Automatic 3-D Model Synthesis from Measured Range Data

In Kyu Park, *Student Member, IEEE,* Il Dong Yun, *Member, IEEE,* and Sang Uk Lee, *Senior Member, IEEE*

***Abstract*—In this paper, we propose an algorithm to construct 3-D surface model from a set of range data, based on non-uniform rational B-splines (NURBS) surface-fitting technique. It is assumed that the range data is initially unorganized and scattered 3-D points, while their connectivity is also unknown. The proposed algorithm consists of three stages: initial model approximation employing K-means clustering, hierarchical decomposition of the initial model, and construction of NURBS surface patch network. The initial model is approximated by both polyhedral and triangular model. Then, the initial model is represented by a hierarchical graph, which is efficiently used to construct the G^1 continuous NURBS patch network of the whole object. Experiments are carried out on synthetic and real range data to evaluate the performance of the proposed algorithm. It is shown that the initial model as well as the NURBS patch network are constructed automatically with tolerable computation. The modeling error of the NURBS model is reduced to 10%, compared with the initial mesh model.**

***Index Terms*—Hierarchical decomposition, K-means clustering, NURBS, range data, 3-D surface model.**

I. Introduction

SINCE it provides explicit geometrical information on the surface of an underlying object, range data plays an important role in 3-D modeling. 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 [1], [2], [18], multiple range data, obtained from different views, can be transformed into a common coordinates system, so that complete and seamless 3-D range data for 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 should be obtained for further high-level processing. Thus, the modeling technique to convert the raw range data into a suitable surface model is quite of important issue, and much efforts have been made to develop such techniques so far [1]–[5], [7], [9], [12]. Applications of this technique can be found in the field of 3-D modeling, including reverse engineering and virtual environments, as well as 3-D video acquisition and display. Note that in the field of 3-D video technology [21], [22], the representation of the 3-D model is quite an important issue for data compression and transmission.

Among several 3-D models, the triangular mesh model has been commonly used in many previous works [1]–[3], [7], [9], [10]–[12]. Since the range image is acquired in a grid array, the mesh model can be obtained by utilizing the connectivity information. The triangular mesh model has been the most popular 3-D model in the fields of computer vision and graphics, as the triangular mesh model is easy to construct and manipulate. In addition, multiresolution modeling is possible, by deleting nodes, swapping edges, merging triangles, etc. However, since the amount of range data for 3-D object is tremendous, it requires a heavy computation and additional post-processing procedure, such as mesh optimization [8]. Moreover, the range data itself suffers from the residual noise, which is inherent during acquisition, registration, and integration, making the resultant triangular mesh also apt to be irregular. Therefore, to overcome the disadvantages of triangular mesh model, B-spline surface models recently receive much attention for modeling 3-D object [4]–[6], [13], [14].

There are several advantages of the non-uniform rational B-splines (NURBS) surface for reconstructing the 3-D shape. First, in the fields of computer-aided design and computer graphics, a general class of B-spline, i.e., the NURBS, is adopted for a standard representation of curves and surfaces [20]. The NURBS can be easily found in standard data formats, such as Initial Graphics Exchange Specification (IGES) and Standard for the Exchange of Product (STEP). We also believe that, in the near future, the NURBS will be very popular for 3-D video synthesis and display technology. Another advantage is that NURBS is mathematically defined, therefore it is possible to obtain analytic surface model from the range data. Furthermore, the NURBS surface model is known to be robust to the noise inherently residing in the range data, since the noise affects only the local area of the whole surface.

In this paper, we propose a new efficient algorithm to construct the 3-D surface model from measured range data, based on the NURBS surface-fitting technique. In the proposed algorithm, it is assumed that the input range data is scattered and unorganized 3-D points cloud, which is the most primitive format in real world. The proposed algorithm consists of three steps, including initial model approximation, hierarchical decomposition of the initial model, and construction of the NURBS surface patch network. One of the most difficult problems in surface modeling, based on the NURBS, is that the input data should be mapped on a regular grid structure. However, since a single NURBS patch can only represent surfaces of simple topological

Manuscript received March 15, 1999; revised September 30, 1999. This paper was recommended by Guest Editor K. N. Ngan

I. K. Park and S. U. Lee are with the Signal Processing Laboratory, School of Electrical Engineering, Seoul National University, 151-742 Seoul, Korea (e-mail: pik@sting.snu.ac.kr; sanguk@sting.snu.ac.kr).

I. D. Yun is with the Department of Control and Instrumentation Engineering, Hankuk University of F. S., 449-791 Yongin, Korea (e-mail: yun@computer.org).

Publisher Item Identifier S 1051-8215(00)02016-4.

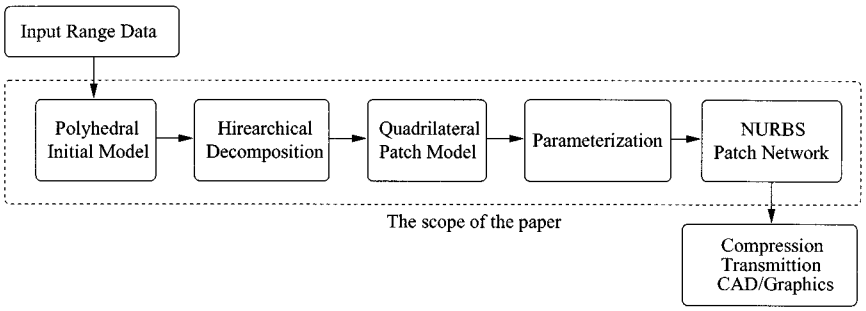


Fig. 1. Overview of the modeling algorithm.

type, a surface of arbitrary topology should be defined as a network of the NURBS patches. It is a difficult task to construct both a network of patches and a parameterization of the range data over these patches automatically. The main contribution of our work is that we propose an efficient method for constructing the network and performing the parameterization automatically.

