Models and Algorithms for Efficient Multiresolution Topology Estimation of Measured 3-D Range Data

In Kyu Park, Kyoung Mu Lee, and Sang Uk Lee

***Abstract*—In this paper, we propose a new efficient topology estimation algorithm to construct a multiresolution polygonal mesh from measured three-dimensional (3-D) range data. The topology estimation problem is defined under the constraints of cognition, compactness, and regularity, and the algorithm is designed to be applied to either a cloud of points or a dense mesh. The proposed algorithm initially segments the range data into a finite number of Voronoi patches using the K-means clustering algorithm. Each patch is then approximated by an appropriate polygonal and eventually a triangular mesh model. In order to improve the equiangularity of the mesh, we employ a dynamic mesh model, in which the mesh finds its equilibrium state adaptively, according to the equiangularity constraint. Experimental results demonstrate that satisfactory equiangular triangular mesh models can be constructed rapidly at various resolutions, while yielding tolerable modeling error.**

***Index Terms*—3-D range data, dynamic mesh, K-means clustering, multiresolution, polygonal mesh, topology estimation.**

I. Introduction

In many computer graphics and computer vision applications, range data plays an important role 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 techniques, multiple range views obtained in different views can be transformed and merged in a common coordinate system, so that a complete 3-D data can be reconstructed.

In computational geometry and computer graphics community, it has been also quite an important issue to achieve topology estimation that convert the raw range data into a surface model with desirable resolution. Note, that since it can represent complex free-form objects efficiently, the triangular mesh model has been used most widely as a surface representation scheme. Computational geometry researchers have focused mainly on the problem of estimating 3-D mesh models from unorganized range data [3], [5], [7], [10]–[13], while computer graphics researchers have on the surface simplification problem [8] and [9].

In order to produce a 3-D surface model from a set of unorganized point data, most of previous methods have employed the Delaunay triangulation technique in a bottom-up manner [3], [7], [10]–[12]. However, in general, it is computationally very complex and even erroneous to construct a Delaunay triangulation over a set of unorganized 3-D points, especially for nonconvex shapes. For the problem of surface simplification, known as the levels of detail (LOD) problem, a number of algorithms have been developed. They differ by the goodness-of-fit measures, mesh operations, and methods for determining new node positions and selecting vertex/edge orders [9]. However, in their approaches, the low resolution models can only be obtained after long sequence of simplifying processes, which means that they are eventually bottom-up approaches. Therefore, in these methods, the modeling errors are likely to be accumulated, and also unstable mesh operations often invoke undesirable mesh folding and topology corruption problems.

In this paper, we propose a novel algorithm to estimate the underlying topology of a 3-D points cloud. The proposed approach is perfectly new in that the triangulation is not performed at a point level as in the conventional approaches, but done in a very sparse "codebook" level, so that the computational complexity can be alleviated significantly. Moreover, by employing an iterative mesh adaptation process using mass-spring model, we can obtain a desirable estimated topology of a near-equiangular triangular mesh.

The main contribution of this paper could be summarized as follows: Both problems of topology estimation and simplification are solved simultaneously in a top-down manner, avoiding the use of complex computational geometry. Therefore, the proposed approach achieves low computational complexity, high modeling stability, and regularized near-equiangular mesh modeling.

This paper is organized as follows. In Section II, the problem of topology estimation is defined and the proposed approach is introduced briefly. In Section III, K-means clustering technique is discussed to generate Voronoi point patches. In Section IV, polygonal model approximation and triangular mesh generation are described in detail. In Section V, we introduce the dynamic mesh briefly and present the mesh adaptation algorithm. Next, the computational complexity of the proposed algorithm is addressed in Section VI. Experimental results are provided in Section VII. Finally, the conclusive remarks are drawn in Section VIII.

II. Problem Statement and Algorithm Overview

A. Problem Statement: Topology Estimation

In modern mathematics, intensive theoretical researches have been performed on the algebraic, combinatorial, differential, and point-set topology. The literal meaning of "topology" is "the study of places" or "the study of localities." This subject is a natural outgrowth of the Euclid's study of plane geometry. In this paper, we use the term "topology" in a reduced meaning of topological geometry [1], which is originated from the Euler's theorem. In this section, we first define "complete topological structure" of a 3-D surface, and then the problem of topology estimation for the sake of completeness.

A 3-D digital surface, or a triangular mesh, is defined as complete topological structure (CTS), if it is isomorphism with a sphere or a torus with N body holes. In other words, if a triangular mesh satisfies the following conditions, then it is CTS.

