0, S_1, \dots, S_{n-2}\}, M_{n-1})$ forms a MOM representation of M. Each simplification S_i, $0 \le i < n-1$, may be represented by the tuple $S_i = \{\mathcal{V}_i, \mathcal{F}_i, \mathcal{V}'_i, \mathcal{F}'_i\}$ where $\mathcal{V}_i$ and $\mathcal{F}_i$ are the sets of vertices and faces which are eliminated from M_i, and $\mathcal{V}'_i$ and $\mathcal{F}'_i$ are the sets of vertices and faces which are added to M_i to generate, finally, M_{i+1}.

3 DISPLAYING A VARIABLE RESOLUTION LOD

Let us explain the meaning of variable resolution lod with an example. In Fig. 3, a sphere interactively located by a user selects the area to be represented in high detail. This area is drawn using faces of the highest resolution lod (light gray). The rest of the object is drawn with the coarsest resolution possible while maintaining the connectivity of the mesh. Thus, two sets of faces can be observed: a set of faces of the lowest resolution lod (dark gray) and another set of faces (medium gray) which performs the transition between highest and lowest resolution, ensuring connectivity.

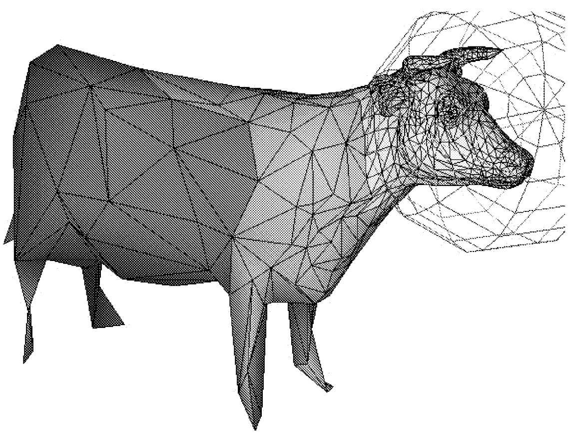


Figure 3: An example of variable resolution lod.

A criterion or set of criteria is needed to retrieve a variable resolution lod. The criterion decides which part of the object is simplified and which part is refined. The user could decide interactively which regions should be displayed with higher or lower resolution, or the application also could take the decision. Several authors [Hoppe97, Xia97, Luebk97, Elsan99] have defined criteria about the automation of the selection of the areas to be represented in higher detail, which are:

- Local illumination, increasing detail in a direction perpendicular to, and proportional to, the illumination gradient across the surface.
- View frustum, increasing detail in the regions inside the view frustum.
- Silhouette boundaries, increasing detail in the regions where there are edges for which one of the adjacent faces is visible and the other is invisible.
- Orientation surface, increasing detail to the regions oriented near the viewer.
- Screen-space projections, increasing or decreasing detail in the region depending on the length of its screen-space projection.

The solutions proposed in the literature can easily be added to our scheme, given that our algorithm is independent of the criteria to be used.

3.1 Basic Strategy

The algorithm starts from the highest resolution lod, traverses the sequence of simplifications $\{S_0, \dots, S_{n-1}\}$ and decides whether each simplification must be performed or not by means of a test function, which includes the evaluation of the criterion defined by the application.

Bearing in mind the results of the test function, if simplification S_i does not have to be performed, each face $f' \in \mathcal{F}'_i$ and each vertex $v' \in \mathcal{V}'_i$ can not belong to the required lod. This generates a dependency on later simplifications, $\{S_{i+1}, \dots S_{n-2}\}$. For example, Fig. 4 represents a simplification consisting of the elimination of vertex v and the triangulation of the hole with faces *fm, fn* and *fp*. None of these faces, *fm, fn* and *fp*, can be eliminated in later simplifications if vertex v is not eliminated previously. In order to check these dependencies, we define R as the set of faces which can not be eliminated. That is, $R \leftarrow R \cup \mathcal{F}'_i$ for each simplification S_i not performed (where initially $R = \emptyset$). Therefore, regardless of the result of the evaluation of the criterion defined by the application, a simplification S_i will not be performed if there is at least one face f such that $f \in \mathcal{F}_i$ and $f \in R$.