In our approach, the range data is first segmented into a finite number of point patches, using the K-means clustering algorithm [16]. Each point patch is then approximated by an appropriate polygon and divided into triangles, yielding a triangular mesh for an initial model. Before obtaining the NURBS patch network, the polyhedral initial model is represented by an adjacency graph, which is then partitioned into several cooperative subgraphs. The quadrilateral domain for the NURBS patch network is constructed in each subgraph. In this way, the initial model is interpreted as a hierarchical tree structure: adjacency graph—subgraphs—polygonal nodes—triangles, in which the hierarchy is efficiently utilized to alleviate the computational complexity. Finally, the NURBS surface fitting is performed on each quadrilateral domain, yielding the NURBS patch network, which is tangent plane (G^1) and curvature (C^2) continuous on its boundary and the inner region of each patch, respectively.

This paper is organized as follows. In Section II, a brief survey of the previous work is presented. Next, an algorithm for the initial model approximation is described in Section III. In Section IV, the proposed hierarchical decomposition is introduced. In Section V, we describe an algorithm for generating the NURBS patch network in detail. The experimental results are provided in Section VI. Finally, in Section VII, we present the conclusions. The overall block diagram for the proposed algorithm is shown in Fig. 1.

II. Previous Work

Surface-modeling techniques, based on the NURBS fitting, are roughly classified as semi-automated and automated, depending on the amount of manual operation by users. They can also be categorized, depending on whether a single open NURBS patch or NURBS patch network is used. Note that single NURBS patch can only represent a simple topological type, such as an ellipsoid.

Semi-automated techniques [13], [14] first attempt to identify a subset of points that are to be approximated. Parameterization of data points is usually accomplished by a user-guided procedure, such as the projection of the points onto a manually constructed base plane or surface, followed by generating the NURBS surface. Since this technique is only capable of approximating the local area of the object, it is difficult to obtain the entire or complete surface model for arbitrary shaped object.

In contrast, an automated technique [4], [5] is normally combined with a technique for constructing the NURBS patch network to separately perform the parameterization. Eck and Hoppe [4] proposed an algorithm to construct the NURBS patch network satisfying a prespecified error bound. While [4] generates high-quality surfaces, it requires a number of optimization steps, making it computationally expensive. Krishnamurthy and Levoy [5] proposed a technique, based on dense triangular mesh, to interactively segment the input range data into a number of regions that are to be approximated by the NURBS patches. However, [5] requires manual operations, and thus, it cannot be considered a full automated algorithm.

Recently, 3-D video coding attracts considerable attention as a promising alternative to the conventional 2-D video coding. Malassiotis and Strintzis [21] proposed an object-based coding algorithm, using intensity and $2(1/2)$-D depth image, acquired from stereo image pair. Note that a detailed review on the 3-D video synthesis and visual communication can be found in [22], in which several key techniques are discussed from both computer vision and image coding point of views.

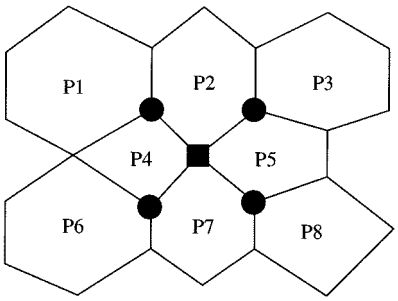
III. Initial Model Approximation

After registering and integrating each range view, the range data, forming complete 3-D surface of an object, can be obtained in a common coordinate. In this section, we describe the proposed algorithm to construct the initial triangular mesh from the range data.

A. Polyhedral Initial Model-Approximation

Initial model-approximation begins with partitioning the range data, followed by the polygonal approximation. The detailed algorithm can be found in [17], which can be summarized as follows.

The sampled surface is segmented into finite number of point patches by using the K-means clustering technique. After clustering into K point patches, based on the adjacency to the neighboring patches, each point patch is then approximated by a proper polygon, which can be done easily by constructing the Patch Adjacency Table (PAT). Notice that the PAT describes the adjacency relation between every pair of point patches,



(a)

	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. 2. Example of the polygonal approximation. (a) Point patch layout. (b) Corresponding PAT of (a).

represented by a binary adjacency matrix. In Fig. 2, an example of PAT is shown, in which $P_0 - P_8$ denote the point patches. In order to approximate each point patch by a polygon, it is sufficient to determine the vertices of the polygon. In our approach, the junction points where three or four patches meet together are considered as the vertices of the polygons. Assume that the point patches P_i, P_j, and P_k are adjacent each other, then nine elements in i, j, kth row and column in the PAT are 1. In this way, the adjacency of any three patches can be simply tested by examining the 3×3 matrix, which is augmented from the PAT. Also, it can be tested in the same way, when four point patches are adjacent each other. By approximating each point patch with a polygon, a polyhedral initial model is obtained, which topologically approximates the sampled surface. In the K-means clustering, the normal vector of range data is also examined, preventing each side of thin surface from being clustered in a same point patch. Note that the normal of range data is easily obtained in the acquisition and registration stages.

B. Triangular Mesh Generation

The triangular mesh structure is constructed by connecting the vertices and the centroid of the polygonal facet. Note that the vertices of polygon and the centroid of the point patch are projected on the locally approximated surface before division, forcing the nodes of the triangular mesh located almost exactly on the true surface.

Since we deal with 3-D data which is integrated from multiple range views, it is reasonable to assume that the constructed initial mesh model represents a closed object. Under this assumption, it is worthy to note that the number of triangles in the mesh N is always even. From the observation that three edges form a triangle and two triangles share an edge, the total number of edges in the triangular mesh is $3N/2$. However, since $3N/2$ should be an integer, N is an even number. In addition, the number of odd-sided polygons in the polyhedral model is also even. Since the sum of the number of triangles which are mapped to odd-sided polygon should be even, the number of odd-sided polygon is also even.

Note that the discussed properties can be effectively used for constructing the parameter space for the NURBS network generation, which will be presented in Sections IV and V. That is, since the NURBS surface fits the grid-structured point array, a $u - v$ parameter space is preferred to the parameterization of the input data, which can be obtained by simply merging a pair of triangles. Therefore, the total number of triangles should be even, so that we can employ the quadrilateral model and define the NURBS patch network.