1) **Locally manifold:** Each vertex should be regular, i.e., its edge is formed of one simple polygonal curve. In a manifold triangular mesh, a pair of adjacent triangles share exactly one edge.
2) **Euler's theorem:** The numbers of vertex, edge, face, body hole should satisfy the Euler's formula.

The topology estimation problem is then can be defined as constructing a CTS triangular mesh from an input data. In order to construct a nice CTS mesh, three constraints are considered and should be optimized. First, the estimated topology should be easy to understand the underlying 3-D shape (cognition constraint). Secondly, the data size should be small (compactness constraint). Thirdly, the shape of triangles in the mesh should be regular enough (regularity constraints). Therefore, it is desirable to use the minimum number of triangles, while

Manuscript received November 15, 2001; revised July 15, 2002. This paper was recommended by Guest Editor H. Saito.

I. K. Park is with the Multimedia Laboratory, Samsung Advanced Institute of Technology, Yongin 449-712, Korea (e-mail: pik@ieee.org).

K. M. Lee is with the Visual Information Processing Laboratory, Department of Electronics and Electrical Engineering, Hong-Ik University, Seoul 121-791, Korea (e-mail: kmlee@wow.hongik.ac.kr).

S. U. Lee is with the Signal Processing Laboratory, School of Electrical Engineering and Computer Science, Seoul National University, Seoul 151-742, Korea (e-mail: sanguk@sting.snu.ac.kr).

Digital Object Identifier 10.1109/TSMCB.2003.814301

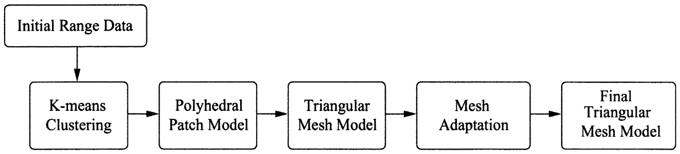


Fig. 1. Overview of the proposed algorithm.

maintaining the visual correctness of a 3-D model which consists of near-equiangular triangles. In our approach, it is assumed that the input data format could be either a cloud of points or a dense polygonal mesh. For different choices of the input format, the subproblems of topology estimation is defined as follows.

1) **Topology estimation of points cloud:** The problem is to estimate the underlying topology of an object by constructing the CTS triangular mesh. All the constraints should be considered.
2) **Topology estimation of dense meshes:** The problem is reduced to the accelerated and memoryless mesh reduction problem, in which compactness and regularity are improved while maintaining the CTS mesh structure.

Note that, in topology estimation of dense mesh, initial dense mesh is not reduced in an iterative manner as in the conventional algorithms [9]. Instead, it creates a low level mesh without any prior knowledge of other levels.

B. Algorithm Overview

In this paper, we propose a new algorithm to estimate the underlying topology using a triangular mesh through a top-down strategy. The block diagram of the proposed algorithm is shown in Fig. 1. The proposed algorithm employs a statistical pattern recognition technique, i.e., K-means clustering. The K-means clustering technique is used to partition the input range data into a finite number of Voronoi point patches. Utilizing the adjacency relation, each point patch is then approximated by a polygon, yielding an initial polyhedral surface model. From this polyhedral model, a CTS triangular mesh is obtained. The resolution of the initial polyhedral model and the final triangular mesh is controlled by varying the number of clusters in the K-means clustering stage.

In constructing a triangular surface model, one common measure for the "goodness" of a triangulation is the near-equiangular property. The theoretical background is traced back to the Delaunay triangulation [3] and [7]. One important property of the Delaunay triangulation is that it maximizes the minimum angle of any triangle in the triangulation. Therefore, the resultant triangles are as close to equiangular as possible. For graphics applications, a near-equiangular triangle yields less distortion in rendering and is more useful for mesh editing and tessellation than triangles with sharp angle. Note also that since it regularizes the mesh and thus reduces the entropy of the connectivity information, near-equiangular structures are strongly recommended in 3-D mesh compression.

In order to increase the near-equiangular property of the estimated topology in our approach, iterative mesh adaptation scheme is proposed. In mesh adaptation, a coarse mesh configuration is updated iteratively to resemble an imaginary reference mesh, yielding finally a more equiangular mesh. The mesh is represented by a dynamic spring model and the reference mesh is designed to be most equiangular with the constraint that the total mesh area is equal.

III. Voronoi Partitioning

A. 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 polygonal approximation. In order to find the center of the point patches $C = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_K\}$, where K is the number of clusters, we use the well-known LGB algorithm [2]. The LGB algorithm can be summarized as follows.

