

Multiresolution Topology Estimation of 3-D Range Data

In Kyu Park

Multimedia Lab.

Samsung Advanced Institute of Technology

YONGIN 449-712, KOREA

saitpik@sait.samsung.co.kr

Kyoung Mu Lee

Dept. of Electronics and Electrical Eng.

Hong-Ik University

SEOUL 121-791, KOREA

kmlee@wow.hongik.ac.kr

Ju Yong Jang Sang Uk Lee

School of Electrical Eng.

Seoul National University

SEOUL 151-742, KOREA

{jangbon, sanguk}@diehard.snu.ac.kr

Abstract

In this paper, we propose a new efficient algorithm to construct a multiresolution polygonal mesh as a topology estimation method. The proposed algorithm initially segments the range data into a finite number of patches using the K-means clustering algorithm. Each patch is then approximated by an appropriate polyhedron and divided into triangles, yielding finally a triangular mesh model. By controlling the tolerance of the modeling error, multiresolution representation of the estimated topology also can be established efficiently. Moreover, in order to improve the equiangularity of each triangle, we employ the dynamic mesh model [3], so that the mesh adaptively find its equilibrium state, according to the equiangularity constraint. Experimental results demonstrate that satisfactory equiangular mesh models are constructed efficiently at various resolutions, while yielding tolerable error.

1 Introduction

In computer vision and computer graphics, range data plays an important role in many applications, since it provides an explicit geometrical information on the surface of an underlying 3-D object. Recent progress in range-finding techniques, such as laser range scanner and space encoding range finder, allow us to acquire dense range data with tolerable error. In addition, by employing proper registration and integration techniques [8], multiple range data of an object obtained in different views can be transformed into a common coordinates system, so that a complete 3-D range data of the physical model can be reconstructed.

However, since the range data is in itself merely a set of dense points, an explicit 3-D model for the underlying object should be obtained for further high level processing. In this context, the modeling technique to convert the raw range data into a suitable surface model is quite an important issue, and much efforts have been made to develop such technique [5][12]. Applications of this technique can

be found in the field of 3-D modeling including rapid prototyping, reverse engineering, virtual and augmented environments.

So far, a number of algorithms have been proposed by several researchers to construct a surface model from a set of range data, which can be roughly categorized into two groups based on their approaches: polygonal mesh based [5] and B-splines based methods [12]. In this paper, we focus on the techniques using polygonal mesh model. Note that triangular mesh has been used most widely in 3-D graphics and vision, since it can represent complex free-form objects efficiently.

In this paper, we propose a new algorithm to estimate the underlying topology using triangular mesh, which is achieved in a top-down strategy. This paper is organized as follows. The problem is defined and the proposed approach is briefly introduced in Section 2. In Section 3, K-means clustering technique is discussed, with which point patches are generated. In Section 4, the topology estimation algorithm, including polygonal approximation and triangular mesh generation, is described in detail. In Section 5, we introduce the dynamic mesh briefly and present the mesh adaptation algorithm. Next, experimental results are provided in Section 6. Finally, we give the conclusive remark in Section 7.

2 Problem Statements and Overview

In our approach, it is assumed that the input data format could be either points cloud or dense polygonal mesh. For the different choices of the input format, the problem of topology estimation is defined as follows.

- Topology estimation of points clouds: If the input is a points cloud, the problem is to estimate the underlying topology of the object by constructing polygonal mesh.
- Topology estimation of dense mesh: If the input is a dense mesh, the problem reduces to a accelerated and memoryless mesh simplification, in which the low resolution mesh is obtained quickly.

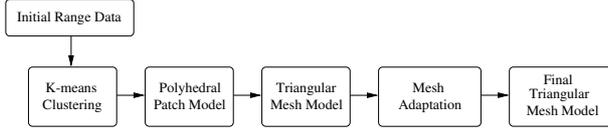


Figure 1. Overview of the proposed algorithm.

Note that, in topology estimation of dense mesh, initial dense mesh is not simplified in the iterative manner as the previous algorithm. Instead, it creates a low level mesh without any prior knowledge of other levels.

