Curved Surface Reconstruction Using Active Tactile Sensing and Surface Normal Information

Liu, Fuming
Department of Intelligent Systems, Faculty of Information Science and Electrical Engineering, Kyushu University

Hasegawa, Tsutomu
Department of Intelligent Systems, Faculty of Information Science and Electrical Engineering, Kyushu University

https://doi.org/10.15017/1515731

出版情報：九州大学大学院システム情報科学紀要. 6 (2), pp.161-168, 2001-09-26. 九州大学大学院システム情報科学研究院

九州大学大学院
システム情報科学紀要
第6巻 第2号 平成13年9月

Research Reports on Information Science and
Electrical Engineering of Kyushu University
Vol.6, No.2, September 2001

Curved Surface Reconstruction Using Active Tactile Sensing and Surface Normal Information

Fuming LIU* and Tsutomu HASEGAWA*

(Received June 12, 2001)

Abstract: In this paper we propose an approach to surface reconstruction using active tactile sensing and surface normal information. Unlike previous works which consider primarily the problem of reconstructing a single tensor B-spline patch based on deformed planar or spherical domain, the surface reconstructed here is a network of triangular B-spline patches based on arbitrary topological domain and can maintain tangent plane continuity. The input is unorganized tactile sensing 3D data points with normal information of arbitrary topological type surfaces, which are obtained by a dextrous multi-fingered hand equipped with force/moment tactile sensors. The surface reconstructed is a parameterization of the data over these patches. An adaptive refinement is explored on the patch network in order to satisfy a given error tolerance. Demonstrations are carried out on simulated and experimental data.

Keywords: Surface reconstruction, Active tactile sensing, Surface normal information, Triangular B-spline surface

1. Introduction

For future robots to be able to replace human being in various tasks, they must have the capability of recognizing the unknown surrounding environment. In particular, robots must be able to recognize the shape of an unknown object *a priori*. We, human being can obtain the shape of an object by seeing or touching. Similarly, there are two means to obtain the shape of an object in robotics, one is robot vision, the other is tactile sensing. Robot vision can provide global shape information of an object quickly by utilizing visual tools such as CCD camera and laser range finder. Tactile sensing can detect the shape of an object through utilizing a tactile sensor to directly touch the object. If an object appears in a scene of visually bad conditions, robot vision fails easily. If an object is occluded from view, robot vision becomes useless. In these cases, tactile sensing may be the best alternative means to robot vision. Because the occlusion of view cannot be avoided in the vision system, in order to obtain shape data points of a whole object, the object must be scanned from multiple view points. Consequently, the synthesizing of these data points is necessary, this is a very tremendous work. Tactile data can be used directly, although it is generally sparse compared with the scanned data in visual tools. This sampling density can be improved through repeated tactile sensing.

Tactile sensor is an essential prerequisite for the implementation of tactile sensing. Some of the most useful parameters to recover through tactile sensor are angle and magnitude of force, surface normal, location of contact, curvature, and type of contact (point, line, plane). There are mainly two different types of tactile sensors to obtain contact information in robotics. One is a fingertip force/moment sensor that gives location, resultant force and surface normal using strain gauge structure[1]. The other is an array sensor that has an advantage for determining contact shape from a single contact measurement. Fingertip force/moment sensors are convenient in manipulation. Array sensors however are not very useful for manipulation because they are usually designed in flat shape.

Since tactile sensor provides only local information of contact location, in order to obtain the whole shape information of the unknown object, robots have to keep sliding and/or rolling contact with the object through manipulation. This type of combined manipulation and tactile sensing, or the so-called active tactile sensing, is essential to tactile perception.

Considering the slip between tactile sensors and an object, it is so difficult to keep pure rolling contact that we only utilize sliding contact to explore the shape of an object. Using the sliding exploratory procedure, local and global shape properties of an object can be explored.

Assume the object is fixed, we can use any fin-

* Department of Intelligent Systems

ger or all fingers simultaneously of multi-fingered hand to slide along the object to obtain shape data. Because semi-spherical fingertip force/moment tactile sensors are fixed on the hand, the position and surface normal information are simultaneously obtained while exploring is being implemented. These 3D data points with normal information gathered about the shape of the object are unorganized and used to reconstruct the surface of the object. The reconstructed surface of the object can guide the further shape recognition of the object.