1) `Choose an initial set of centroids` $C = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_K\}$.
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 patche s P_i $(i = 1, 2, \ldots, K)$, using the following minimum distance criterion.

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

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

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

There are several advantages in adopting the K-means clustering algorithm. First, by using the divide-and-conquer method, the computational burden can be reduced significantly in the subsequent processes. Note that since the size of 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, since the resultant patches are regular in shape due to the characteristics of the Voronoi region, the approximated polygons become very regular as well. This is a very desirable feature for the construction of a mesh structure.

B. Subclustering of Point Patches

Since the distance measure in (2) is the Euclidian distance, the resultant point patch might not have a local manifold structure. A point patch is called manifold structured when every point pair has at least one geodesic path on the surface sampled by the point patch. For example, when the object is relatively thin compared with the radius of the point patch in 3-D, some of the resultant patches could consist of points which lie on the opposite side of the thin surface, as shown in Fig. 2(a). They are nonmanifold point patches, which will yield topologically incorrect folded polygons.

This problem can be resolved by using an alternative distance measure such as the geodesic distance. However, the geodesic distance is computationally costly. Even though a few fast algorithms exist, they are impractical for this application, due to the large size of the range data. In our approach, in order to handle this problem, a nonmanifold patch is subclustered by considering the normal direction of the points, yielding two manifold point patches, as shown in Fig. 2(b). If a new patch has few points, the relative size of it may be too small and it

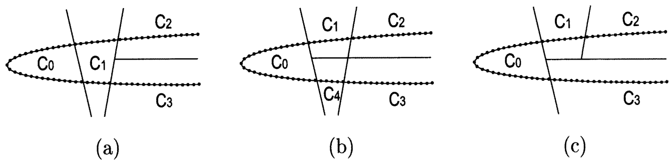
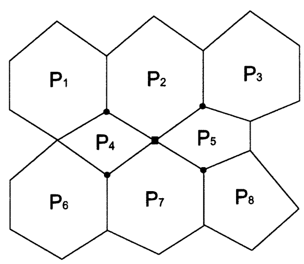


Fig. 2. Subclustering of a point patch in 2-D. (a) Non-manifold point patch $(\mathbf{C_1})$. (b) Removing nonmanifold patch by creating a new patch $(\mathbf{C_4})$. (c) Removing nonmanifold patch by neighborhood-reclustering ($\mathbf{C_3}$ and $\mathbf{C_4}$).



	P1	P2	P3	P4	P5	P6	P7	P8
P1	1	1	0	1	0	1	0	0
P2	1	1	1	1	1	0	1	0
P3	0	1	1	0	1	0	0	0
P4	1	1	0	1	1	1	1	0
P5	0	1	1	1	1	0	1	1
P6	1	0	0	1	0	1	1	0
P7	0	1	0	1	1	1	1	1
P8	0	0	0	0	1	0	1	1

(b)

Fig. 3. Example of the polygonal approximation. (a) Patch partitions and the vertices. (b) Corresponding PAT.

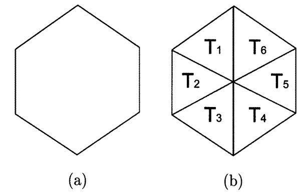


Fig. 4. Polygonal division. (a) An initial hexagonal patch. (b) Triangulation by division.

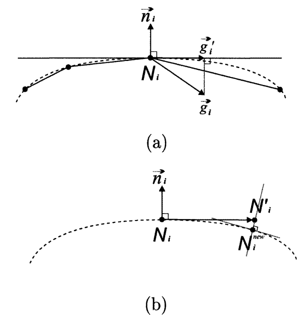


Fig. 5. Movement of a node. (a) Determine the tangential nodal force $\vec{\mathbf{g}}_i'$ first by projecting $\vec{\mathbf{g}}_i$ onto the tangent plane. (b) Then locate the new position $\boldsymbol{N}_i^{\text{new}}$ by projecting the tangential displacement on the sampled surface.

usually causes irregular topology in further processing. Thus, in this paper, we merge the smaller part [lower subcluster $\mathbf{C_4}$ in Fig. 2(b)] to the neighboring cluster as shown in Fig. 2(c).

IV. Triangular Mesh Construction

In this section, we describe the proposed algorithm to estimate the underlying topology by constructing the CTS triangular mesh from the point patches. For the generality, the algorithms is developed for a cloud of points, since a dense mesh structure is eventually a special case of a points cloud.