IV. Constructing Parameter Space Using Hierarchical Decomposition

In order to build the NURBS model from 3-D points set, the input data should be reorganized as a mapping from $u-v$ parameter space. Therefore, the $u-v$ parameter space should be constructed before such a mapping is considered. Since it is almost impossible to find the parameter space from the input data or the initial model directly, an intermediate representation is required for further processing. In our approach, a quadrilateral patch model is intermediately built, by hierarchically decomposing the initial model. The constructed quadrilateral patch network is efficiently used for the parameter space for the NURBS surface fitting, as described in Section V.

As we mentioned previously, the triangular mesh is made up of even number of triangles, which is the necessary condition for the existence of the quadrilateral patch network. With this condition satisfied, the basic notion behind obtaining a quadrilateral patch is to group a pair of adjacent triangles together. Even though it is conceptually simple, a brute-force search method leads to an NP-complete problem, unfortunately, since there exist too many triangles in the mesh. Therefore, a more sophisticated approach is necessary to perform the overall pairwise grouping with tolerable computational complexity.

A. Hierarchical Decomposition

The adjacency graph is generated from the initial polyhedral model, in which each node denotes the individual polygonal facet. The adjacency graph forms the top level of the hierarchy. In the adjacency graph, the nodal attribute is determined according to the type of the polygon, i.e., even-sided and odd-sided. Then, the adjacency graph is divided into a finite number of subgraphs, depending on the nodal attribute. The subgraph is defined as the cluster of the connected nodes with same

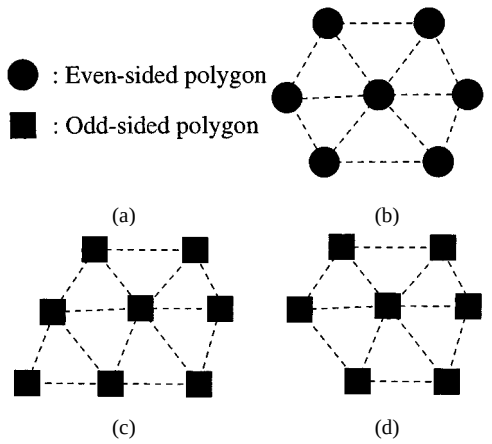


Fig. 3. Three types of subgraph. (a) Nodal attributes. (b) Type I subgraph. (c) Type II subgraph (cardinality is 8). (d) Type III subgraph (cardinality is 7).

attribute. Depending on the nodal attribute and the cardinality[1], the generated subgraph can be classified into one of three types defined as follows.

Type I : Nodal attribute is even-sided polygon, regardless of the cardinality.

Type II : Nodal attribute is odd-sided polygon, while the cardinality is even.

Type III : Nodal attribute is odd-sided polygon, while the cardinality is odd.

In Fig. 3, an example of three types of subgraph is illustrated, in which rectangles and circles indicate odd-sided and even-sided polygon, respectively. Each subgraph consists of a set of polygons, representing the third level decomposition. Also, note that each node in the graph can be subdivided into triangles, yielding the lowest level in the hierarchical structure. The hierarchical tree structure is shown in Fig. 4.

B. Quadrilateral Model Construction

Based on the tree structure, the quadrilateral model is constructed at the lowest (triangular mesh) level by merging two neighboring triangles. A basic notion behind merging the triangle for both types is illustrated in Fig. 5. For an even-sided polygon, a pair of triangles can be merged inside the polygon, since pairwise grouping is always possible. A more complicated case is the odd-sided polygon, since there always exists a remaining triangle after grouping. In this case, two adjacent polygons are considered together. That is, as shown in Fig. 5(d), the neighboring triangles are merged first, yielding quadrilateral Q4. Then, the remaining triangles are grouped pairwise, yielding quadrilaterals Q1, Q2, Q4, Q5, and Q6.

However, in general, notice that the solution for pairwise grouping of odd-sided polygons does not always exist. For instance, consider the case of odd-sided polygonal facet which is surrounded by even-sided polygons. Therefore, in our approach, the quadrilateral model is generated after further processing of each type of the subgraph.

[1]The cardinality of a graph is defined as the number of nodes.

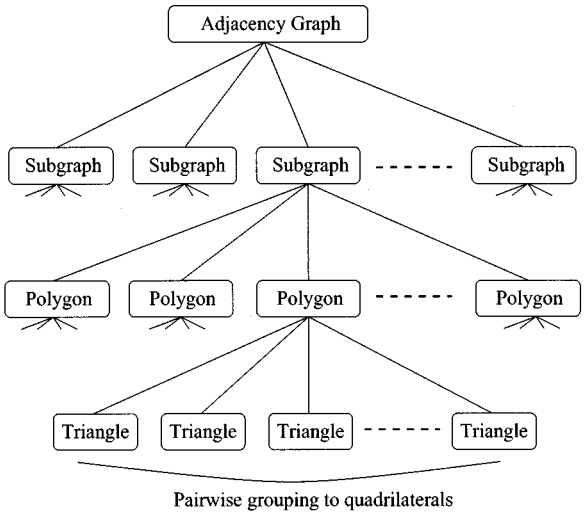


Fig. 4. Hierarchical decomposition of the initial polyhedral model.

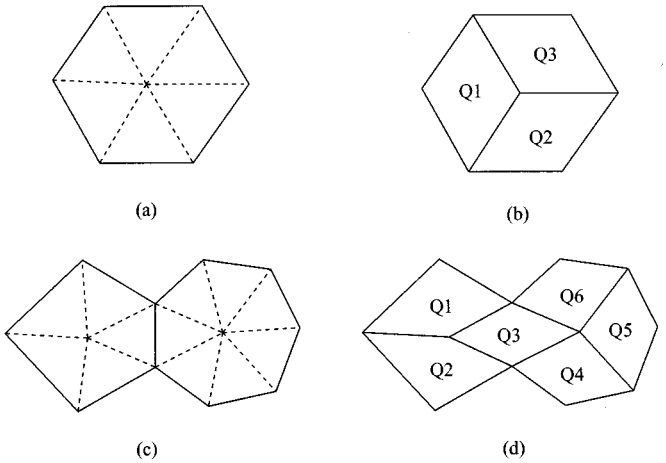


Fig. 5. Generating quadrilaterals under the polygonal constraint. (a) An even-sided polygon. (b) Resultant quadrilaterals from (a). (c) Odd-sided polygons neighboring each other. (d) Resultant quadrilaterals from (c).