In this paper we present an approach to automatic reconstruction of a triangular B-spline surface of arbitrary topological type from an unorganized set of 3D data points with normal information, which are obtained by a dextrous multi-fingered hand equipped with tactile sensors. Unlike previous work which considers primarily the problem of reconstructing a single B-spline patch based on deformed planar or spherical domain, the surface reconstructed here is a network of triangular B-spline patches based on arbitrary topological domain and can maintain tangent plane continuity. The surface reconstructed is a parameterization of these data points over these patches. An adaptive refinement is explored on the patch network in order to satisfy a given error tolerance. Demonstrations are carried out on simulated and experimental data.

The paper is organized as follows. Related works are described in the next section. In the following sections, a brief overview of triangular B-spline surface is first provided, and the reconstruction algorithm is described in detail. Then, the the results of simulation, experimental setup and the results of experiment are shown. Finally, we make the conclusion.

2. Related Works

There have not been so many works discussing surface reconstruction from tactile sensing. Bays[2] proposed a simple multi-fingered surface exploration procedure, in which only the normal direction of the force sensor information is utilized for estimating the surface parameter in a Kalman-filtering style. Hong Zhang *et al.*[3] have developed a scheme that the unknown object is expressed as a set of local contact frames and the surface parameters associated with each frame, and an active tactile sensing strategy for shape recovery. Rao *et al.*[4] have recovered the convex shape of a polygonal part from a sequence of projections. Projecting the part onto an axis in the plane of the part produces a scalar measure, the diameter, which is a function of the angle of projection. The diameter of a part at a particular angle can be measured using an instrumented parallel-jaw gripper. Lindenbaum *et al.*[5] give algorithms for the approximate reconstruction of arbitrary convex planar shapes using line probes. That work need not restrict the objects to be considered to be polygonal. Fearing[6] has attempted to recover the shape of a certain class of generalized cones, known as Right Linear Straight Homogeneous Generalized Cones(RLSHGC), from multi-fingered sparse tactile information including point contacts, surface normal and surface curvature. Beccari *et al.*[7] describe techniques for recognizing convex polyhedra from sparse tactile data, using volumetric representations and active tactile exploration. Charlebois *et al.*[8] describe a method for curvature estimation, which entails dragging a set of fingers over the surface, then fitting the resulting contact data to a tensor B-spline description of the surface patch. Allen *et al.*[9] performs exploratory operations such as contour following to recover shape, fitting the resultant tactile data to a superquadric description of the unknown shape.

These papers described so far mainly focus on the reconstruction of surfaces of simple topological type, such as deformed planar regions and spheres, and address the problem of reconstructing surfaces using tensor product B-spline patches. Unfortunately, tensor product surfaces are inherently rectangular in nature. While tensor products have proven themselves an excellent tool for the modeling of largely regular surfaces, they have well-known draw-backs if the modeling of largely irregular objects is required. Triangular patches can represent any irregular objects, therefore, not surprisingly, reconstructing surface of triangular patches is being regarded as a matter of course.

3. Overview of Triangular B-spline Surface

This section provides a brief overview of triangular B-spline surface. Triangular B-splines were introduced by Dahmen, Micchelli and Seidel in[10].

3.1 Simplex Splines

Simplex splines form the individual basis functions for the triangular B-spline surface. Let $T = \{\triangle(I) = [v_i, v_j, v_k] \mid I = (i,j,k) \in \mathcal{I} \subseteq \mathcal{Z}_+^3\}$ be an arbitrary triangulation of the parameter plane R^2, or of some bounded domain $D \subseteq R^2$, and suppose that a sequence of knots $\{v_{i,0}, \ldots, v_{i,n}\}$

is assigned to every vertex v_i in this triangulation in such a way that $v_{i,0} = v_i$, and such that any three knots form a proper triangle. Let $V_\beta^I = \{v_{i_0}, \ldots, v_{i_{\beta 0}}, v_{j_0}, \ldots, v_{i_{\beta 1}}, v_{k_0}, \ldots, v_{i_{\beta 2}}\}$.

Consider the three knots $V = (v_0, v_1, v_2)$, the *constant simplex spline* is defined as

$$M(u|V) = \begin{cases} \frac{1}{|d(V)|} & if \quad u \in [V) \\ 0 & otherwise \end{cases} \tag{1}$$

where d(V) is twice the area of the triangle $\triangle(v_0, v_1, v_2)$, $[V)$ is half-open convex hull of V.

Let $W = \{w_0, w_1, w_2\}$ be any subset of affinely independent knots from V_β^I, and the *higher order simplex spline* is defined recursively as

$$M(u|V_\beta^I) = \sum_{j=0}^{2} B_j(u|W) M(u|V_\beta^I \backslash \{w_j\}) \tag{2}$$

where $B_j(u|W)$ are the barycentric coordinates of u with respect to W.