In constructing triangular surface model, the common measure for the ‘good’ triangulation is the approximation error and the equiangular property. The approximation error is directly affected by the resolution of the mesh model, and this can be efficiently adjusted by the number of initial patches in the proposed algorithm. In our work, we prefer equiangular triangle rather than sharp and narrow triangle, since it has less distortion in shaded rendering and is more useful in further processing including mesh editing and tessellation. Equiangular mesh is also known to increase upper bound on the curvature of sampled surface, assuming that the shape is locally sphere-like.

The proposed modeling algorithm consists of three stages: K -means clustering of the input data, polygonal approximation and triangular mesh generation, and mesh adaptation. The overall block diagram is shown in Figure 1. In our approach, firstly by using the K -means clustering technique, the input range data is partitioned into point patches. Each pair of point patches is then tested to retrieve the adjacency information. And using this adjacency relations, each point patch is approximated by a polygon, yielding the initial polyhedral surface model. Then a triangular mesh is obtained from this polyhedral model by means of polygonal division. In this procedure, note that the resolution of the initial polyhedral model and eventually that of the final triangular mesh can be controlled by varying the number of clusters in K -means clustering stage. In mesh adaptation, in order to increase the equiangular property of the estimated topology in our approach, the mesh configuration is updated iteratively to converge to the reference mesh, yielding finally an equiangular mesh. Note that the mesh is modeled as dynamic spring model and the reference mesh is configured to be most equiangular deformation.

3 Voronoi Partitioning

The proposed mesh construction algorithm begins with partitioning of the range data using the K -means clustering technique. K -means clustering is performed based on the nearest neighbor criterion, yielding Voronoi partitioning of the range data.

3.1 K -Means Clustering

The K -means clustering [6] algorithm clusters multidimensional data, by minimizing the sum of the distances between each point and the cluster centers. This is useful technique in clustering unorganized data, especially when no additional information is provided, except the position of the data. In our approach, the K -means algorithm is adopted to partition 3-D range data into point patches for the polygonal approximation. In order to find the center of the point patches, we use the LGB algorithm [1] to determine the center of each cluster, $C = \{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_K\}$, where K is the number of clusters. The LGB algorithm can be summarized as follows.

- (1) Choose an initial set of centroids
 $C = \{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_K\}$.
- (2) Determine the Voronoi region for each \mathbf{y}_i .
- (3) Compute the centroid of each Voronoi region.
- (4) If it has not converged, go to step 2. Otherwise stop.

In determining the Voronoi region, each range data is clustered into K point patches P_i ($i = 1, 2, \dots, K$), using the following minimum distance criterion.

$$P_i = \{\mathbf{x} \mid d(\mathbf{x}, \mathbf{y}_i) \leq d(\mathbf{x}, \mathbf{y}_j) \text{ for all } j (\neq i)\} \quad (1)$$

$$d(\mathbf{x}, \mathbf{y}_i) = \mathbf{x} \cdot \mathbf{y}_i - \frac{1}{2} \|\mathbf{y}_i\|^2. \quad (2)$$

In LGB algorithm, the choice of the initial set of centroid is important, since the algorithm would converge to a local minimum depending on the initial selection. In our approach, we use the splitting technique, in which the number of clusters is increased from 1 to K . The centroid of point patch with largest variance is split into two by choosing two centroids as random perturbations.

There are several advantages in applying the K -means clustering algorithm. First, by using the divide-and-conquer method, the computation burden can be reduced significantly in further processing. Note that since a set of range data is very large, the conventional point-wise manipulation requires enormous amount of computational cost. On the other hand, by partitioning the data into a number of patches and manipulate them, our algorithm can reduce the search space drastically and alleviate the computational cost. Second, the resultant patches are regular in shape, due to the characteristics of Voronoi region. Thus, in turn, the approximated polygon becomes also very regular, which is one of the desirable features in constructing the mesh structure.