C. Processing of Type I and II Subgraph

Type I is the most simple case. Each polygon can be individually converted into quadrilaterals in the same way as shown in Fig. 5(b). Note that some nodes in the Type I subgraph help process the Type III subgraph, which will be described later.

On the other hand, since the polygons are odd-sided, simple method for Type I subgraph does not work for the Type II subgraph. In this case, pairwise grouping of neighboring polygons is performed beforehand, which is always possible since the cardinality of Type II subgraph is even. Then, each pair of polygons is simultaneously converted to quadrilaterals as shown in Fig. 5(d).

D. Processing of Type III Subgraph

This is the most complicated case. Since the cardinality is odd, the pairwise grouping is not possible. In order to solve the problem, in our approach, we merge a pair of Type III subgraphs. Fortunately, it can be inferred that the number of the Type III subgraph is always even, since the number of odd-sided polygon is even.

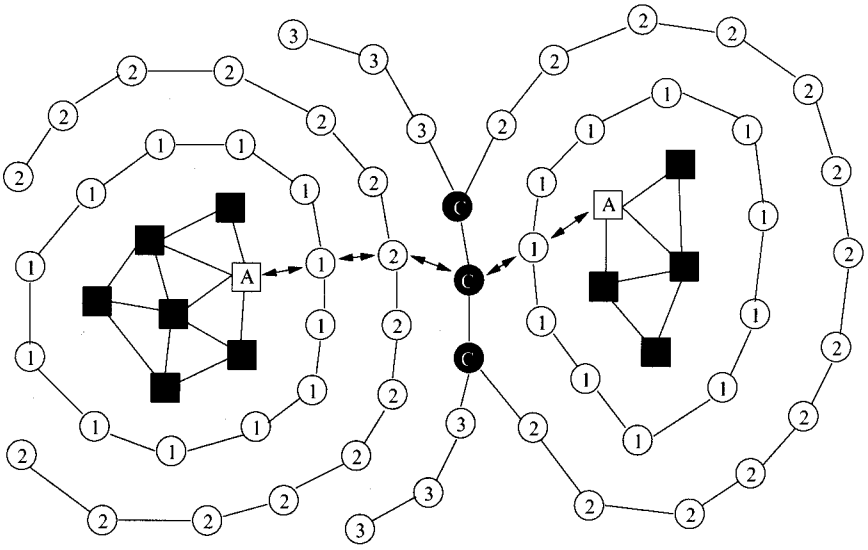


Fig. 6. Finding the shortest path between Type III subgraphs. The number indicates the level of expansion. The nodes with lable "A" denote the anchor node of each subgraph.

Based on this notion, attempts are made to merge a pair of Type III subgraphs, in which merging is performed by connecting the subgraphs with the path which lies between them. In order to find the path efficiently, each subgraph is incrementally expanded, by finding the layer of the neighboring nodes, which are actually inside Type I subgraph. The procedure is shown in Fig. 6, in which the Type III subgraphs are represented by the connected nodes, specified by rectangle. Note that the cardinalities are 7 and 5. The nodes in Type III subgraph itself are given the label of level 0, then the obtained neighbors are labeled according to the level at which they are layered, as shown in Fig. 6. As the expansion continues, the expanded subgraphs should meet at some common nodes (specified by the label 'C'). By obtaining the common nodes, the shortest path can be found, which begins and ends at the anchor node of each subgraph. The obtained shortest path is specified by the arrows in Fig. 6.

The quadrilateral model is constructed in the path nodes and the subgraphs. In the path nodes, pairwise grouping of triangles is performed by sequentially merging the neighboring triangles and then merging the remaining triangles inside each polygon, as shown in Fig. 7(a). In Fig. 7(b), the obtained quadrilaterals are specified. On the other hand, since the Type III subgraphs have lost a node, i.e., the anchor nodes, they are already converted to the Type II. For example, in Fig. 6, the cardinality of the left side subgraph is reduced to 6, after losing the anchor node. Therefore, by applying the same method for the Type II subgraph, the quadrilateral model is finally obtained.

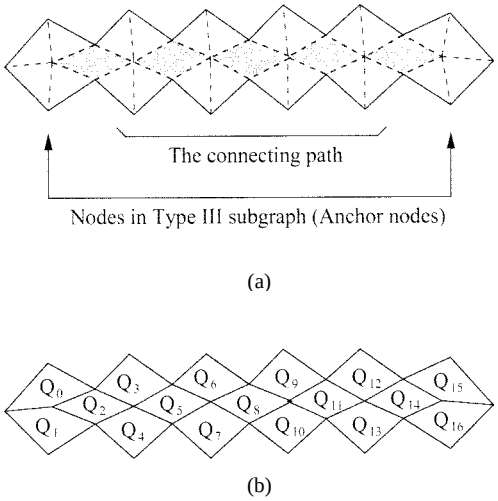


Fig. 7. Quadrilateral patch generation in the connection nodes. (a) Initial layout. (b) Quadrilateral patches.

V. NURBS Patch Network Generation

In this section, based on the quadrilateral patch model, the NURBS surface fitting is performed. Since the NURBS surface fitting requires that the input data should be arranged in a grid array structure, parameterization of range data and construction of regular grid structure are performed before the surface fitting. In order to obtain the smooth and accurate NURBS surface model, the regular grid structure should precisely approximate the range data as mush as possible.