In order to make these splines useful for modeling applications, they must be renormalized to ensure affine invariance. Let $d_\beta^I = d(w_{\beta_0}, w_{\beta_1}, w_{\beta_2})$, the normalized B-splines $N(u|V_\beta^I)$ are defined as

$$N(u|V_\beta^I) = d_\beta^I M(u|V_\beta^I) \tag{3}$$

As a result, the normalized simplex splines form a partition of unity.

3.2 Triangular B-spline Surface

For every triangle I and every triple of indices β, a control point $c_\beta^I \in R^3$ is defined. An arbitrary triangular B-spline surface of degree n over a given triangulation T is then defined as

$$S(u) = \sum_{I \in \mathcal{I}} \sum_{|\beta = n|} c_\beta^I N(u|V_\beta^I) \tag{4}$$

The points c_β^I control the shape of the surface S(u). Some of the main properties of the triangular B-spline surface are summarized below:

- Piecewise Polynomial: S(u) is a piecewise polynomial of degree n.
- Convex Hull Property: S(u) lies in the convex hull of its control points c_β^I.
- Locality: Movement of the control points c_β^I only influences the region of the surface on the triangle $\triangle$(I) and on those triangles surrounding it.
- Smoothness: A degree n triangular B-spline surface is a piecewise polynomial of degree n over the subtriangulation induced by its knot net that is C^{n-1} continuous everywhere if its knots are in general position.
- Affine Invariance: The relationship between the control points and the triangular B-spline surface is invariant under affine coordinate transformations.

4. Algorithm

Our triangular B-spline surface reconstruction algorithm consists of 4 successive phases, which is shown in **Fig.1** and described in details in the latter.

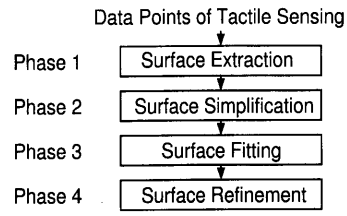

Fig.1 Flow chart of surface reconstruction.

4.1 Phase 1: Surface Extraction

In this phase, we implement the extraction of surface based on Marching Cube Algorithm[11]. The Marching Cube Algorithm was created to render volumetric-based medical imagery. Such imagery consists of a set of three dimensional voxels, which contain data from medical sensors such as CT scans. The algorithm takes its name from the fact that it marches around the voxels and produces a set of triangles for each voxel element.

Before we can implement the Marching Cube Algorithm, we must convert these 3D data points with normal information into a volume dataset. The volume dataset is defined as a distance volume. That is, the value stored at each voxel is the shortest distance of these 3D data points to the surface of the object being reconstructed by the volume dataset. Because each data point has normal information, using the 3D value and normal value of each data point, we can assign a tangent plane to each data point. These tangent planes may serve as local linear approximations to the surface. However because of the noise of data point with normal information, the reconstructed surface is not reliable using these tangent planes. We assign a narrow region to every 3D data point and obtain a new 3D data point($\bar{p}$) with normal information($\bar{n}$), which is

shown in **Fig.2**. The new 3D data point is the center of gathered 3D data points in the narrow region, and the new normal vector is obtained by summing the normal vectors of 3D data points. Then, these new 3D data points is converted into distance volume.

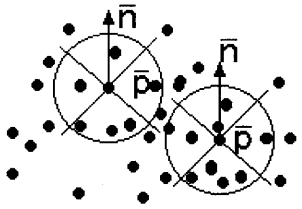

Fig.2 New 3D data Points with normal vectors.

4.2 Phase 2: Surface Simplification

The Phase 1 generally produces a large number of triangles. The sizes of these triangles vary significantly. We must simplify the surface reconstructed to satisfy surface fitting in the next phase through reducing the number of triangles. In this phase, we only applied the Mesh Optimization method which is developed by Hoppe *et al.*[12].

4.3 Phase 3: Surface Fitting

Based on the triangulation T obtained in Phase 2, we first calculated all the triangular B-spline basis functions. Now the work of surface fitting is to find the control points c_β^I of all patches of T such that the distance of these data points P_i to the surface S(u) is minimized. More precisely, we minimize the distance functional

$$E(S)_{dist} = \sum_{i=1}^{i=N} \|P_i - S(P_i)\|^2 \tag{5}$$

This is a type of nested minimization problem. Iterative method has been developed to solve this. In the method, each iteration consists of two steps:

- 1. Fitting step: For fixed triangulation T, the optimal control points c_β^I are found by solving a linear least-squares problem.
- 2. Triangulation T correction step: For fixed control points c_β^I, new Triangulation T_{new} is produced by calculating the values of all vertices of Triangulation T.