A. Polygonal Approximation of a Point Patch

Once the range data is clustered, each point patch is then approximated by a proper polygon based on the adjacency information of neighboring patches. This can be done easily, by constructing the patch adjacency table (PAT), which describes the adjacency relations between every pair of point patches.

Note that the PAT is a binary matrix, in which (i, j) element is 1, if and only if the point patches P_i and P_j are adjacent each other, and is 0 otherwise. The diagonal elements are set to be 1 by default. A simple example of a PAT is shown in Fig. 3. For the layout of the patches in Fig. 3(a), in which the point patch is depicted as a polygon for simplicity, patch adjacency is evaluated, yielding the PAT as shown in Fig. 3(b). Note that for the approximation of each point patch to a polygon, it is sufficient to determine the vertices of the polygon only. In our approach, the junction points where three or four patches meet together are considered as the vertices of the polygons. Five or more patches seldom meet together by the K-means clustering, which is also confirmed experimentally. In Fig. 3(a), the circular points denote the junction points of three adjacent patches, while the rectangular point denotes that of four. These junction points can be easily identified using the PAT. Assume that the point patches P_i, P_j, and P_k are adjacent each other, then nine elements in the i, j, kth rows and columns in the PAT would be 1. In this way, the adjacency of any three patches can be simply tested, by examining the corresponding 3×3 submatrix from the PAT. Then, the mid-point of the centroids of the patches P_i, P_j, and P_k, is computed and projected perpendicularly onto the sampled surface. The junction point is defined as the image of this projection. Junction points of four adjacent patches can be identified, similarly.

After determining the vertices using junctions, the polyhedral surface model is obtained by simply connecting the vertices. Note that the vertices of each polygon lie exactly on the sampled surface, due to the projection step. The vertices of a polygon may not be coplanar, but this does not cause any problem since the polyhedral model is an intermediate model which will be converted into a triangular mesh.

TABLE I
Computational Complexity

Processing Module	Complexity
Step1: Voronoi Partitioning	$O(n \cdot K)$
Step2: Triangular mesh generation	$O(\frac{n^2}{K})$
Step3: Mesh adaptation (for 1 iteration)	$O(K)$

B. Mesh Generation

The triangular mesh structure can be constructed by connecting the vertices of each polygon to the centroid of the corresponding point patch. A simple example is illustrated in Fig. 4. Note that both the vertices of each polygon and the centroid of the point patch lie on the sampled surface by projection.

C. Multiresolution Topology

Multiresolution description technique have been widely used to control the visual description in multiple levels of detail (LOD). In our proposed topology estimation approach, the multiresolution description can be easily obtained, by simply controlling the number of clusters, K in the K-means algorithm for the initial polyhedral model. In general, there exists a trade-off relationship between the resolution of modeling and the approximation error. By increasing or decreasing the number of polygonal patches, it is possible to control the resolution and approximation error.

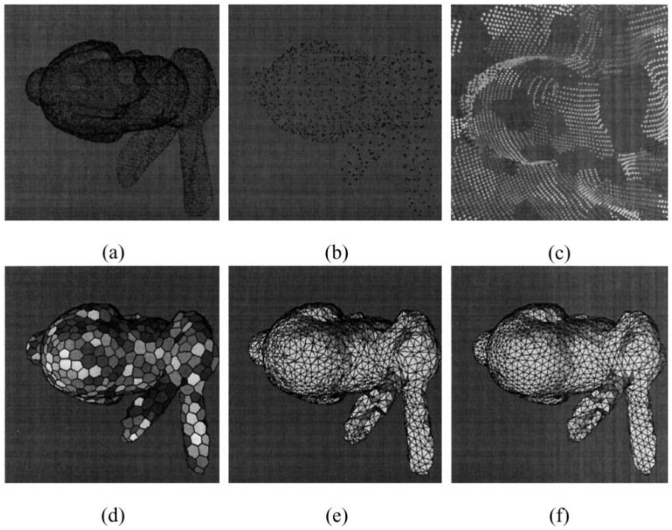


Fig. 6. Topology estimation result of the Bunny model. (a) Input range data $(\boldsymbol{N} = \mathbf{35947})$. (b) Codebook vector $(\boldsymbol{K} = \mathbf{800})$. (c) Point patches around the tail of the Bunny. (d) Polyhedral model. (e) Triangular mesh model before mesh adaptation. (f) Triangular mesh model after mesh adaptation.