A. Parameterization of Range Data

The range data, which is unorganized and scattered, is now parameterized on the parameter space. The parameterization is performed by projecting each point onto the quadrilateral patch. Note that each vertex of a quadrilateral patch corresponds the coordinate $(0,0)$, $(0,1)$, $(1,0)$, and $(1,1)$ of parameter space. The u, v parameter of the point is then determined by the coordinate on the quadrilateral domain.

B. Construction of Regular Grid Structure on Each Quadrilateral Patch

Based on the parameterization, a regular grid patch is constructed, in order to be used for the input to the NURBS surface fitting. In our approach, the initial quadrilateral is refined to a grid array hierarchically, approximating the parameterized

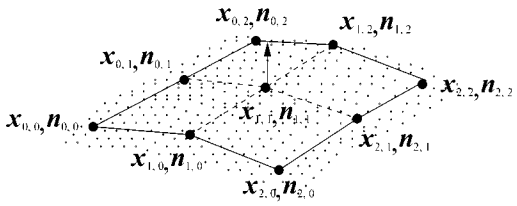


Fig. 8. Determining position and normal vector of new grid points during recursive division.

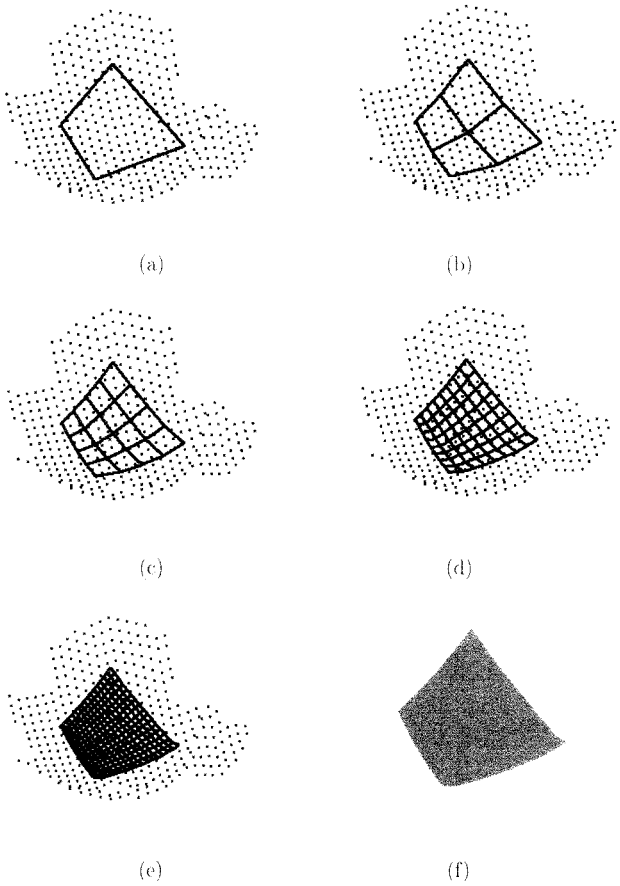


Fig. 9. Constructing a regular grid structure from an initial quadrilateral patch. (a) Initial quadrilateral patch. (b) 3 × 3 grid. (c) 5 × 5 grid. (d) 9 × 9 grid. (e) 17 × 17 grid. (f) Reconstructed NURBS surface [from (e)].

range data, while increasing the resolution. The basic notion behind the recursive division is illustrated in Fig. 8, in which $\mathbf{x}_{i,j}$ and $\mathbf{n}_{i,j}$ denote the position and the normal vector, respectively. The initial quadrilateral consists of four vertices $\{\mathbf{x}_{0,0}, \mathbf{x}_{0,2}, \mathbf{x}_{2,0}, \mathbf{x}_{2,2}\}$, which is then divided into 3 × 3 grid. New grid points are obtained in the following way. The middle points on the border are first computed, by averaging the neighbor grid points. In a similar manner, the normal vectors of the points are computed. In order to locate the new point on the sampled surface, the new points are projected onto the locally approximated surface in the normal direction. The projected points are then considered as the new grid points $\{\mathbf{x}_{1,0}, \mathbf{x}_{0,1}, \mathbf{x}_{2,1}, \mathbf{x}_{1,2}\}$. Finally, the neighboring range data of the new grid points is collected to compute the new unit normal vector. The inner grid point $\mathbf{x}_{1,1}$ can be obtained in the same way, using the 4-connectivity neighboring points $\{\mathbf{x}_{1,0}, \mathbf{x}_{0,1}, \mathbf{x}_{2,1}, \mathbf{x}_{1,2}\}$.

By recursively applying the division procedure, we can obtain grid structure of 5 × 5, 9 × 9, and so on. In Fig. 9, the procedure for obtaining 17 × 17 grid from the initial quadrilateral is shown, in which each point denotes the approximated range data points. Note that since the new grid points are anchored on the locally approximated surface, the generated grid structure approximates the sampled surface accurately, forming regularly spaced mesh. After generating the grid structure for all the patch on the network, the NURBS surface fitting is then followed.

TABLE I
LIST OF SYMBOLS USED IN THE NURBS SURFACE FITTING

Symbol	Description
$N_{i,p}(u)$	NURBS basis function in u direction
$N_{j,q}(v)$	NURBS basis function in v direction
$\mathbf{b}_{i,j}$	Control point
n+1	Size of control point mesh in u direction
m+1	Size of control point mesh in v direction
p	Order of the NURBS surface in u direction
q	Order of the NURBS surface in v direction
U	Knot vector in u direction
V	Knot vector in v direction
r+1	Number of elements in U
s+1	Number of elements in V
$\mathbf{Q}_{k,l}$	A point in the input grid array
N+1	Size of input grid array in u direction
M+1	Size of input grid array in v direction
$\overline{u}_k$	Estimated parameter at $\mathbf{Q}_{k,l}$ in u direction
$\overline{v}_l$	Estimated parameter at $\mathbf{Q}_{k,l}$ in v direction

C. NURBS Surface Fitting

The parameters of NURBS surface consist of control points, weight, knot vector, and the basis function. The mathematical definition of NURBS surface is given in (1), in which p and q are the order of the surface in u, v direction, respectively

$$\mathbf{S}(u,v) = \frac{\sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u)N_{j,q}(v)w_{i,j}\mathbf{b}_{i,j}}{\sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u)N_{j,q}(v)w_{i,j}}, \quad 0 \le u, v \le 1 \quad (1)$$

where

$$U = \{u_0, u_1, \ldots, u_r\}$$
$$V = \{v_0, v_1, \cdots, v_s\}$$

In Table I, the symbols used in the NURBS surface fitting are listed.