Usually the accuracy of fitting is improved considerably after only a few iterations (we typically use 4).

One problem with surface fitting is that the resulting surface may have unwanted "wiggles". It is therefore common to augment the energy functional with an additional fairness term. That is, we must minimize the new energy functional

$$E(S) = E(S)_{dist} + \lambda \cdot E(S)_{fair} \tag{6}$$

Because the normal vectors(N_i) of surface can be obtained during tactile sensing, we use the surface normal information in the fairness term. First the normal vectors(N_i) of surface are converted into derivatives in the u and v directions using the following equation

$$T_u = [1, 0, -\frac{n_{ix}}{n_{iz}}] \ , \ T_v = [0, 1, -\frac{n_{iy}}{n_{iz}}] \tag{7}$$

where, the vector $[n_{ix}, n_{iy}, n_{iz}]$ is the value of the normal vectors (N_i) of surface in the coordinate $\{u, v, u \otimes v\}$, the $\otimes$ is the output product. The $\frac{\partial s_f}{\partial u}$ and $\frac{\partial s_f}{\partial v}$ are the derivatives of the triangular B-spline surface in the u and v directions. The fairness term is then defined as

$$E(S)_{fair} = \sum_{i=1}^{i=N} \left((T_u - \frac{\partial s_f}{\partial u})^2 + (T_v - \frac{\partial s_f}{\partial v})^2 \right) \tag{8}$$

There remains the problem of finding a reasonable choice for the fairness weight λ. The λ makes the surface fit to preferentially approximate the position or the normal vector at each data point.

About continuity, simple geometry continuity (G^0) is first achieved trivially by sharing control points along the boundaries of adjacent patches. Because the control points of these triangular B-spline patches are computed using local combinations of vertices in triangulation T, and the triangular B-spline patches then automatically meet with tangent plane (G^1) continuity.

4.4 Phase 4: Adaptive Refinement

The surface fitting algorithm described in Section 4.3 minimizes the total squared distances $\sum_i \|P_i - S(P_i)\|^2$ of these data points P_i to the B-spline surface S. It is often desirable to specify a maximum error tolerance for the fitting. Phase 4 attempts to find a surface S such that $\max(\|P_i - S(P_i)\|) < \epsilon$ for a user-specified error tolerance ϵ. To achieve a given tolerance within our least squares optimization framework, it may be necessary to introduce new degrees of freedom into the surface representation. One could achieve this by globally subdividing the Triangulation T (using templates in **Fig.3**). However, this would introduce degrees of freedom uniformly over the whole surface, even if data points

exceed the error tolerance only in isolated neighborhoods.

We instead develop an adaptive refinement scheme. We firstly define a face of Triangulation T onto which any point P_i projects with $\|P_i - S(P_i)\| > \epsilon$ as Larger Face(LF), and the face that have a common edge with LF as Adjacent Face(AF), then the goal of this refinement scheme is to subdivide any LF of Triangulation T, while at the same time ensuring that the resulting subdivided faces still form a valid Triangulation T_n. We use the template 1 in **Fig.3** to subdivide LF , and use the template 2 in **Fig.3** to subdivide AF, the edge to be subdivided in template 2 is common edge of a LF and a AF. The values of new vertices introduced in Triangulation T_n lie at the midpoints of edges are calculated using triangular B-spline. Having constructed the locally refined Triangulation T_n, we update the parameterizations of these data points P_i.

After adaptive refinement, the fitting method of Phase 3 is reinvoked. The resulting surface may still not be within ϵ of all the points, indicating that further refinement is necessary. We repeat the phases of refinement and fitting until the error tolerance ϵ is satisfied.

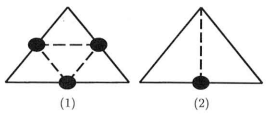

Fig.3 Subdivide templates.