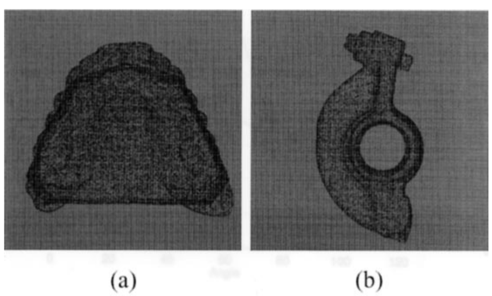


Fig. 7. Range data used in the experiments. (a) Teeth $(\boldsymbol{N} = \mathbf{58303})$. (b) Rocker arm $(\boldsymbol{N} = \mathbf{30132})$.

V. Mesh Adaptation Using Dynamic Model

Adaptive mesh has been proposed for nonuniform sampling and reconstruction of the intensity and range image data [4]. It is a dynamic model assembled as topologically regular collections of nodal masses connected by adjustable springs. The dynamic mesh automatically updates itself until it reaches the equilibrium state, driven by the nodal and data forces.

In our approach, we employ an adaptive dynamic mesh technique to improve the equiangular property 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 given by

$$\vec{\mathbf{s}}_{ij} = \frac{ce_{ij}}{\|\vec{\mathbf{r}}_{ij}\|}\vec{\mathbf{r}}_{ij} \tag{3}$$

where

$$\begin{aligned}\vec{\mathbf{r}}_{ij} &= \vec{\mathbf{x}}_j - \vec{\mathbf{x}}_i\\ e_{ij} &= \|\vec{\mathbf{r}}_{ij}\| - l_{ij}.\end{aligned} \tag{4}$$

In (4), $\vec{\mathbf{x}}_i$ and $\vec{\mathbf{x}}_j$ are the positional vectors 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 the following second-order nonlinear ordinary differential equation, given by

$$m\frac{d^2\vec{\mathbf{x}}_i}{dt^2} + \gamma\frac{d\vec{\mathbf{x}}_i}{dt} + \vec{\mathbf{g}}_i = \vec{\mathbf{f}}_i \tag{5}$$

$$\vec{\mathbf{g}}_i = \sum_j \vec{\mathbf{s}}_{ij} \tag{6}$$

where $\vec{\mathbf{f}}_i$ is the 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 (5) can be solved numerically using the Euler time-integration method [15], and the corresponding iterative equations can be derived as

$$\vec{\mathbf{f}}_i^{\text{new}} = \vec{\mathbf{f}}_i - \gamma\vec{\mathbf{v}}_i - \vec{\mathbf{g}}_i \tag{7}$$

$$\vec{\mathbf{a}}_i^{\text{new}} = \frac{\vec{\mathbf{f}}_i^{\text{new}}}{m} \tag{8}$$

$$\vec{\mathbf{v}}_i^{\text{new}} = \vec{\mathbf{v}}_i + \Delta t\vec{\mathbf{a}}_i^{\text{new}} \tag{9}$$

$$\vec{\mathbf{x}}_i^{\text{new}} = \vec{\mathbf{x}}_i + \Delta t\vec{\mathbf{v}}_i^{\text{new}} \tag{10}$$

where Δt is the time step. We observe that as Δt increases, the convergence rate is accelerated, however, the probability of converging to local minima is also increased. Using (10), the initial mesh is updated iteratively to be stabilized, with proper choice of m, γ, and Δt.

In order to improve the near-equiangular property 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_{ij}, 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_{ij} is not uniquely determined. Thus, to overcome this nonuniqueness problem, we define the mean of the two candidates as the natural length. Let S_1 and S_2 be the areas of two triangles which share a common spring. Then, the natural length l_{ij} becomes

$$l_{ij} = \frac{l_1 + l_2}{2} \tag{11}$$

where the length l_i is given by

$$l_k = \sqrt{\frac{4\sqrt{3}}{3}S_k}, \quad k = 1, 2. \tag{12}$$

Note that since usually the shapes of adjacent triangles are similar enough, the mean natural length, l_{ij}, of the shared spring can improve the near-equiangular property of both triangles simultaneously.

In general, since the approximation error is affected significantly by the nodal movement, modeling and determining the external force, i.e., data force $\vec{\mathbf{f}}_i$ in (5) is an another important issue in designing the 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 this problem, in our approach, we constrain the movement of each node to be only on the sampled surface, so that the modeling error, and thus the data force at each node become zero. In Fig. 5, a two-dimensional (2-D) 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 be $\vec{\mathbf{g}}_i$. Then, $\vec{\mathbf{g}}_i$ is projected onto the tangent plane, resulting in the tangential nodal force $\vec{\mathbf{g}}'_i$, which is the effective nodal force in this case. Now, as shown in Fig. 5(b), the node N_i is moved to the position of N'_i 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 onto it. Since the nodal position is tied on the sampled surface, the approximation error does not vary rapidly during the iteration process, while the near-equiangular property is significantly improved.