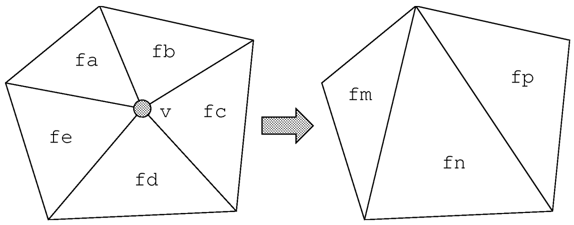


Figure 4: Faces *fm, fn y fp* can not appear in the solution if vertex v has not been previously eliminated ($\mathcal{V}_i = \{v\}$, $\mathcal{F}_i = \{fa, fb, fc, fd, fe\}$, $\mathcal{V}'_i = \{\}, \mathcal{F}'_i = \{fm, fn, fp\}$).

3.2 Algorithm

The algorithm for visualizing a variable resolution lod is shown in Fig. 5. In order to encode set R we define a list of marks, the *flag* array in the algorithm, that indicate which faces belong to R. Initially, all the faces are capable of belonging to the solution, and they are marked as *DRAWABLE*. Then, two stages can be observed in the proposed algorithm.

First Stage

The first stage consists of traversing the sequence of $n-1$ simplifications which, starting from M_0, generate the remaining $n-1$ lods stored in the multiresolution representation.

For each simplification S_i a *test()* function is evaluated. This function includes the evaluation of the criterion defined by the application. However, it is not necessary to evaluate the criterion for all the simplifications. Given that the algorithm starts from the lod with highest resolution, M_0, it is only necessary to evaluate the criterion for those simplifications containing any face of F_0. That is, the criterion will be evaluated for each simplification S_i if and only if there is at least one face f such that $f \in \mathcal{F}_i$ and $f \in F_0$.

To sum up, the *test*() function returns *true* (preserves detail) in all of the following cases and *false* otherwise:

1. The criterion defined by the application decides to preserve detail and there is at least one face f such that $f \in \mathcal{F}_i$ and $f \in F_0$.

// Initialize the list of marks
flag[0..#faces_in_M^r]= DRAWABLE;

FIRST STAGE
// Traverse the set of simplifications
for each simplification $S_i \in [S_0..S_{n-2}]$ **do**
  if test(S_i)= TRUE **then**
    // Mark the faces not generated
    $\forall f \in \mathcal{F}'_i flag[f]$= NO_DRAWABLE
    // Traverse the set of faces $\mathcal{F}_i$
    for each face $f \in \mathcal{F}_i$ **do**
      if $flag[f]$= DRAWABLE
        then draw(f);
      end if
    end for
  end if
end for

SECOND STAGE
// Traverse the set of faces F_{n-1}
for each face $f \in F_{n-1}$ **do**
  if $flag[f]$= DRAWABLE
    then draw(f);
  end if
end for

Figure 5: Algorithm for visualizing a variable resolution lod.

2. There is at least one face $f \in \mathcal{F}'_i$ such that $flag[f]$= *NO_DRAWABLE*.

Bearing in mind the result of the *test()* function, the actions to perform are:

1. If $test(S_i) = true$, $\forall f \in \mathcal{F}'_i$ we set $flag[f]$ to *NO_DRAWABLE*, and for each face $f \in \mathcal{F}_i$ if $flag[f]$= *DRAWABLE* then f is drawn.
2. If $test(S_i) = false$, the algorithm performs the simplification, which means that no action is carried out and the next simplification is processed.

Second Stage

The second stage refers to faces belonging to the lowest resolution lod, M_{n-1}. It consists of traversing the part of the data structure with regard to M_{n-1} and drawing the faces marked as *DRAWABLE*. These faces complete the required lod.

Example

Fig. 3 shows the object *Cow*, where three sets of faces can be distinguished according to the color. Each color identifies the stage and the reason why the faces have been drawn:

- Light gray: faces drawn in the first stage because the criterion defined by the application decides to preserve detail in that area of the object.
- Medium gray: faces drawn in the first stage because, although the criterion defined by the application decides to simplify them, the simplification is not performed due to dependencies, that is, there are faces marked as *NO_DRAWABLE*.
- Dark gray: faces drawn in the second stage of the algorithm.