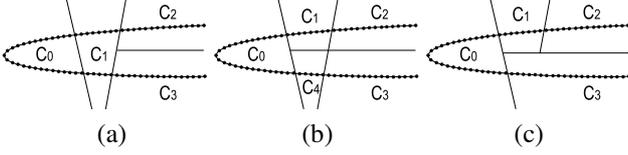


Figure 2. Subclustering of a point patch. (a) Non-minifold point patch (C_1). (b) Removing non-manifold patch by creating a new patch (C_5). (c) Removing non-manifold patch by neighborhood-reclustering (C_0 and C_3).

3.2 Subclustering of Point Patch

Since the distance measure in (2) is Euclidian distance, the resultant point patch can be non-manifold structured¹. For example, when the object is relatively thin compared with the radius of the point patch, some of the resultant patches could consist of points which lie on opposite side of the thin surface, as shown in Figure 2 (a). In further processing, this produces folded polygons which are topologically incorrect.

The problem can be solved by adopting different distance measure, *i.e.*, geodesic distance. However, the geodesic distance computation is computationally costly. Even though fast algorithm is available, it is definitely impractical in this application due to the large size of the range data. In our approach, in order to solve the problem, non-manifold patch is subclustered, yielding two manifold point patches, as shown in Figure 2 (b). On the other hand, if the new patch has few points, the relative size would be too small, which causes irregular topology in further processing. Therefore, in this case, the smaller part (lower subcluster of C_2 in Figure 2 (a)) is reclustered to the neighboring clusters as shown in Figure 2 (c).

4 Triangular Mesh Construction

In this section, we describe the proposed algorithm to estimate the underlying topology by constructing initial triangular mesh from the point patches. Algorithms are developed for points cloud, since the dense mesh structure is eventually a special case of points cloud. Note that much of the details can be simplified by using the connection information of mesh if the input is dense triangular mesh.

4.1 Finding the Adjacency between Point Patches

Consider two point patches P_i and P_j , which consists of p and q points, respectively, as

$$P_i = \{\mathbf{x}_{i0}, \mathbf{x}_{i1}, \dots, \mathbf{x}_{ip-1}\} \quad (3)$$

¹A point patch is called manifold structured when all the point in the patch is geodesically adjacent.

$$P_j = \{\mathbf{x}_{j0}, \mathbf{x}_{j1}, \dots, \mathbf{x}_{jq-1}\} \quad (4)$$

Let $O_\delta(\mathbf{x})$ be an openball centered at \mathbf{x} with radius δ , and define $O_\delta(P_i)$ and $O_\delta(P_j)$ using the openballs as (6).

$$O_\delta(P_i) = O_\delta(\mathbf{x}_{i0}) \cup O_\delta(\mathbf{x}_{i1}) \cup \dots \cup O_\delta(\mathbf{x}_{ip-1}) \quad (5)$$

$$O_\delta(P_j) = O_\delta(\mathbf{x}_{j0}) \cup O_\delta(\mathbf{x}_{j1}) \cup \dots \cup O_\delta(\mathbf{x}_{jq-1}) \quad (6)$$

Assume that P_i and P_j are adjacent each other. Then, for some δ , at least one point of P_j is included in $O_\delta(P_i)$ and also at least one point of P_i is included in $O_\delta(P_j)$. In other words, if the proposition (7) is true, then P_i and P_j are considered to be adjacent.

$$(O_\delta(P_i) \cap P_j \neq \emptyset) \wedge (O_\delta(P_j) \cap P_i \neq \emptyset) \quad (7)$$

If the radius, δ , is too large, there would be false alarm for the patches which are not actually adjacent. On the contrary, if δ is too small, the adjacency could not be detected. In this research, we found empirically that one fourth of the mean radius of P_i and P_j results in good performance, such that

$$\delta = \frac{1}{4} \left(\frac{r_i + r_j}{2} \right), \quad (8)$$

where the radius of a point patch is defined as the maximum distance between patch points and the centroid,

$$r_i = \|\mathbf{x} - \mathbf{y}_i\|_\infty, \text{ for all } \mathbf{x} \in P_i \quad (9)$$

4.2 Polygonal Approximation of a Point Patch

After clustering the data, based on the adjacency information to neighboring patches, each point patch is then approximated by a proper polygon. This can be done easily, by constructing the Patch Adjacency Table (PAT), which describes the adjacency relations between every pair of point patches. For the detail, refer to our previous work [15].