VI. Computational Complexity

Let n denote the number of points in the range data set, and K denote the number of clusters in the K-means clustering. Then, the com-

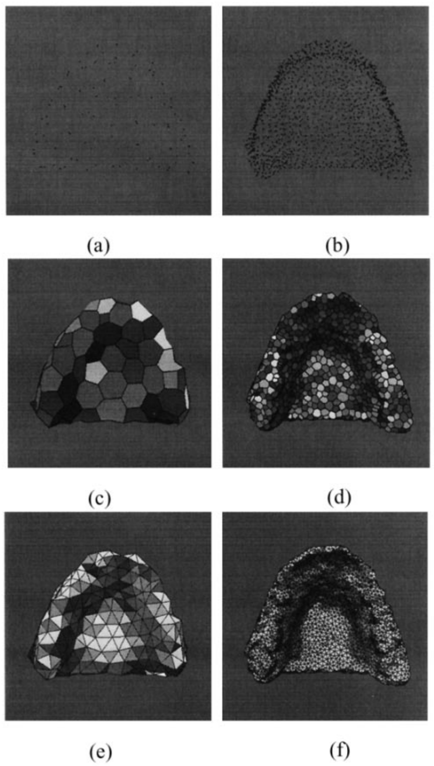


Fig. 8. Multiresolution topology estimation of the Teeth model. (a) Codebook vector $(\boldsymbol{K} = 100)$. (b) Codebook vector $(\boldsymbol{K} = 1500)$. (c) Polyhedral model $(\boldsymbol{K} = 100)$. (d) Polyhedral model $(\boldsymbol{K} = 1500)$. (e) Triangular mesh model $(\boldsymbol{K} = 100)$. (f) Triangular mesh model $(\boldsymbol{K} = 1500)$.

putational complexity at each step can be described as follows. First, during the K-means clustering step, the amount of computation is proportional to $n \cdot K$, since the distance to the cluster center should be computed from each data point. On the other hand, during the polygonal approximation and the triangular mesh generation step, the main portion of the computational complexity is devoted to the generation of the PAT. Thus, the required computation in this step is propositional to $\binom{n}{2}$ times the square of n/K, which is actually the mean number of the range data in each point patch. However, as described in Section IV, since $\binom{n}{2}$ can be reduced to n fortunately, it becomes proportional to n^2/K. Note that in the mesh adaptation procedure, the amount of computation is proportional to the number of the nodes, which is in turn proportional to the cluster number K. Table I shows the summary of the computational complexity of each step in our algorithm.

VII. Experimental Results

In order to evaluate the performance of the proposed algorithm, we perform the experiments on several range data sets. In Fig. 6, the result of topology estimation is illustrated for the Bunny model. Fig. 6(a) shows the input range data. For the codebook vectors in Fig. 6(b), the range data is partitioned into point patches as shown in Fig. 6(c). Each point patch is then approximated by a polygon, yielding polyhedral model as shown in Fig. 6(d). Note that, if the range data is distributed uniformly on the object surface and the curvature of object surface is constant, then the polygons in the polyhedral model would be a perfect hexagon. However, in real world, the constructed polygons are observed to be from quadrilaterals to octagons, since the density of the range data and the local shape is indeed not uniform. Fig. 6(e) shows the triangular mesh model, which is generated from the polyhedral model.