5. Simulation Study

The algorithm proposed in Section 4 is executed on a Solaris workstation which has a 600 MHz Intel Pentium III processor. To test effectiveness of the algorithm, we first used the simulated data, which has 1229 data points with normal information. These data points with normal information are shown in **Fig.5**. The surface extracted by the Phase 1 is the **Fig.6**, which has 1314 triangular patches. The surface simplified by the Phase 2 is the **Fig.7**, which has 174 triangular patches. The surface fitted by the Phase 3 is the **Fig.8**, which has error sum 0.0023. The surface refined by the Phase 4 is the **Fig.9**, which has 195 triangular patches. The error tolerance ϵ of each patch is set as 0.0005, and the error sum of all patches is 0.0014. Utilizing the algorithm we proposed, any arbitrary topological type surface can be reconstructed. As a result, we can generate any resolution surface. A high resolution surface is generated and shown in **Fig.10**. The execution times(minutes) of 4 successive phases are 0.03, 0.2, 28, 35, respectively.

6. Experimental Study

We also used experimental data to test the effectiveness of the algorithm proposed in Section 4 on the same Solaris workstation as used in the simulation study. The data is obtained from the tactile sensing with a multi-fingered hand. The experimental setup is described in the following subsection.

6.1 Experimental Setup

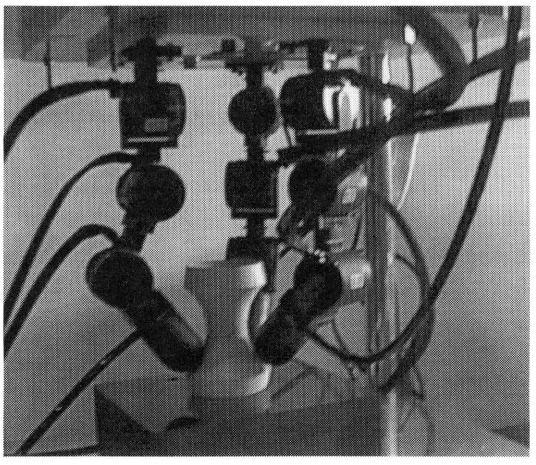

Fig.4 Experimental setup.

The experimental setup for our research is a dextrous hand contains three fingers, which is manufactured by YASKAWA Electric Corporation and shown in **Fig.4**. Each finger consists of three actuators and three rigid links, and has three degrees of freedom. Each of these actuators is a motor unit with a reduction gear and an encoder in its body. The motor is AC torque with 0.71Nm rated torque. The gear ratio of motor is 80. Each fingertip of three fingers is equipped with a six-axis force/moment sensor. The radius of semi-sphere at each fingertip is set to 20[mm]. The detected contact point error of six-axis force/moment sensor is under 0.5[mm], and the error of normal vector is under 3.0 degrees.

The real time computational platform is a SPARC CPU-3C, which implements the capabilities of the SUN MicroSystems SPARCclassic workstation on a single board with VME bus and connected to a local network. The real time operating system is VxWorks developed by Wind River Systems. The CPU cycle of the microSPARC microprocessor is 110MHz. Except for the time of moving of fingers, it takes 5ms to obtain a contact point data

with normal information.

6.2 Results of Experiment

To save time, we first only use a finger to explore a part of the object and obtain 374 data points. Then we make twice different coordinate transformation of these data points and assume the transformed data points be the remained part of the object. The data obtained from experiment has 1122 data points. These data points with normal information are shown in **Fig.11**. The surface extracted by the Phase 1 is the **Fig.12**, which has 860 triangular patches. The surface simplified by the Phase 2 is the **Fig.13**, which has 93 triangular patches. The surface fitted by the Phase 3 is the **Fig.14**, which has error sum 0.0248. The surface refined by the Phase 4 is the **Fig.15**, which has 169 triangular patches. The error tolerance ϵ of each patch is set as 0.003, and the error sum of all patches is 0.009. A high resolution surface is generated and shown in **Fig.16**. The execution times(minutes) of 4 successive phases are 0.02, 0.2, 38, 47, respectively.

7. Conclusion

We have developed an approach to reconstruction of a G1 triangular B-spline surface of arbitrary topological type from 3D data points with normal information without user assistance. The approach makes use of a surface spline construction to obtain G1 continuity, we show that such an approach leads to an efficient B-spline fitting method. We have introduced an adaptive refinement scheme. Finally, we have applied our approach to reconstruct B-spline surfaces within user specified maximum error tolerances on simulated and experimental data. There exist a number of areas for future research. The approach should be extended to allow reconstruction of piecewise smooth surfaces that contain discontinuities such as creases and corners. Currently our algorithm has difficulty with such features, as it approximates them by adaptively refining the smooth surface numerous times. Identifying these discontinuities on the surface may require some user intervention. Hopefully semi-automated segmentation methods can be developed that do not require complete specification of patch boundaries. Such methods could replace Phase 1 and 2 of our approach.

