基礎論文 TVRSJ Vol.5 No.4 pp.1079－1086, 2000

The Transformation of Range Data Based on the Combined Application of Image Segmentation and NURBS Surface

Guoqing DONG,[*,**] Yasuyuki YANAGIDA,[*] Naoki KAWAKAMI,[*] Taro MAEDA,[*] and Susumu TACHI[*]

イメージ分割とNURBS曲面の結合応用によりレンジデータの変更

董 国卿[*,**]，柳田康幸[*]，川上直樹[*]，前田太郎[*]，舘 暲[*]

Abstract: A real object is often described by its geometric surface and color image. The surface properties and color features sometimes correspond with each other. This paper introduces a new method for the redesigning of 3D model based on the combined application of image processing and computer graphics. In particular, we focus on the 2D planar images and 3D range data of a real object, which are produced by a laser range finder and determined in an identical coordinates system.

Such a method allows us to create a variety of complex models from an original model. Image processing is used to specify the features of a real object according to its RGB values so as to avoid the inconvenience of analyzing surface properties. We first detect the feature region through image processing, for example, color thresholding, edge detection, and labeling. Then, we displace the original range data within the detected region with nonuniform rational B-spline (NURBS) surface data. Finally, we reconstruct the new range data set with a smooth subdivision surface and map it with an original image. From just one real object, this method can provide a number of transformed models.

Keywords: mesh, image segmentation, laser range finder, 3D shape recovery, texture mapping

1 Introduction

Generally, a virtual environment is constructed by making up primitives or hand-modeled geometric representations in computer graphics (CG) [1], a method that is well known as model-based rendering (MBR). However, attempting to create an even more realistic model by such a method presents numerous disadvantages, such as requiring an enormous amount of time and special skills. Moreover, detailed and realistic representations of some real objects can hardly be created by MBR due to the inherently limited possibilities of an individual. Concerning a practical application of virtual reality such as a virtual museum or a virtual operation-training system, the real object serving as the original for the virtual object has been an existing object. If the real object can be expressed directly in the virtual environment, it is possible both to provide a sense of excellent presence and reality and reduce the amount of time and effort [2][3][4]. Furthermore, if a more complicated model can be generated based upon the original appearance of a real object without regenerating from the beginning, the process to design a new model will become much less time-consuming and complex. Our particular area of interest focuses on this intent, that is, transforming the partial range data of a real object within a segmented region to develop a new type of model. From just one real object, this method can provide a number of complicated models, which are suitable for virtual reality and computer-aided design (CAD) as well as many other areas.

1.1 Background and Motivation

A notable property of the current range acquisition techniques is the ability to capture range and color data from the same viewpoint [5]. Various studies have attempted new ways to enhance color images of real scenes by providing range data to each pixel correspondingly, that is, by means of a unified application of image representation and CG [3][4][6][7]. By matching the range data to the color image in image-based rendering (IBR), it is possible to accommodate correct occlusions and change the viewpoint freely [3]. On the other hand, Shade et al. [6] and Oliveira et al. [7] proposed an image-based representation of real objects with layered depth image. However, these algorithms only attempt to represent the original appearance of an object and render the model from arbitrary viewpoints but do not consider transforming the original model. In other words, the

*東京大学大学院工学系研究科
**（現在）イメージ情報科学研究所
*School of Engineering, University of Tokyo
**Laboratories of Image Information Science and Technology (now)

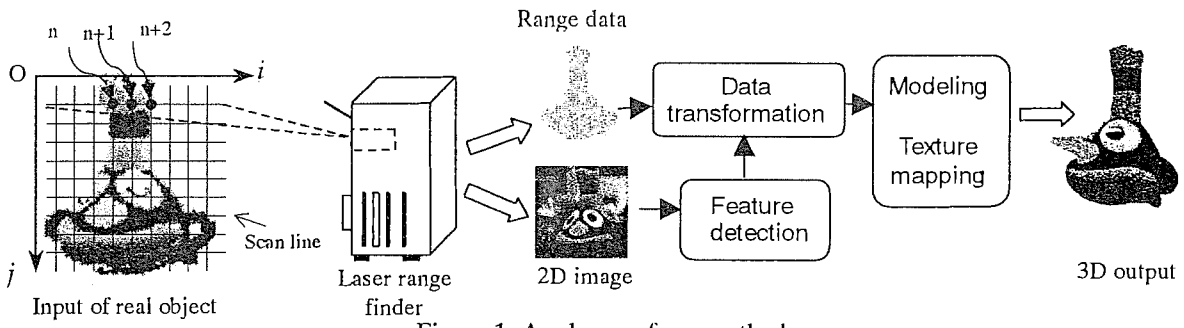


Figure 1. A schema of our method

range data serves the need of image representation.