The proposed NURBS surface fitting is performed on each individual patch in NURBS network as follows. For given input grid array $\mathbf{Q}_{k,l}$ $(k = 0, 1, \cdots, N;\ l = 0, 1, \cdots, M)$, the NURBS surface-fitting procedure is essentially to find the control mesh $\mathbf{b}_{i,j}$ and the knot vector U, V. Note that once the control mesh and the knot vectors are obtained, the whole

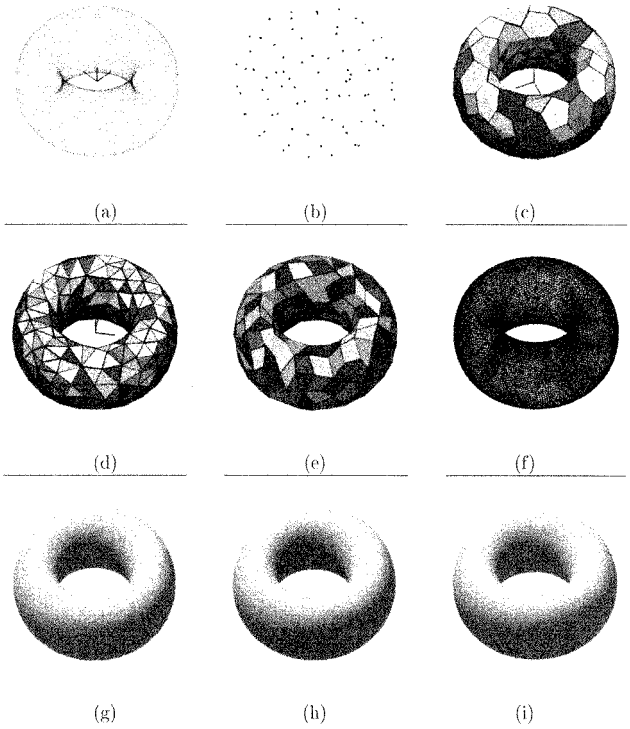


Fig. 10. Modeling result of torus data ($K = 80$). (a) Range data. (b) Centroids of the point patches. (c) Polyhedral initial model. (d) Triangular initial model. (e) Quadrilateral patch model (NURBS parameter space). (f) NURBS control mesh ($n = m = 8$). (g) Reconstructed NURBS surface ($n = m = 4, p = q = 2$). (h) Reconstructed NURBS surface ($n = m = 8, p = q = 2$). (i) Reconstructed NURBS surface ($n = m = 16, p = q = 2$).

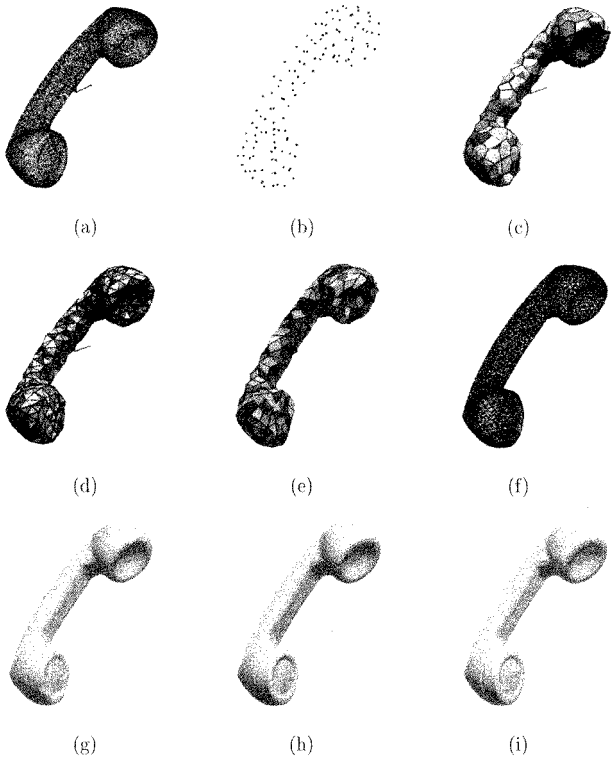


Fig. 11. Modeling result of phone data ($K = 130$). (a) Range data. (b) Centroids of the point patches. (c) Polyhedral initial model. (d) Triangular initial model. (e) Quadrilateral patch model (NURBS parameter space). (f) NURBS control mesh ($n = m = 8$). (g) Reconstructed NURBS surface ($n = m = 4, p = q = 2$). (h) Reconstructed NURBS surface ($n = m = 8, p = q = 2$). (i) Reconstructed NURBS surface ($n = m = 16, p = q = 2$).

surface patch can be generated, since it is possible to calculate every surface point at $\mathbf{S}(u, v)$ from (1).

From the definition in (1), the following relation holds between $\mathbf{Q}_{k,l}$ and $\mathbf{b}_{i,j}$, given by

$$\mathbf{Q}_{k,l} = \mathbf{S}(\bar{u}_k, \bar{v}_l) = \frac{\sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(\bar{u}_k) N_{j,q}(\bar{v}_l) w_{i,j} \mathbf{b}_{i,j}}{\sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(\bar{u}_k) N_{j,q}(\bar{v}_l) w_{i,j}} \tag{2}$$

where $\bar{u}_k$, $\bar{v}_l$ denote the parameter of $\mathbf{Q}_{k,l}$.

The general procedure for obtaining the control mesh $\mathbf{b}_{i,j}$ and the knot vector U, V is described in detail in [20], which is summarized as follows.