Another example of a variable resolution lod obtained with the algorithm in Fig. 5 is shown in Fig. 6.

3.3 Computational Cost

The analysis of the computational cost shows that in the worst case the algorithm is $O(nK + F^r + V^r)$, where n is the number of simplifications, K is the cost of the evaluation of the criterion defined by the application, and F^r, V^r are the total number of faces and vertices of the multiresolution representation, respectively. If K is constant, the first term is of lower order and it can be disregarded. F^r and V^r are, in the worst case, quadratic with respect to the size of the original mesh (F and V, respectively). However, taking into account that each mesh vertex is shared by six adjacent faces on average, and using any simplification method based on edge contraction or vertex decimation in the construction process of the multiresolution representation, F^r and V^r are linear in relation to F and V, respectively.

4 RESULTS

The experiments were carried out using a Silicon Graphics RealityEngine 2, with a MIPS R10000 at 194 MHz and 256 Mb RAM. Coding of the model was in *C++* and the *OpenGL* graphics library was used. The simplification method used to construct the multiresolution representations was that proposed by Garland

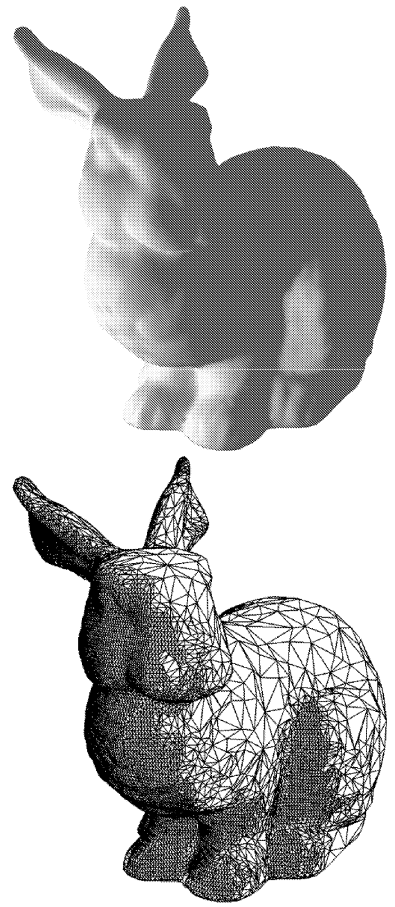


Figure 6: Example of a variable resolution lod. The illuminated area of the object is represented with the highest resolution. The visualization of the object is shown at the top, and the generated mesh is shown at the bottom.

and Heckbert [Garla99a] based on edge contraction, but forcing the contraction point to be one of the edge vertices. The meshes come from the *Stanford University Computer Graphics Laboratory* (http://www-graphics.stanford.edu/data/3Dscanrep/) and *Cyberware* (http://www.cyberware.com/models/).

The aim of the experiments is to observe and compare the visualization times of the proposed algorithm when the area in high detail varies in size. Results from two models are shown: *Bunny* and *Phone*. The characteristics of these models are shown in table 1. For each model we select an area where detail is preserved. This area is defined by using a sphere whose center is indicated interactively and whose radius varies, so that the surface outside the sphere is simplified.

Figs. 7(a) and 7(b) show the performance of the

	Bunny	Phone
Vertices	34,835	83,045
Faces	69,451	165,963
Lods	33,990	81,668
MB.	2.876	6.940

Table 1: Characteristics and storage costs.

algorithm as the size of the area in high detail increases. Axis X represents the number of faces of the visualized lod. Axis Y represents the number of frames per second achieved. A curve, called *Uniform*, has been included which represents the number of frames per second achieved when visualizing the original model. Figs. 7(c)(d)(e) and 7(f)(g)(h) show three images of each model, which correspond to three different variable resolution lods of this experiment.

As a result of these experiments, we conclude that the algorithm increases its performances when the area in high detail is small compared with the size of the model. The smaller the size of the area, the better the performance of the algorithm and vice versa, the performance decreases as the size of the area increases.