For the transformation of a model, morphing has been an area of active research in the past years. Morphing is the process of gradually changing a source object through intermediate objects into a target object [8][9][10]. However, such an algorithm requires not only having more than two model data sets but also computing the correspondence between all of the models.

With regard to the segmentation of a polygonal model, V. Krishnamurthy et al. [11] proposed a new method, namely to pick a sequence of points manually on a 2D projection of polygon meshes and smooth them on the surface of the model by means of a fitted B-spline curve. This method is complex and expensive to find a feature surface.

The objective of our work is to detect the feature contour of a real object according to its RGB values and then transform the original range data enclosed in the contour to design a new model. One reason for this idea is that, in many cases, a number of scanned objects are mostly multicolored, and the color features correspond to variations in surface orientation or surface property. If the object has only one color, such as manufacturable parts, we may tag the local area where we want to redesign with various colors before scanning. Another reason is that features are more easily and effectively detected on a planar image than on a 3D surface, and 3D edges are often mapped by imaging process into critical points of the 2D color profile formed in the eye or camera. Once the local feature is found, we may transform its geometric shape constructed with range data by using a nonuniform rational B-spline (NURBS) surface relying on the given correspondence between range and color data.

In brief, we first utilize an image-processing algorithm, such as color thresholding, edge detection and labeling, to extract the feature region without having to analyze surface properties. Second, we select the part of range data within the extracted region as control points to produce NURBS surface data in place of original range data and reconstruct entire range data set with a subdivision surface. Then the regenerated model is mapped with the texture of the real object. The schema of our algorithm is shown in Figure 1.

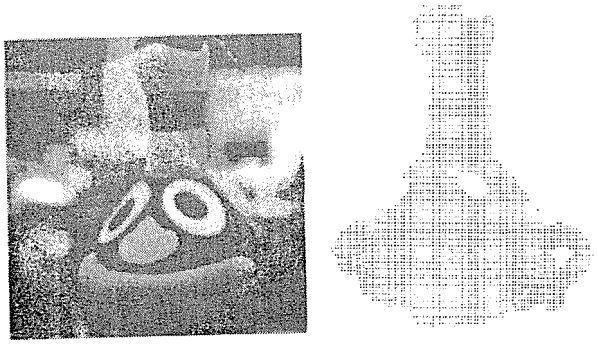
1.2 Advantages of NURBS

Besides the NURBS method, Camull Clark's subdivision scheme is another method used for smooth surface representation [12][13][14]. However, this method has the drawbacks of poor mathematical expression and rigid control net. For our study, we selected the NURBS algorithm to transform the range data because it offers following excellent advantages. First, the surface passes through the end points and lies within the convex hull of control points. Thus, the regenerated shape surface can fit in with the boundary constraints. It is not necessary to consider the polygon connectivity at the boundary or worry about the overlap with the range data outside the detected region. Second, the NURBS surface can offer one common mathematical form for both standard analytical shapes and free form shapes; this can help us reconstruct arbitrary geometric shape for desirable characters. Third, the smoothness and resolution of the surface can be controlled independently in both directions. If the shape is modified frequently in any direction, we can increase the resolution; conversely, we may reduce the resolution for the purpose of fast modeling. Fourth, the NURBS surface can be evaluated reasonably fast in numerically stable and accurate algorithms.

Although the NURBS algorithm has had many commercial applications such as Softimage [15], Alias Wavefront [16], and other reported work [17], these are only used to create pure man-made model and cannot handle range data. In contrast, our method deals with the range data and can be added to the utility software of laser range finder (LRF) driver.