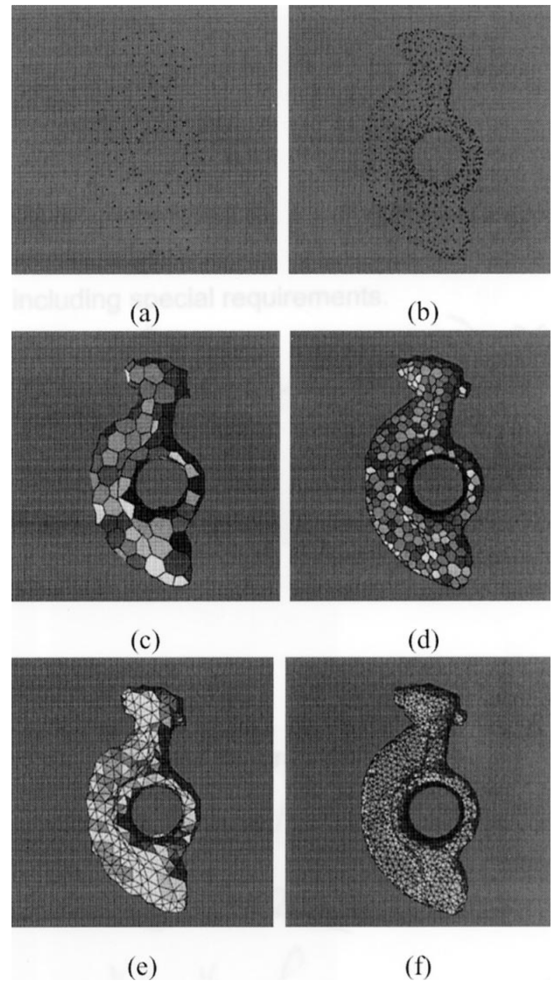


Fig. 9. Multiresolution topology estimation of the Rocker Arm model. (a) Codebook vector $(\boldsymbol{K} = 200)$. (b) Codebook vector $(\boldsymbol{K} = 1000)$. (c) Polyhedral model $(\boldsymbol{K} = 200)$. (d) Polyhedral model $(\boldsymbol{K} = 1000)$. (e) Triangular mesh model $(\boldsymbol{K} = 200)$. (f) Triangular mesh model $(\boldsymbol{K} = 1000)$.

As the iteration continues in the mesh adaptation stage, the mesh converges to a steady state. In our experiments, 50 iterations are enough for the convergence. Note that as the mesh converges to a steady state, the distribution of the inner angle of triangle is getting narrow-banded, centered at 60°. It is confirmed visually in Fig. 6(f), in which the triangles are much more equiangular than triangles in Fig. 6(e). Narrow and sharp triangles are seldom observed in the final mesh.

Next, multiresolution topology estimation is performed for the models in Fig. 7, while polyhedral and triangular mesh with different choices of K are constructed. The results are shown in Fig. 8 and Fig. 9, respectively. It is observed that the triangles on the reconstructed meshes are quite equiangular, because of the mesh adaptation. Fig. 10(a) and (b) show the histogram changes of those models before and after the mesh adaptation. Each initial mesh is updated iteratively for 50 times. It is observed that the equiangular property is significantly improved, and thus the resultant triangular meshes look quite regular.

In order to examine the modeling error for each test, the distance from the original range data to the reconstructed triangular mesh model is computed, followed by a normalization with respect to the diagonal length of the bounding box of the object [14]. The modeling errors of Teeth and Rocker arm model are summarized in Table II, in which the distance is measured using signed/unsigned L^1 and L^∞ norm. It is

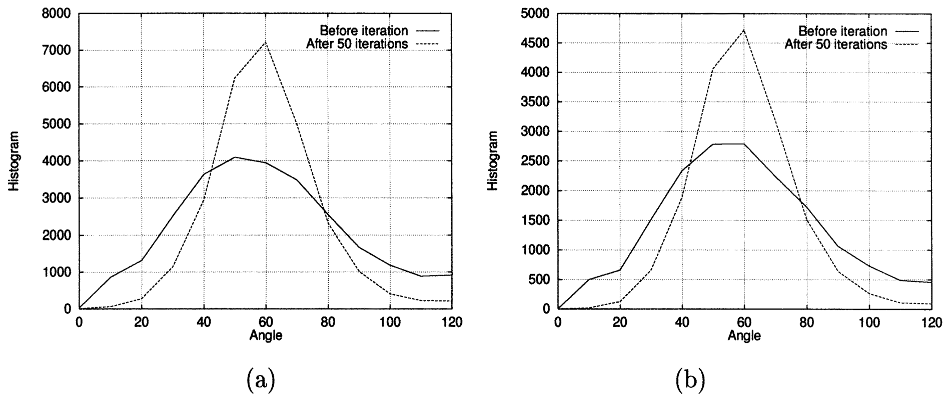


Fig. 10. Angle histogram before and after mesh adaptation. $m = 0.01, \gamma = 0.2, c = 1, \Delta t = 0.05$. (a) Histogram variation of Teeth model ($K = 1500$). (b) Rocker arm model ($K = 1000$).