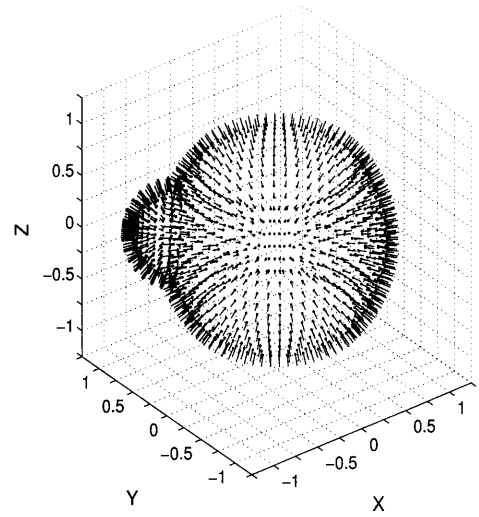

Fig.5 Simulated data points with normal vectors.

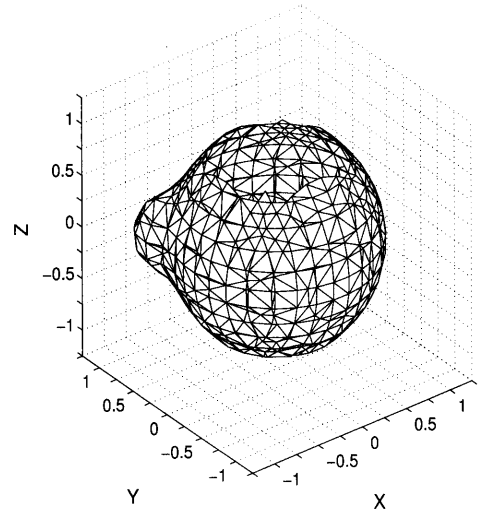

Fig.6 Extracted surface of simulated data.

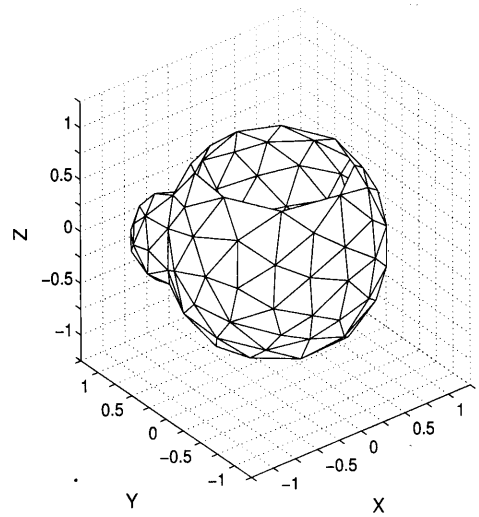

Fig.7 Simplified surface of simulated data.

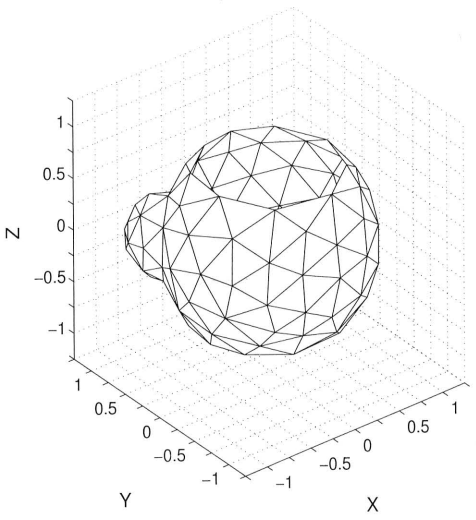

Fig.8 Fitted surface of simulated data.

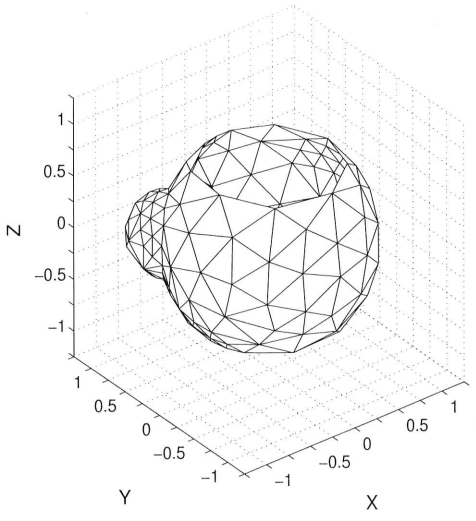

Fig.9 Refined surface of simulated data.