1.3 Overview

The remainder of the paper is organized as follows. In the next section, we describe the supporting algorithms that support our scheme. In Section 3, we present the image processing used to extract the feature



(a) 400×400 texture image (b) 200×200 range data

Figure 2. Image and range data from LRF

contour. In Section 4, we describe how to transform the range data with NURBS method. In Section 5, we briefly introduce the texture mapping of the regenerated model. Section 6 closes with conclusions and a proposal for future work.

2 Supporting Algorithms

Before introducing our approach, we first give a description of the supporting algorithms that not only inspired us to propose the idea of this paper but also removed many roadblocks for us. These algorithms include range and image data acquisition and 3D shape recovery.

2.1 Range Data and Color Image Acquisition

Our prototype data-collecting device is a commercial LRF (VIVID700) from Minolta Co. Ltd. [5] controlled by an SGI workstation (Indigo^2, IRIX6.2). A CCD camera mounted with a laser scanner takes color images of 400×400 pixels (cf. Figure2.a) and simultaneously receives the reflected light from objects as a sensor for 200×200 spots range data (cf. Figure2.b). However, only those range data from the reflecting spot can be acquired. The distance resolution of VIVID700 is approximately 0.35mm (x,y coordinate) and 0.11mm (z coordinate). R, G, and B are 8 bits per component.

The range and color data produced by VIVID700 from the same viewpoint can be recorded in identical coordinates. That is, the correspondence between image value vector $C(I,J)=(R(I,J),G(I,J),B(I,J))$ and range data vector $\mathbf{R}(i,j)=(X(i,j),Y(i,j),Z(i,j))$ is internally guaranteed, where $I,J\in[400,400]$ and $i,j\in[200,200]$. The image value at position $(I=2i,J=2j)$ corresponds to range data of position (i,j). This property makes it unnecessary to compute the corresponding relationship. In fact, in many similar studies, researchers have had to set up such a relationship at the start [3].

In addition, the attached utility software for VIVID700 is capable of filling the hole where the laser beam is not reflected from the object. Therefore, the extracted region can be filled fully with the range data.

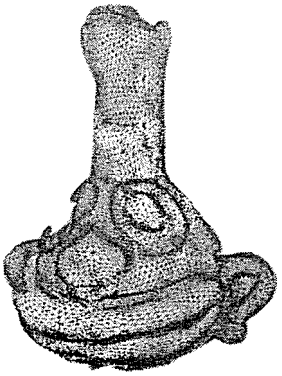


Figure 3. The original models reconstructed with the subdivision algorithm of Hoppe et al.

2.2 3D Shape Recovery

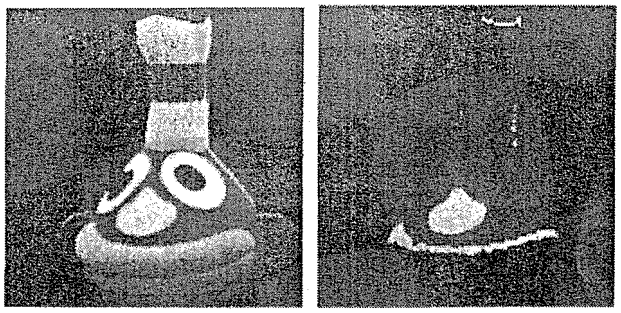
LRF merely gives a set of scattered points of a scanned object, which is typically a rectangular grid of distances from the sensor to the object. Therefore, it is necessary to connect these points in some way to represent the shape of the object, that is, modeling. Although there are many possible choices for modeling, among the most popular ones are subdivision algorithms, whose main advantages are their capability to generate smooth surfaces and recover sharp features such as creases and corners efficiently. In order to reconstruct scattered range data into a subdivision surface, we have chosen the subdivision method of Hoppe et al. [18] for our work. Figure 3 shows the original model of a doll recovered by suing the Hoppe method.

3 Detection of Object Feature

As mentioned above, the corresponding relationship between the 2D image and the 3D range data of a real object has been established in advance. Thus, we can rely on image processing to detect the features of an object according to its RGB values. The objective of image processing is to separate the components of an image into subsets that correspond to the local features of an object. For example, with respect to the doll indicated in Figure 2.a, the "nose" and "eye" are its particular features that present distinct colors from their surroundings. Hence, we can segment them from the "body" through image segmentation. In this section, we describe the steps to detect feature regions, and pick out the range data of the feature.