1) Determine the order p, q in direction u, v.
2) Compute $\bar{u}_k$ and $\bar{v}_l$, the parameter of $\mathbf{Q}_{k,l}$ in the parameter space.
3) Calculate the knot vector U, V from $\mathbf{Q}_{k,l}$.
4) Obtain $(N+1) \times (M+1)$ equations for $\mathbf{b}_{i,j}$ using (2).
5) Solve the equations using the least mean square (LMS) estimation technique.

D. Continuity of the NURBS Surface Network

In the presence of multiple patches, the continuity across patch boundary is another issue. Actually, since the order of NURBS surface is set to 2, up to C^2 continuity should be satisfied inside each patch. On the other hand, without further processing, C^0 continuity across a shared boundary curve is achieved, by averaging the end control points between adjacent patches. However, in order to satisfy the minimal level of continuity, which is required for graphic animation and industrial fields, G^1 continuity is achieved by modifying the end control points such that the tangent control points line up in a fixed ratio over the length of the boundary. For patch corners, averaging four adjacent points of the shared patches will suffice the G^1 continuity on the corner.

VI. Experimental Results

To evaluate the performance of the proposed algorithm, simulation results are presented in this section. Experiments are carried out on the torus and the phone data set, which is composed of 16 200 and 41 522 points, respectively. The torus data is a synthetic data set, while the phone data set is obtained by registering and integrating ten different range views [2], which are shown in Fig. 10(a) and Fig. 11(a), respectively.

We first approximate the initial model from the input data. The number of clusters K is set to 80 and 130 for the torus and phone model, respectively. According to [17], it is observed that the initial polyhedral model is successfully built, when the number of points in a point patch is approximately 200–500, depending on the topological complexity of the model.

The centroid of each point patch is obtained by employing the Lloyd algorithm [15], as shown in Figs. 10(b) and 11(b), respectively. Then the range data is segmented into K point patches. The polyhedral approximation of each point patch is shown in Figs. 10(c) and 11(c), respectively. After simple division of the polygons, the triangular mesh model is constructed as shown in

TABLE II
ANALYSIS OF MODELING ERROR AND EXECUTION TIME

Step		Torus (K=80)		Phone (K=130)	
		E_{avg}	Execution time (sec)	E_{avg}	Execution time (sec)
Initial model		0.37	42	0.25	160
Quadrilateral patch model		0.40	27	0.28	108
NURBS network ($n \times m$)	$n = m = 4$	0.031	73	0.038	194
	$n = m = 8$	0.029	276	0.029	730
	$n = m = 16$	0.028	1089	0.027	2875

Figs. 10(d) and 11(d), which is approximated by 470 and 722 triangles, respectively.

As described in Section V, the initial model is represented by the hierarchical graph, in which the polyhedral and triangular model form the top and bottom level, respectively. The parameter space of the NURBS patch network is then constructed in quadrilateral patch form, using the subgraph operations. The results are shown in Figs. 10(e) and 11(e), which consists of 235 and 361 quadrilaterals, respectively. After constructing the NURBS parameter space, parameterization of each parameter space is followed, yielding the input mesh for the NURBS fitting. Then, by the surface fitting, the NURBS control mesh and the knot vector are generated. In Figs. 10(f) and 11(f), the generated control meshes are illustrated. The final construction of the NURBS surface model is shown in Figs. 10(g)–(i) and 11(g)–(i), respectively, at different resolutions. Note that, in the fitting procedure, the second order NURBS surface is employed, yielding C^2 continuous in each patch and C^1 continuous on the boundary of the neighboring patches. As shown in Figs. 10(g)–(i) and 11(g)–(i), the visual accuracy of the generated surface is further improved at the finer resolution. In Figs. 10(g)–(i) and 11(g)–(i), it is observed that the NURBS surfaces are rendered realistically.

In Table II, we present the modeling error and the execution time. Note that the algorithm is implemented on a 266-MHz Pentium II processor. The normalized modeling error E_{avg} is defined as the ratio of the mean absolute distance D_{avg} from the input data to the surface model and the diagonal length L of the bounding box of the object, given by

$$D_{\text{avg}} = \frac{\sum_{i=0}^{N-1} |d_i|}{N} \tag{3}$$

$$E_{\text{avg}} = \frac{D_{\text{avg}}}{L} \tag{4}$$

where d_i and N denote the signed distance from the range data x_i and the number of the total range data, respectively. It is observed that the modeling error of the NURBS surface is almost negligible and is reduced to about 10% to the error of the triangular mesh model.

It is worthy to note that the proposed approach is still valid when the range data is obtained from the volumed object. Although the object is thin or concave, since we can obtain the normal vector of each point by utilizing the information from the acquisition and registration stage, it is possible to distinguish the direction of thin or concaved surface and to obtain the model. However, in the extreme case, when the range data is very thin in its limiting case, such as a sheet of paper, the proposed algorithm would return with incorrect result, which is in fact the limitation of the algorithm.

VII. CONCLUSION

In this paper, we proposed a new algorithm to build a 3-D surface model from scattered and unorganized range data, based on NURBS surface-fitting techniques. It was shown that initial polyhedral/triangular model was obtained from the input data, and the NURBS patch network was generated efficiently by using the proposed hierarchical graph representation. By constructing the NURBS patch network, it is possible to obtain a surface model for arbitrary shaped object, while not requiring any manual operations.

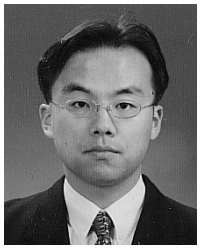
There are a number of issues for future research. The proposed algorithm could be extended to allow multiresolution modeling, by considering the local density of the range data and the surface curvature. Since the representation of a real-world object in 3-D video technology requires more compressed and compact model, the multiresolution property is of important factor in evaluating the performance of the algorithm. In addition to the geometrical topology, it is also expected that various mathematical properties of the NURBS surface can be applied to obtain the multiresolution modeling.