4.3 Mesh Generation

The triangular mesh structure can be constructed by connecting each vertices of polygon and the centroid of the point patch. Note that the vertices of polygon and the centroid of the point patch are exactly on the sampled surface, thus the nodes of the triangular mesh is also coincide with the sampled data.

4.4 Multiresolution Triangulation

Multiresolution description technique have been widely used to control the visual description in multiple levels of detail (LOD), according to the graphic performance in the field of computer graphics, vision and virtual environments.

In our proposed 3-D surface modeling approach, the multiresolution description can be easily implemented, by

controlling the number of the clusters for the initial polyhedral model. In general, there exists a trade-off relationship between the resolution of modeling and the approximation error. Thus, by increasing or decreasing the number of polygonal patches, it is possible to control the resolution and approximation error. Note that, in our approach, the number of polygonal patches can be adjusted easily by the number of clusters, K , in the K -means algorithm.

Therefore, if the approximation error of the initial model is above a specified error bound, we can reduce the error to be under the bound, by increasing K appropriately.

5 Mesh Adaptation Using Dynamic Model

Adaptive mesh [3] has been proposed for nonuniform sampling and reconstruction of the intensity and range image data. It is a dynamic model assembled as topologically regular collections of nodal masses connected by adjustable springs. The dynamic mesh automatically updates itself until the equilibrium state, driven by the nodal and data forces.

In our approach, we employ the adaptive dynamic mesh technique to improve the equiangularity condition of the reconstructed triangular mesh. Consider a node N_i which is connected to node N_j by a spring with natural length l_{ij} and stiffness c . Then, the force that the spring exerts on N_i is defined as

$$\vec{s}_{ij} = \frac{c e_{ij}}{\|\vec{r}_{ij}\|} \vec{r}_{ij} \quad (10)$$

where,

$$\begin{aligned} \vec{r}_{ij} &= \vec{x}_j - \vec{x}_i \\ e_{ij} &= \|\vec{r}_{ij}\| - l_{ij}. \end{aligned} \quad (11)$$

In (11), \vec{x}_i and \vec{x}_j are the positional vector of N_i and N_j , respectively. Due to the nodal and external forces, the position of each node with mass m and the damping coefficient γ is governed by following second-order nonlinear ordinary differential equation, given by

$$m \frac{d^2 \vec{x}_i}{dt^2} + \gamma \frac{d \vec{x}_i}{dt} + \vec{g}_i = \vec{f}_i \quad (12)$$

$$\vec{g}_i = \sum_j \vec{s}_{ij}, \quad (13)$$

where \vec{f}_i is external force at node N_i . Note that the convergence speed can be controlled, by adjusting m and γ , and it becomes slower as m and γ increase.

Equation (13) can be solved numerically by using the Euler time-integration method, and the corresponding iterative equations can be derived as

$$\vec{f}_i^{new} = \vec{f}_i - \gamma \vec{v}_i - \vec{g}_i \quad (14)$$

$$\vec{a}_i^{new} = \frac{\vec{f}_i^{new}}{m} \quad (15)$$

$$\vec{v}_i^{new} = \vec{v}_i + \Delta t \vec{a}_i^{new} \quad (16)$$

$$\vec{x}_i^{new} = \vec{x}_i + \Delta t \vec{v}_i^{new}, \quad (17)$$

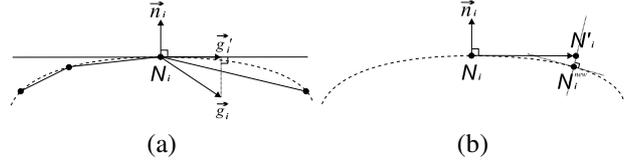


Figure 3. Movement of a node. (a) Determine the tangential nodal force \vec{g}'_i first by projecting \vec{g}_i onto the tangent plane. (b) then locate the new position N_i^{new} by projecting the tangential displacement on the sampled surface.

where Δt is the time step. We observe that as Δt increases, although the convergence rate is accelerated, the probability of converging to local minima also increases. By using (17), the initial mesh is updated iteratively to be stabilized, with proper choice of m , γ , and Δt .