(1) Separating the target image from background

We refer to the image of Figure 2.a again. In the



(a) (b)

Figure 4. (a) Image of the target object, reduced to 200×200 dot (b) Extracted regions in thresholding

400×400 pixels image, besides the target object, unnecessary objects appearing as background are taken in as well. This makes it more difficult to find a feature region from the target object. Therefore, it is first necessary to separate the image of the target object from the background.

Before doing this, the corresponding relationship between the color and range data must be modified to a one-to-one correspondence. Since the color data $\mathbf{C}(I,J)$ in $(2i,2j)$ correspond to the range data $\mathbf{R}(i,j)$ in (i,j), we scale the image by applying the scale factors S_i and S_j to the coordinates I and J, respectively, as follows [19],

$$[i,j,1]=[I,J,1]\begin{bmatrix} S_i & 0 & 0 \\ 0 & S_j & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \begin{matrix} I,J\in[400,400] \\ i,j\in[200,200] \end{matrix} \qquad (1)$$

where $S_i=1/2$ and $S_j=1/2$. Hence, we can define the image value with $\mathbf{C}(i,j)=(R(i,j),G(i,j),B(i,j))$, using the same variable as range data.

Then, the single image of the target object can be separated from background by following the expression below.

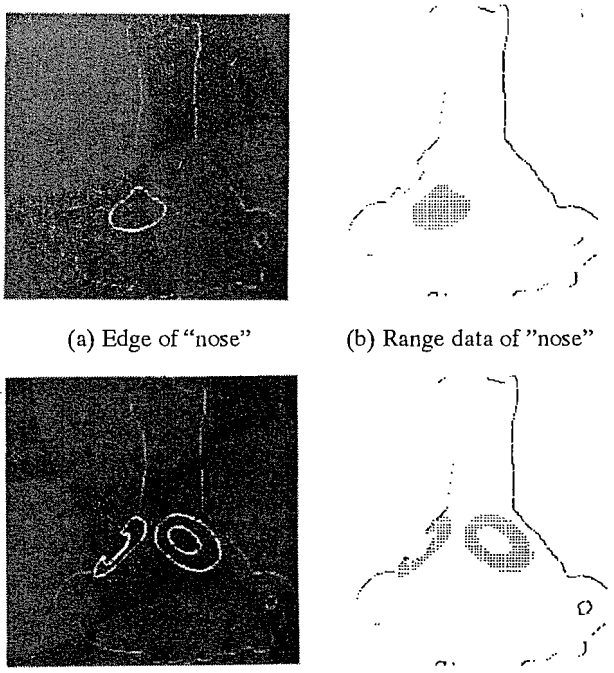
$$\mathbf{C}(i,j)=\begin{cases} \mathbf{C}(i,j) & \mathbf{Z}(i,j)\neq\infty \\ 0 & \mathbf{Z}(i,j)=\infty \end{cases} \qquad (2)$$

The values of a pixel where the range data are cut off are defined to zero. The resultant segmentation is shown in Figure 4.a.

(2) Region segmentation

In the color image of a multicolor object, connected sets of pixels corresponding to a feature or a local surface property commonly share a color attribute such as similar color. These sets give us the desired regions.

Region-oriented schemes may segment the image based on the similarity of pixel attributes. Here, we use the color thresholding algorithm to implement the region segmentation. However, finding a good threshold value for each of R, G, B components gives rise to some difficulties. Although a statistical classification based on a histogram can be helpful [20][21], it was found to be too computationally expensive and cannot provide satisfying results. In many cases, it is more convenient to specify the threshold value directly and manually. In this research, we have modified the traditional threshold method that specifies a unique value for a threshold to adopt an extent for the threshold. Successful performance of this method is achieved with the "trial and error" method. For example, we try to segment the "nose" from the image of the doll by specifying the range: R=130~190, G=110~160, B=0~20. The resultant image is shown in Figure 4.b, but unnecessary segmentations turn up because their color values are in the same threshold extent with the "nose."



(a) Edge of "nose" (b) Range data of "nose"

(c) Edge of "eye" (d) Range data of "eye"

Figure 5. The result of segmentation

In order to remove the extra segmentations, we then convert the segmented regions into intensity image following $I(i,j)=0.3R(i,j)+0.59G(i,j)+0.11B(i,j)$ [19]. Next, every connected set of pixels is labeled with a different number (i.e., labeling method) [19]. Therefore, the region of "nose" can be clearly identified according to its number.