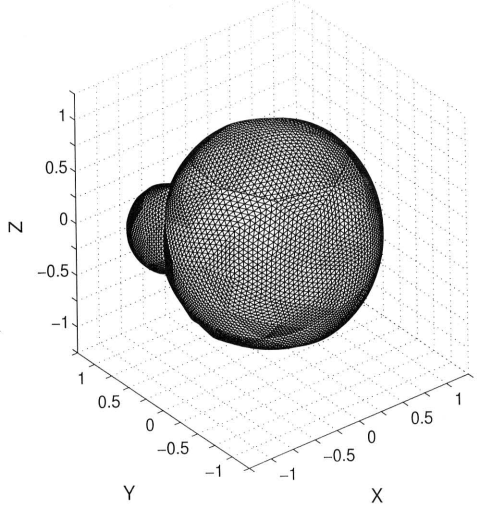

Fig.10 Generated surface of simulated data.

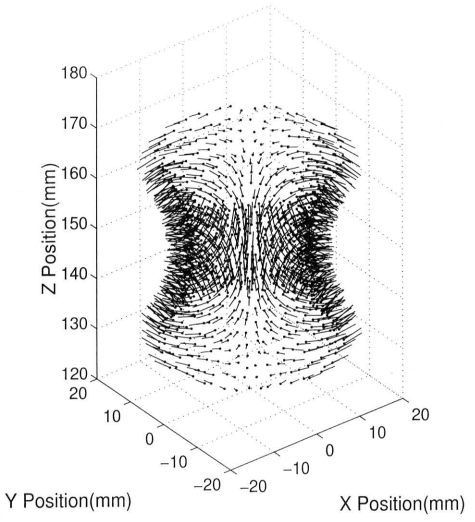

Fig.11 Experimental data points with normal vectors.

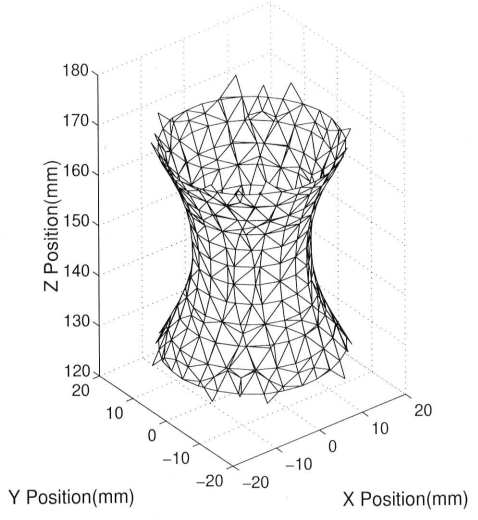

Fig.12 Extracted surface of experimental data.

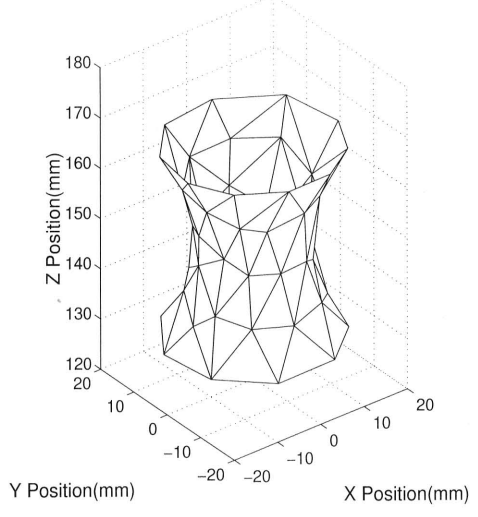

Fig.13 Simplified surface of experimental data.

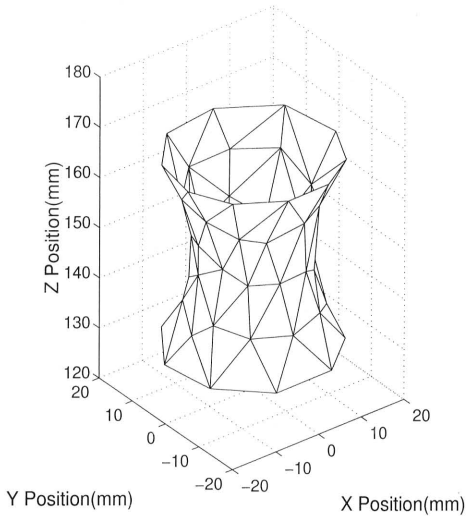

Fig.14 Fitted surface of experimental data.