5 CONCLUSIONS

In this paper an algorithm to retrieve variable resolution lods from a MOM representation has been presented. There is no need to reorganize the data in a new structure, or to explicitly store new information about dependencies between the different triangulations of the represented object. MOM implicitly encodes enough information to correctly retrieve any variable resolution lod. Consequently, the MOM construction process creates a multiresolution representation from which both uniform and variable resolution lods can be retrieved, without the need for an additional process which could increment its cost and complexity. In addition, the generality property of MOM is retained: both manifold and non-manifold surfaces can be represented and any simplification method can be used in the construction process.

The experimental results show an improvement in performance when the area in high detail is small compared with the whole surface. However, we think that further improvement can be achieved. From now on, this work will proceed towards the improvement of the algorithm performance. In order to do this, the aim is to reduce the number of simplifications to traverse by maintaining a dynamic list which contains those simplifications that contribute to the solution with some of their faces. The dynamic list can be created for each frame, or it can be updated by exploiting the frame-to-frame coherence property, i.e., by eliminating and adding simplifications to the current dynamic list. In this way we expect the visualization times to be further improved. Given that the dynamic list is not expected to be large, the increment in the use of memory is expected to be low.

REFERENCES

[Abadj99] Abdajev, V., del Rosario, M., Lebedev, A., Migdal, A. and Paskhaver, V.: Metastream, *Proc. of VRML'99*, 1999.

[Berg95] De Berg, M. and Dobrindt, K. On levels of detail in terrains, *Proc. of 11th ACM Symposium on Computational Geometry*, pp. C26–C27, 1995.

[Ciamp97] Ciampalini, A., Cignoni, P., Montani, C. and Scopigno, R.: Multiresolution Decimation Based on Global Error, *The Visual Computer*, Vol. 13, pp. 228–246, 1997.

[Cigno97] Cignoni, P., Puppo, E. and Scopigno, R.: Representation and visualization of terrain surfaces at variable resolution, *The Visual Computer*, Vol. 13, pp. 199–217, 1997.

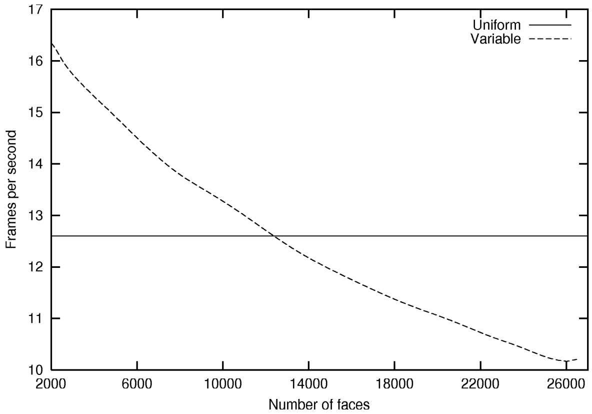
[Elsan99] El-Sana, J. and Varshney, A.: Generalized View-Dependent Simplification, *Proc. of EUROGRAPHICS '99*, pp. 83–94, 1999.

[Flori96] De Floriani, L., Marzanoo, L. and Puppo, E.: Multiresolution models for topographic surface description, *The Visual Computer*, Vol. 12(7), pp. 317–345, 1996.

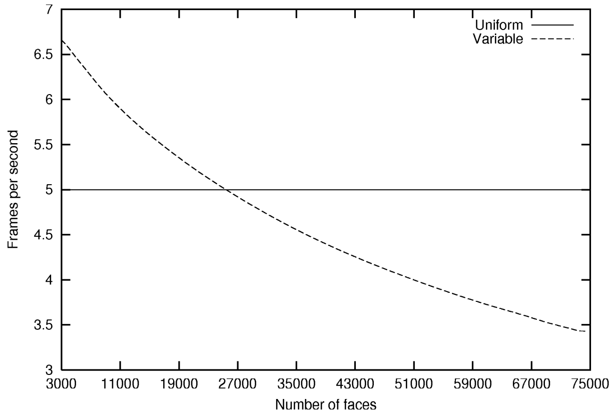
[Flori97a] De Floriani, L., Puppo, E. and Magillo, P.: A Formal Approach to Multiresolution Hypersurface Modeling, *Geometric Modeling: Theory and Practice*, W. Straber, R. Kein, R. Rau (Editors), Springer-Verlag, 1997.