(3) Edge detection

If the intensity of the image changes in a discontinuous way at the edge, the gradient algorithm is the most common one to detect the edge. The segmentation found above is perfectly suitable for this method.

Many gradient algorithms have been developed to approximate the gradient in a digital domain. The simplest method is differential calculus. In window

coordinates, the gradient $\partial I / \partial i$ and $\partial I / \partial j$ are computed by

$$I_i = \frac{\partial I(i,j)}{\partial i} = \frac{I(i+\Delta i, j) - I(i,j)}{\Delta i}$$
$$I_j = \frac{\partial I(i,j)}{\partial j} = \frac{I(i, j+\Delta j) - I(i,j)}{\Delta j} \qquad (3)$$

where $\Delta i = 1$ and $\Delta j = 1$. The gradient magnitude $\phi(i,j) = \sqrt{I_i^2 + I_j^2}$ expresses the strength of edge.

The detected edges are thick lines that are not entirely identical and have a width of more than two pixels. Thus, the edge is thinned to one pixel by eliminating the outside pixel [19]. The Figure 5.a and c show the edges of the "nose" and "eye."

Since there is correspondence between the color and range data associated with identical coordinates, the range data of the "nose" and "eye" can also be picked out easily on the basis of the image edge. The range data of the "nose" and "eye" are presented in Figure 4.b and d respectively.

4 Generating New Objects Based on the Original Model

Because the polygonal geometry of original range data is heavy enough to support a smooth curvature, it is inherently difficult to alter the geometric shape in a way that maintains the meaningful smoothness. For example, if we pull out a single vertex, we get a very sharp spike. Pulling lots of points independently to maintain curvature is also difficult. In order to create new meaningful shapes from original range data, the NURBS method is a valid method.

4.1 NURBS Surface Algorithm

The equation for a NURBS surface $\mathbf{S}(u,v)$ can be written as:

$$\mathbf{S}(u,v) = \frac{\sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,k}(u) N_{j,l}(v) \omega_{ij} \mathbf{P}_{ij}}{\sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,k}(u) N_{j,l}(v) \omega_{ij}} \qquad (4)$$

$$N_{i,1}(u) = \begin{cases} 1 & if \quad t_i \le u < t_{i+1} \\ 0 & else \end{cases}$$
$$N_{i,k}(u) = \frac{(u - t_i) N_{i,k-1}(u)}{t_{i+k-1} - t_i} + \frac{(t_{i+k} - u) N_{i+1,k-1}(v)}{t_{i+k} - t_{i+1}} \qquad (5)$$

where $u, v \in [0.0, 1.0]$; ω_{ij} are weights; $\mathbf{P}_{ij}$ are control points; $N_{i,k}(u)$ and $N_{j,l}(v)$ are normalized B-spline basis functions of degree k at the u direction and degree l at the v direction, and $n+1$ and $m+1$ are control points number in each direction, respectively; $N_{j,l}(v)$ is formed in the same way as $N_{i,k}(u)$ analogously; and t_i forms a knot vector $\mathbf{T}_u$ or $\mathbf{T}_v$. A detailed overview of NURBS algorithm can be found in Les Piegl's article and book [22][23].

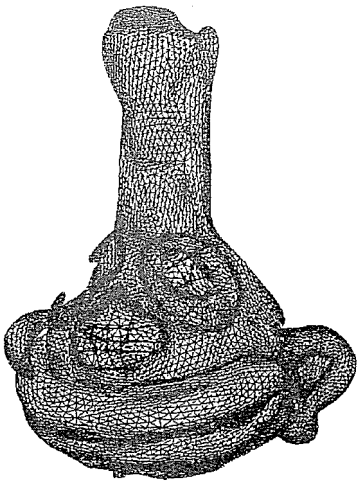


Figure 6. The primary control meshes and primary surface normal from ranged data of "nose" and "eye"

Especially, the parameters with respect to the weights and the knot vectors strongly influence the shape of a surface. Furthermore, they can be freely defined in a wide extent as long as certain definitions are followed. This flexibility enables the design of a large variety of shapes. For example, the scanned object in this research has been given new shapes that are natural-looking and funny.