In order to improve the equiangularity of the initial mesh using the adaptive mesh technique, we first establish a virtual reference mesh to which the initial mesh should converge. Each triangle of the reference mesh is equiangular and has the same area of the corresponding triangle of the initial mesh. Now, in order to assure the convergence, the natural length, l_{ref} , of each spring in the initial mesh is set to be the length of a side of the corresponding equiangular triangle in the reference mesh. However, note that since a spring is essentially shared by two triangles, l_{ref} is not uniquely determined. Thus, to overcome this non-uniqueness problem, we choose the mean of two candidates for the natural length. Let S_1 and S_2 be the area of triangles which share a common spring. Then, the natural length l_{ref} is defined as

$$l_{ref} = \frac{l_1 + l_2}{2}, \quad (18)$$

where the length l_i is given by

$$l_i = \sqrt{\frac{4\sqrt{3}}{3} S_i}, \quad i = 1, 2. \quad (19)$$

Note that since the shape of adjacent triangles is similar enough, the mean natural length, l_{ref} , of shared spring is possibly expected to improve the equiangularity of both triangles simultaneously.

In general, since the approximation error is significantly affected by the nodal movement, due to the data force, modeling and determining the external force, *i.e.*, data force \vec{f}_i in (13) is another important issue in designing dynamic adaptive mesh. Usually, this problem becomes more difficult when we deal with scattered range data, rather than intensity or range image. Thus, in order to solve the problem, in our approach, we constrain the movement of the nodes to be only on the sampled surface, so that the modeling error, and thus the data force at each node become zero. In Figure 3,

a two dimensional illustration of our approach is shown, in which the dotted curve denotes the sampled surface. Let the total nodal force at N_i by the attached springs to be \vec{g}_i . Then, \vec{g}_i is projected onto the tangent plane, resulting in the tangential nodal force \vec{g}_i^t , which is the effective nodal force is in this case. Now, as shown in Figure 3 (b), the node N_i is moved to the position of N_i^t by the tangential force, and then the new node N_i^{new} is finally localized on the sampled surface through the projection of N_i^t onto it. Since the nodal position is tied on the sampled surface, the approximation error varies not much during the iteration process, while the equiangularity is greatly improved.

6 Experimental Results

In order to evaluate the performance of the proposed algorithm, we have carried out the experiments on several range data set. Multiresolution topology estimation is performed for the models in Figure 4 (a)~(c), while polyhedral and triangular mesh with different choices of K are constructed. The results are shown in Figure 4 (d)~(r). In Figure 4 (d)~(h), the position of the codebook vector is shown. After K -means clustering and polygonal approximation of each point patch, the polyhedron model is obtained as shown in Figure 4 (i)~(m), in which K is set to 800, 100, 1500, 100, and 1500, respectively. The constructed triangular mesh model is shown in Figure 4 (n)~(r). It is observed that the triangles on the reconstructed mesh is quite equiangular, because of the mesh adaptation.

Note that the purpose of the mesh adaptation is to maximally increase the equiangularity of the initial mesh. Figure 4 (s)~(u) show the histogram change before and after the mesh adaptation. Each initial mesh of the model is modelled as the dynamic mesh and updated iteratively for 50 times. It is observed that the equiangularity is sufficiently improved. Note that the triangular mesh in Figure 4 (n)~(r) has gone through this adaptation procedure. It is visually clear that the resultant mesh has quite regular structure.