[Flori97b] De Floriani, L., Magillo, P. and Puppo, E.: Building and Traversing a Surface at Variable Resolution, *Proceedings IEEE Visualization '97*, pp. 103–110, 1997.

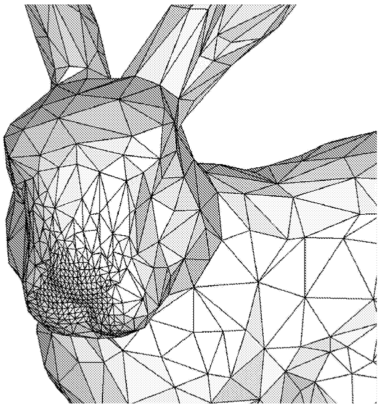
[Flori98] De Floriani, L., Magillo, P. and Puppo, E.: Efficient Implementation of Multi-Triangulations, *Proc. of Visualization '98*, pp. 43–50, 1998.



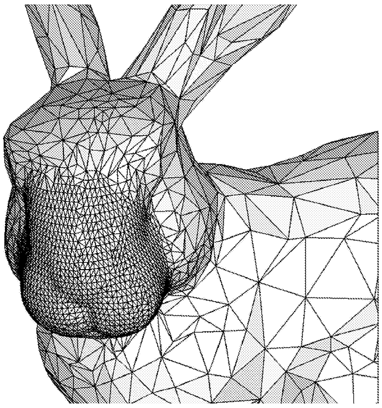
(a) Bunny



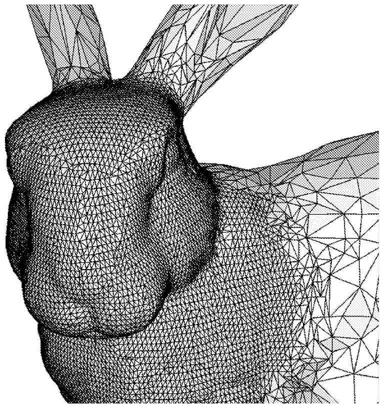
(b) Phone



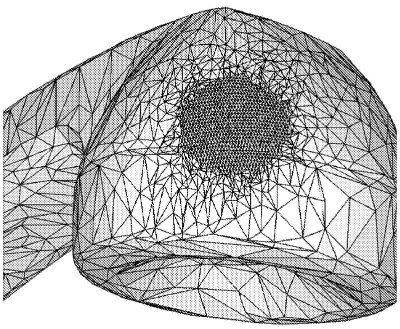
(c) 2,107 faces, 16.4 fps



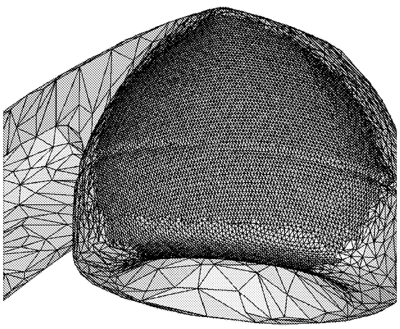
(d) 4,625 faces, 15.4 fps



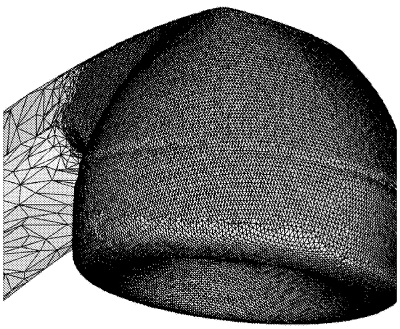
(e) 14,216 faces, 12.1 fps



(f) 4,794 faces, 6.6 fps



(g) 17,464 faces, 5.5 fps



(h) 62,132 faces, 3.7 fps

Figure 7: Results obtained from *Bunny* and *Phone* models.

[Garla99a] Garland, M.: Quadric-Based Polygonal Surface Simplification, *PhD thesis, Carnegie Mellon University, CS Dept.*, Tech. Rept. CMU-CS-99-105, 1999.

[Garla99b] Garland, M.: Multiresolution Modeling: Survey & Future Opportunities, *State of the Art Reports of EUROGRAPHICS '99* , pp. 111–131, 1999.