4.2 Transformation of Range Data

As mentioned above, the purpose of this research is to transform the local range data of a real object by using the NURBS surface algorithm. The basic idea is to use new vertex data computed according to NURBS equation (4) to replace the original range data in the detected region. Then, the entire blended data are calculated into subdivision surface following the method of Hoppe et al. In this subsection, we describe the design of control meshes and the definition of weights and knot vector, and show the regenerated models.

4.2.1 Designing the Control Meshes

The calculation of an NURBS surface based upon the original range data differs from the generation of purely man-made model. First, we compute the unit normal vector of the figure surface modeled with triangle patches by using the range data in the feature region, whose holes are filled, as follows.

$$\mathbf{n}_0 = \frac{\sum_{i=1}^{N} (\mathbf{v}_1^i - \mathbf{v}_0^i) \times (\mathbf{v}_2^i - \mathbf{v}_0^i)}{\left| \sum_{i=1}^{N} (\mathbf{v}_1^i - \mathbf{v}_0^i) \times (\mathbf{v}_2^i - \mathbf{v}_0^i) \right|} \qquad \mathbf{v}^i \in R^3 , \qquad (6)$$

where $\mathbf{v}_0^i, \mathbf{v}_1^i$, and $\mathbf{v}_2^i$ are the vertices of triangle patches and N is the number of these patches. We call the $\mathbf{n}_0$ primary surface normal. Then, we select the part of the vertices on the figure surface as primary control points $\mathbf{P}_{ij}^0$ for a desired shape. In other words, $\mathbf{P}_{ij}^0 = \mathbf{R}(ii, jj)$, ii, and jj are in the domain of the feature region. As an example, the primary control meshes and surface normal of the "nose" and "eye" are illustrated in Figure 6.

With regard to primary control points $\mathbf{P}_{ij}^0$, the points on the edge of the feature region are always given as their boundary points in the u or v direction, and the boundary points are not varied in the process of constructing the control meshes. Therefore, the cloud of new surface data can be in as close agreement as possible with the range data outside the feature region at the boundary, although the edge curve of new surface has undergone a slight change.

We tweak the primary control points $\mathbf{P}_{ij}^0$ along the normal $\mathbf{n}_0$ or the accumulated direction from $\mathbf{n}_0$ to make the control meshes approach the desired form. The control points are obtained with

$$\mathbf{P}_{ij} = \mathbf{P}_{ij}^0 + \sum_{p=0}^{M} s_p \cdot \mathbf{n}_p \qquad (7)$$

where $\mathbf{n}_p$ are the tweaked directions of $\mathbf{P}_{ij}^0$, which is computed on the basis of their previous direction and the user-defined angle with each other; s_p are scalars that lead to the increase at $\mathbf{n}_p$ direction; M is the number of times that direction is changed from $\mathbf{n}_0$. The control meshes for the "eye", and "nose," for instance, are shown in Figure 7.

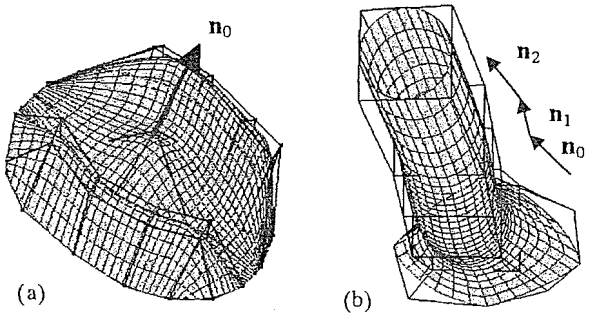


Figure 7. The control meshes and regenerated NURBS surface of the "eye(a)" and "nose(b)"

4.2.2 Weights and Knot Vector

The basic properties of an NURBS surface, such as smooth curvature and sharp crease, are generally specified by defining the weights and the knot vector. For example, if a particular weight is set at 0, then the corresponding control point has no effect at all on the surface. Conversely, if it is set at 1, the surface forms a