7 Conclusion

In this paper, based on a combined statistical and dynamical methods, we proposed an efficient algorithm to estimate the underlying topology of 3-D range data in forms of the equiangular triangular mesh model. Unlike the conventional methods, by adopting a top-down approach, the proposed algorithm can not only manipulate unorganized and scattered 3-D range data efficiently, but also reduce the computational cost required in modeling, especially for large and dense data set. By using the K -means clustering algorithm sequentially, multiresolution topology estimation is effectively achieved, and also the triangulation can be accomplished efficiently. Moreover, in order to increase the

equiangularity of the constructed triangular mesh, the initial mesh is modeled as a dynamic mesh structure, so that it refines itself iteratively through the stabilization process, driven by the nodal forces and the equiangular constraint.

References

- [1] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantization design," *IEEE Trans. on Communication*, vol. 28, no. 1, pp. 84-95, Januray 1980.
- [2] L. Floriani, B. Falcidieno, and C. Pienovi, "Delaunay-based representation of surfaces defined over arbitrarily shaped domains," *Computer Vision, Graphics, and Image Processing*, vol. 32, no. 1, pp. 127-140, 1985.
- [3] D. Terzopoulos and M. Vasilescu, "Sampling and reconstruction with adaptive mesh," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 70-75, June 1991.
- [4] M. Vasilescu and D. Terzopoulos, "Adaptive meshes and shells: irregular triangulation, discontinuities, and hierarchical subdivision," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 829-832, June 1992.
- [5] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, "Surface reconstruction from unorganized points," *Proceedings of SIGGRAPH '92*, pp. 71-78, July 1992.
- [6] J. L. Marroquin and F. Girosi, "Some extensions of the K -means algorithm for image segmentation and pattern recognition," AI Memo 1930, MIT Artificial Intelligence Laboratory, January 1993.
- [7] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, "Mesh optimization," *Proceedings of SIGGRAPH '93*, pp. 19-26, August 1993.
- [8] G. Turk and M. Levoy, "Zippered polygon meshes from range images," *Proceedings of SIGGRAPH '94*, pp. 311-318, August 1994.
- [9] T. Fang and L. Piegl, "Delaunay triangulation in three dimensions," *IEEE Computer Graphics and Applications*, vol. 15, no. 5, pp. 62-69, September 1995.
- [10] M. Eck, T. DeRose, T. Duchamp, H. Hoppe, M. Lounsbury, and W. Stuetzle, "Multiresolution analysis of arbitrary meshes," *Proceedings of SIGGRAPH '95*, pp. 173-182, August 1995.
- [11] M. Soucy and D. Laurendeau, "Multiresolution surface modeling based on hierarchical triangulation," *CVGIP: Image Understanding*, vol. 63, no. 1, pp. 1-14, January 1996.
- [12] V. Krishnamurthy and M. Levoy, "Fitting smooth surfaces to dense polygon meshes," *Proceedings of SIGGRAPH '96*, pp. 313-324, August 1996.
- [13] A. Guéziec, "Locally tolerated surface simplification," *IEEE Trans. on Visualization and Computer Graphics*, vol. 5, no. 2, pp. 168-189, April-June 1999.
- [14] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*, Kluwer Academic Publishers, 1992.
- [15] I. K. Park, I. D. Yun, and S. U. Lee, "Automatic 3-D model synthesis from measured range data," *IEEE Trans. on Circuits and Systems for Video Technology*, vol 10, no. 2, pp. 293-301, March 2000.