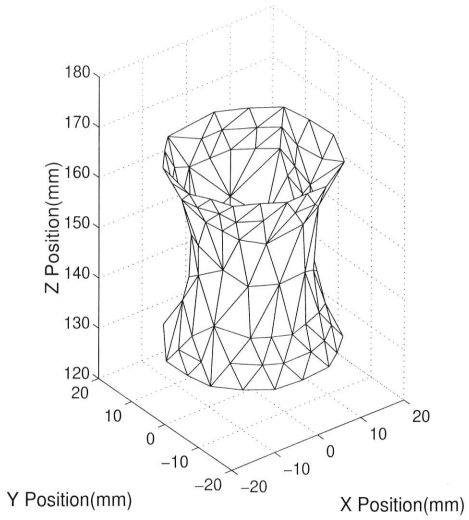

Fig.15 Refined surface of experimental data.

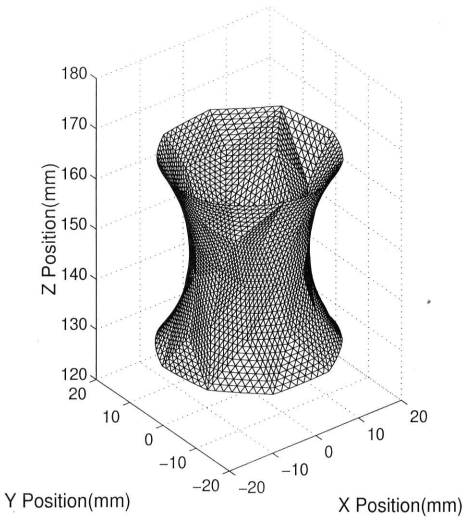

Fig.16 Generated surface of experimental data.

References

1) A. Bicchi: *Intrinsic Contact Sensing for Soft Fingers*, Proceedings of IEEE International Conference on Robotics and Automation, pp.968-973, 1990.
2) J.S. Bays: *Tactile Shape Sensing via Single- and Multi-Fingered Hands*, Proceedings of IEEE International Conference on Robotics and Automation, pp.290-295, 1989.
3) H. Zhang, N. Chen and R. Rink: *Local Object Shape From Tactile Sensing*, Proceedings of IEEE International Conference on Robotics and Automation, pp.3496-3501, 1996.
4) A.S. Rao and K.Y. Goldberg: *Shape from Diameter: Recognizing Polygonal Parts with a Parallel-Jaw Gripper*, International Journal on Robotics Research, Vol. 13, pp.16-37, 1994.
5) M. Lindenbaum and A. Bruckstein: *Blind Approximation of Planar Convex Sets*, IEEE Transactions on Robotics and Automation, Vol. 10, No.4, pp.517-529, 1994.
6) R.S. Fearing: *Tactile Sensing for Shape Interpretation* In *Dexterous Robot Hands*, S.T. Venkataraman, T. Iberall, Berlin: Springer-Verlag, pp.209-238, 1990.
7) G. Beccari, S. Caselli and F. Zanichelli: *Pose-Independent Recognition of Convex Objects from Sparse Tactile Data*, Proceedings of IEEE International Conference on Robotics and Automation, pp.3397-3402, 1997.
8) M. Charlebois, K. Gupta and S. Payandeh: *Shape Description of General, Curved Surfaces Using Tactile Sensing and Surface Normal Information*, Proceedings of IEEE International Conference on Robotics and Automation, pp.2819-2824, 1997.
9) P.K. Allen and P. Michelman: *Acquisition and Interpretation of 3-D Sensor Data from Touch*, IEEE Transactions on Robotics and Automation, Vol. 6, No. 4, pp.397-404, 1990.
10) W. Dahmen, C.A. Micchelli and H.-P. Seidel: *Blossoming begets B-Splines built better by B-Patches*, Mathematics of Computation, Vol. 59, No. 199, pp.97-115, 1992.
11) W. E. Lorensen and H. E. Cline: *Marching Cubes: a high resolution 3D surface construction algorithm*, Computer Graphics (SIGGRAPH '87 Proceedings), pp.163-169, 1987.
12) H. Hoppe, T. DeRose, T. Duchamp, J. McDonald and W. Stuetzle: *Mesh Optimization*, Computer Graphics (SIGGRAPH '93 Proceedings), pp.19-26, 1993.