[Guezi98] Guéziec, A., Taubin, G., Lazarus, F. and Horn, W.: Simplicial maps for progressive transmission of polygonal surfaces, *Proc. of VRML'98*, 1998.

[Guezi99] Guéziec, A., Taubin, G., Horn, B. and Lazarus, F.: A Framework for Streaming Geometry in VRML, *IEEE Computer Graphics and Applications*, Vol. 19(2), pp. 68–78, 1999.

[Heckb94] Heckbert, P. and Garland, M.: Multiresolution modeling for fast rendering, *Proc. of Graphics Interface '94*, pp. 43–50, 1994.

[Heckb97] Heckbert, P. and Garland, M.: Survey of polygonal surface simplification algorithms, *Multiresolution Surface Modeling, Course Notes of SIGGRAPH'97*, 1997

[Hoppe96] Hoppe, H.: Progressive Meshes, *Proc. of SIGGRAPH '96*, pp. 99–108, 1996.

[Hoppe97] Hoppe, H.: View-Dependent Refinement of Progressive Meshes, *Proc. of SIGGRAPH '97*, pp. 189–198, 1997.

[Hoppe98a] Hoppe, H.: Efficient Implementation of Progressive Meshes, *Computer & Graphics*, Vol. 22(1), pp. 27–36, 1998.

[Hoppe98b] Hoppe, H.: Smooth view-dependent level-of-detail control and its application to terrain rendering, *Proc. of Visualization '98*, pp. 35–42, 1998.

[Linds96] Lindstrom, P., Koller, D., Ribarsky, W., Hodges, L.F., Faust, N. and Turner, G.A.: Real-time, continuous level of detail rendering of height fields, *Proc. of Siggraph '96*, pp. 109–118, 1996.

[Luebk97] Luebke, D. and Erikson, C.: View-Dependent Simplification of Arbitrary Polygonal Environments, *Proc. of SIGGRAPH '97*, pp. 199–208, 1997.

[Popov97] Popovic, J. and Hoppe, H.: Progressive simplicial complexes, *Proc. of SIGGRAPH'97*, pp. 217–224, 1997.

[Puppo96] Puppo, E.: Variable resolution terrain surfaces, *Proc. of Eight Canadian Conference on Computational Geometry*, pp. 202–210, 1996.

[Puppo97] Puppo, E. and Scopigno, R.: Simplification, LOD and Multiresolution - Principles and Applications, *Tutorial Notes of EUROGRAPHICS'99*, Vol. 16(3), 1997.

[Puppo98] Puppo, E.: Variable resolution triangulations, *Computational Geometry*, Vol.11(3-4), pp. 219–238, 1998.

[Ribel98] Ribelles, J., Chover, M., Huerta, J. and Quirós, R.: Multiresolution Ordered Meshes, *Proc. of 1998 IEEE Conference on Information Visualization*, pp. 198–204, 1998.

[Ribel99] Ribelles, J., Chover, M., López, A. and Huerta, J.: A First Step to Evaluate and Compare Multiresolution Models, *Short Papers and Demos of EUROGRAPHICS '99*, pp. 230–232, 1999.

[Rossi96] Rossignac, J. and Borrel, P.: Multiresolution 3D approximations for rendering complex scenes, *Modeling in Computer Graphics: Methods and Applications*, pp. 455–465, 1993.

[Scarl90] Scarlatos, L.L.: A refined triangulation hierarchy for multiples level of terrain detail, *Proc. of Image V Conference*, pp. 115–122, 1990.

[Schro92] Schroeder, W.J., Zarge, J.A. and Lorensen, W.E.: Decimation of Triangle Meshes, *Proc. of SIGGRAPH '92*, Vol. 26(2), pp. 65–70, 1992.

[Xia96] Xia, J. and Varshney, A.: Dynamic view-dependent simplification for polygonal models, *Proc. of IEEE Visualization '96*, pp. 327–334, 1996.

[Xia97] Xia, J., El-Sana, J. and Varshney, A.: Adaptive Real-Time Level-of-detail-based Rendering for Polygonal Models, *IEEE Transactions on Visualization and Computer Graphics*, pp. 171–183, 1997.