As we have already discussed, the NURBS is widely employed to represent free-formed object in CAD and graphics society. Since 3-D video technology is still in its infancy, there should be much more research and investigation in various fields, including model representation, 3-D surface data compression, and transmission.

REFERENCES

[1] M. Soucy and D. Laurendeau, "A general surface approach to the integration of a set of range views," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 17, pp. 344–358, Apr. 1995.
[2] G. Turk and M. Levoy, "Zippered polygon meshes from range images," in *Proc. SIGGRAPH'94*, July 1994, pp. 311–318.
[3] B. Curless and M. Levoy, "A volumetric method for building complex models from range images," in *Proc. SIGGRAPH'96*, Aug. 1996, pp. 303–312.
[4] M. Eck and H. Hoppe, "Automatic reconstruction of B-spline surfaces of arbitrary topological type," in *Proc. SIGGRAPH'96*, Aug. 1996, pp. 325–334.
[5] V. Krishnamurthy and M. Levoy, "Fitting smooth surfaces to dense polygon meshes," in *Proc. SIGGRAPH'96*, Aug. 1996, pp. 313–324.
[6] D. R. Forsey and R. H. Bartels, "Surface fitting with hierarchical splines," *ACM Trans. Graphics*, vol. 14, pp. 134–161, Apr. 1995.
[7] M. Soucy and D. Laurendeau, "Multiresolution surface modeling based on hierarchical triangulation," *CVGIP: Image Understanding*, vol. 63, no. 1, pp. 1–14, Jan. 1996.
[8] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, "Mesh optimization," in *Proc. SIGGRAPH'93*, Aug. 1993, pp. 19–26.
[9] 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.
[10] A. Lee, W. Sweldens, P. Schroder, L. Cowsar, and D. Dobkin, "MAPS: Multiresolution adaptive parameterization of surfaces," in *Proc. SIGGRAPH'98*, July 1998, pp. 95–104.
[11] N. Amenta, M. Bern, and M. Kamvysselis, "A new voronoi-based surface reconstruction algorithm," in *Proc. SIGGRAPH'98*, July 1998, pp. 415–422.
[12] Y. Chen and G. Medioni, "Description of complex objects from multiple range images using an inflating balloon model," *CVGIP: Image Understanding*, vol. 61, no. 3, pp. 325–334, May 1995.

[13] M. J. Milroy, C. Bradley, G. W. Vickers, and D. J. Weir, "G1 continuity of b-spline surface patches in reverse engineering," *Computer-Aided Design*, vol. 27, pp. 471–478, 1995.
[14] W. Ma and J. P. Kruth, "Parameterization of randomly measured points for least square fitting of b-spline curves and surfaces," *Computer-Aided Des.*, vol. 27, no. 9, pp. 663–675, 1995.
[15] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Norwell, MA: Kluwer, 1992.
[16] J. L. Marroquin and F. Girosi, "Some Extensions of the K-means Algorithm for Image Segmentation and Pattern Recognition," MIT Artificial Intelligence Laboratory, Cambridge, MA, AI Memo 1930, Jan. 1993.
[17] I. K. Park and S. U. Lee, "Geometric modeling from scattered 3-D range data," in *Proc. IEEE Int. Conf. Image Processing*, Oct. 1997, pp. 712–715.
[18] D. H. Chung, I. D. Yun, and S. U. Lee, "Registration of multiple range views using the reverse calibration technique," *Pattern Recognit.*, vol. 31, no. 4, Apr. 1998.
[19] G. Farin, *Curves and Surfaces for Computer Aided Geometric Design*. New York: Academic, 1990.
[20] L. Piegl and W. Tiller, *The NURBS Book*. New York: Springer-Verlag, 1995.
[21] S. Malassiotis and M. G. Strintzis, "Object-based coding of stereo image sequence using three-dimensional models," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, pp. 892–905, Dec. 1997.
[22] "Special issue on 3D and stereoscopic visual communication," *IEEE Signal Processing Mag.*, vol. 16, May 1999.



In Kyu Park (S'96) was born in Seoul, Korea, in 1972. He received the B.S. and M.S. degrees in control and instrumentation engineering from Seoul National University, Seoul, Korea, in 1995 and 1997, respectively, where he is currently working toward the Ph.D. degree in electrical engineering.

His research interests include computer vision, computer graphics, and image processing.

Il Dong Yun (S'89–M'97) was born in was born in Seoul, Korea, in 1965. He received the B.S., M.S., and Ph.D. degrees in control and instrumentation engineering from Seoul National University, Seoul, Korea, in 1989, 1991, and 1997, respectively.

From 1996 to 1997, he was with Daewoo Electronics, Seoul, Korea, as a Senior Engineer. Since 1997, he has been with the Hankuk University of Foreign Studies, Yongin, Korea, where his work includes object modelling from range data, content-based multimedia indexing, and photometric stereo techniques.

Sang Uk Lee (S'75–M'80–SM'99) received the B.S. degree from Seoul National University, Seoul, Korea, in 1973, the M.S. degree from Iowa State University at Ames, in 1976, and the Ph.D. degree from the University of Southern California at Los Angeles in 1980, all in electrical engineering.

In 1980–1981, he was with General Electric Company, Lynchburg, VA, working on the development of digital mobile radio. In 1981–1983, he was a Member of Technical Staff with M/A-COM Research Center, Rockville, MD. In 1983, he joined the Department of Control and Instrumentation, Seoul National University, as an Assistant Professor, and is now a Professor of the School of Electrical Engineering. He is also affiliated with the Automation and Systems Research Institute and the Institute of New Media and Communications, also at Seoul National University. His current research interests include the areas of image and video signal processing, digital communication, and computer vision.

Dr. Lee served as Editor-in-Chief for the *Transactions of the Korean Institute of Communication Science* from 1994 to 1996. Currently, he is member of the Editorial Board of the *Journal of Visual Communication and Image Representation*, and an Associate Editor for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY. He is a member of Phi Kappa Phi.