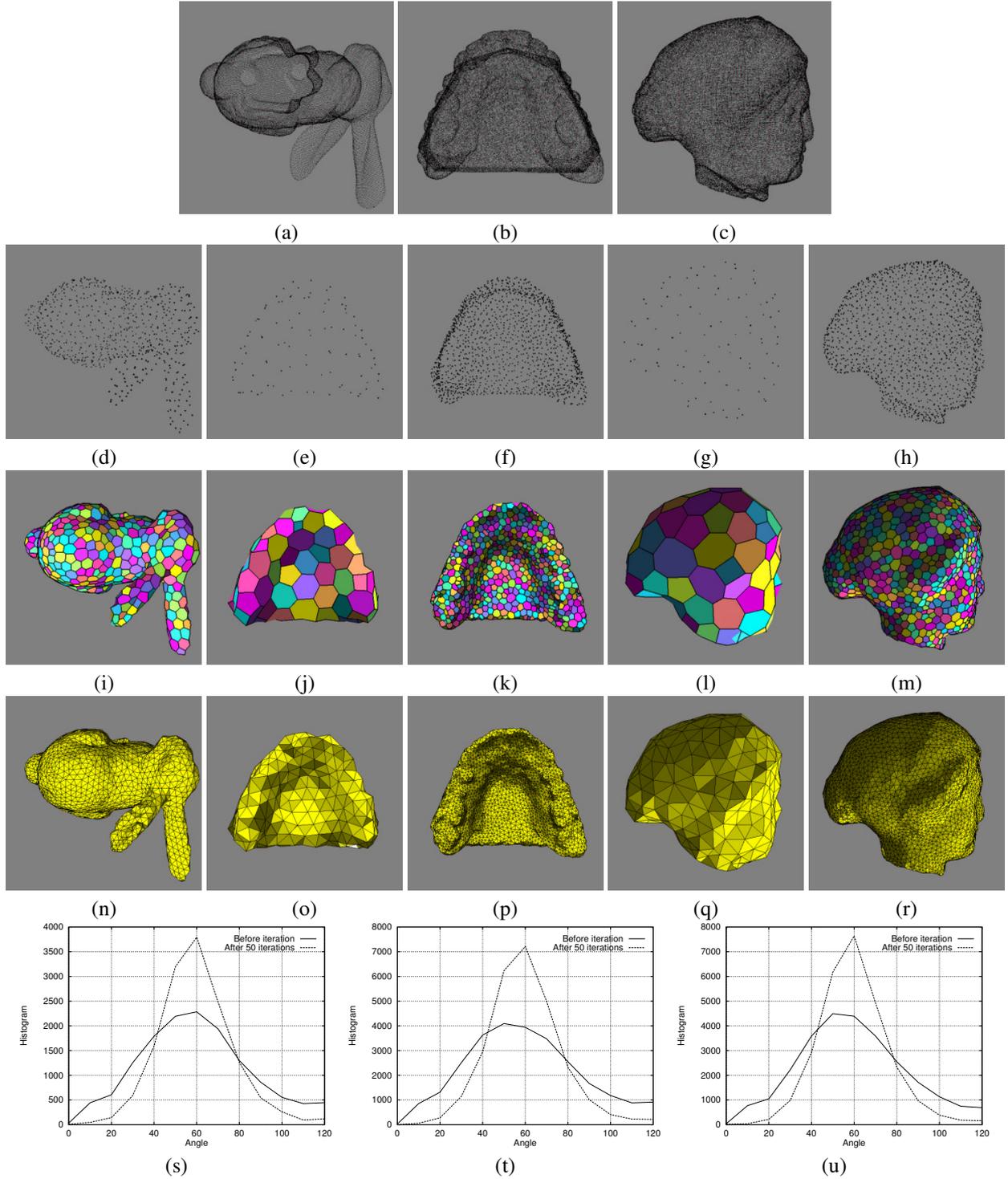


Figure 4. Multiresolution topology estimation results. (a) Bunny data ($N = 35,947$). (b) Teeth data ($N = 58,303$). (c) Venus data ($N = 67,173$). (d) Codebook vector ($K = 800$). (e) Codebook vector ($K = 100$). (f) Codebook vector ($K = 1,500$). (g) Codebook vector ($K = 100$). (h) Codebook vector ($K = 1,500$). (i-m) Polyhedral model. (n-r) Triangular mesh model after mesh adaptation. (s) Angle histogram of Bunny model before and after the mesh adaptation. $K = 800$, $m = 0.01$, $\gamma = 0.2$, $c = 1$, $\Delta t = 0.05$. (t) Angle histogram of Teeth model ($K = 1,500$). (u) Angle histogram of Venus model ($K = 1,500$).