TABLE II
APPROXIMATION ERROR AND EXECUTION TIME ON PENTIUM III 933 MHZ (T: NUMBER OF TRIANGLE, V: NUMBER OF NODE, E: NUMBER OF EDGE, SiAD: SIGNED AVERAGE DISTANCE, USiAD: UNSIGNED AVERAGE DISTANCE, STD: STANDARD DEVIATION, MAX: MAXIMUM DISTANCE)

Model	K	Mesh Statistics			Approximation Error (in %)				Execution Time (in Millisecond)		
		T	V	E	SiAD	USiAD	STD	MAX	Voronoi Partitioning	Mesh Generation	Adaptation
Teeth	100	600	302	900	-0.3627	1.5017	2.0328	7.6922	1,252	1,662	1,672
	1,500	9,016	4,508	13,522	0.0654	0.3753	0.7770	7.6991	10,896	4,166	19,608
Rocker	200	1,206	603	1,809	-0.5466	0.9328	1.2490	8.5700	991	541	2,674
Arm	1,000	5,748	2,874	8.622	-0.0557	0.2708	0.4229	8.5567	3,825	1,522	12,508

observed that the modeling errors are quite small, even for the coarsest models. Also, note that since the output mesh is CTS, the numbers of vertices (V), edges (E), triangles (T), and hole (H) for each case satisfy the Euler equation, given by

$$V - E + T + 2H = 2. \tag{13}$$

The computational complexity of the proposed algorithm is illustrated also in Table II. It is observed that the complexity is quite tolerable. The most complex model requires less than 35 seconds, assuming that the codebook vectors are generated offline, which can be done in the range data merging procedure.

VIII. CONCLUSION

In this paper, based on a combined statistical and dynamical method, we proposed an efficient algorithm to estimate the underlying topology of 3-D range data in the form of near-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 alleviate the computational cost required in modeling, especially for large and dense data set.

REFERENCES

[1] I. Porteous, *Topological Geometry*. New York: Van Nostrand, 1969.

[2] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantization design," *IEEE Trans. Commun.*, vol. 28, pp. 84–95, Jan. 1980.

[3] L. Floriani, B. Falcidieno, and C. Pienovi, "Delaunay-based representation of surfaces defined over arbitrarily shaped domains," *Comput. Vis., Graph., Image Process.*, vol. 32, no. 1, pp. 127–140, 1985.

[4] D. Terzopoulos and M. Vasilescu, "Sampling and reconstruction with adaptive mesh," *Proc. IEEE Conf. Computer Vision Pattern Recognition*, pp. 70–75, June 1991.

[5] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, "Surface reconstruction from unorganized points," in *Proc. SIGGRAPH'92*, July 1992, pp. 71–78.

[6] J. L. Marroquin and F. Girosi, "Some Extensions of the K-Means Algorithm for Image Segmentation and Pattern Recognition," MIT Artific. Intell. Lab., AI Memo Rep. 1930, Jan. 1993.

[7] T. Fang and L. Piegl, "Delaunay triangulation in three dimensions," *IEEE Comput. Graph. Appli.*, vol. 15, pp. 62–69, Sept. 1995.

[8] M. Soucy and D. Laurendeau, "Multiresolution surface modeling based on hierarchical triangulation," *CVGIP: Image Understand.*, vol. 63, no. 1, pp. 1–14, Jan. 1996.

[9] R. Cignoni, C. Montani, and R. Scopigno, "A comparizon of mesh simplification algorithms," *Computers & Graphics*, vol. 22, no. 1, pp. 37–54, Feb. 1998.

[10] N. Amenta, M. Bern, and M. Kamvysselis, "A new Voronoi-based surface reconstruction algorithm," in *Proc. SIGGRAPH'98*, July 1998, pp. 415–421.

[11] N. Amenta and M. Bern, "Surface reconstruction by Voronoi filtering," *Discrete Comput. Geometry*, vol. 22, pp. 481–504, 1999.

[12] N. Amenta, S. Choi, T. Dey, and N. Leekha, "A simple algorithm for homeomorphic surface reconstruction," in *Proc. ACM Symp. Computational Geometry*, June 2000, pp. 213–222.

[13] J. Boissonnat and F. Cazals, "Smooth surface reconstruction via natural neighbor interpolation of distance functions," in *Proc. ACM Symp. Computational Geometry*, June 2000, pp. 223–232.

[14] I. K. Park, K. M. Lee, and S. U. Lee, "Efficient measurement of shape dissimilarity between 3-D models using Z-buffer and surface roving method," *EURASIP J. Applied Signal Processing*, vol. 2002, no. 10, pp. 1127–1134, Oct. 2002.

[15] W. Press, B. Fannery, S. Teukosky, and W. Vetterling, *Numerical Recipes: The Art of Scientific Computing*. Cambridge, MA, UK: Cambridge Univ. Press, 1986.