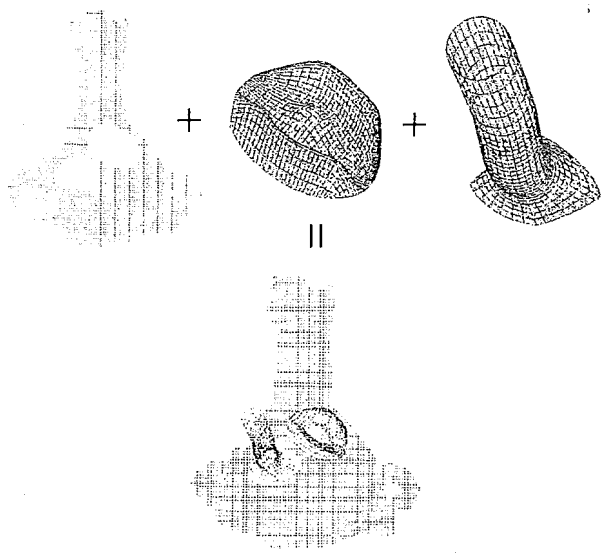
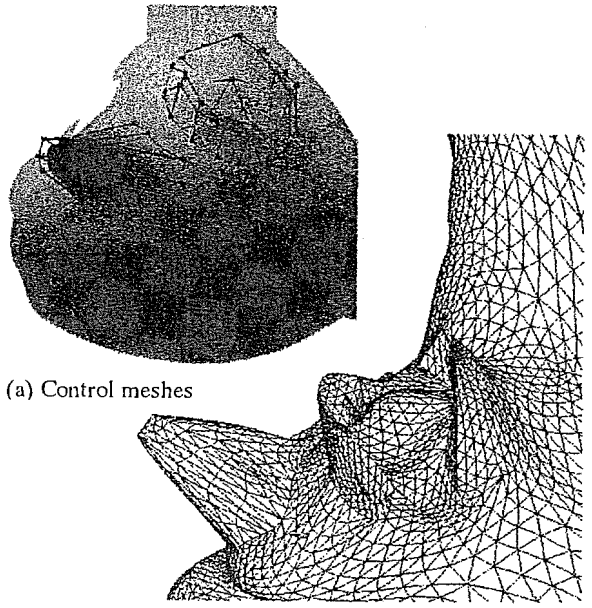


Figure 8. Blending rang data and NURBS surface data into new data set



(b) The result of transformed "nose" and "eye"

Figure 9. Constructed control meshes and transformed result

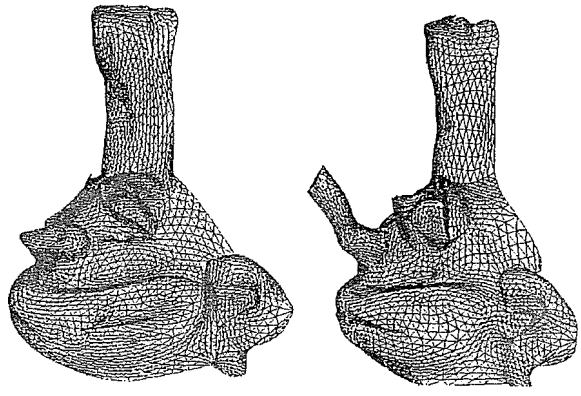


Figure 10. Two transformed models from different range data sets

convex shape towards the control point. Since the control mesh maintains a regular grid structure, we can vary the weight corresponding to those control points on any row or column from 0 to 1 so as to construct a smooth or sharp edge. Usually, we set the weight at 0.5.

In addition, the knot vector uniquely determines the B-spline as defined in equation (5). Each successive pair of knots represents an interval $[t_i, t_{i+1}]$ for the parameter values to calculate a segment of a surface. For simplicity, we always set the degree (k and l) of $N_{i,k}(u)$ and $N_{j,l}(v)$ to 3 and compute the sequence of knots in both directions as follows for a general surface, as shown in Figure7.a.

$$t_i = \begin{cases} 0 & i < k \\ i-k+1 & k \le i \le n \\ n-k+2 & i > n \end{cases} \qquad (8)$$

However, for traditional surfaces such as a cylinder, it is particularly necessary to define the knot vector, which is one of the drawbacks to an NURBS surface. When we create a cylinder as illustrated in Figure7.b, we set the knot vector in the u direction as above and in the v direction as $\mathbf{T}_v$={0,0,0,1,1,2,2,3,3,4,4,4}.

4.3 Regenerated Model

Having finished the calculation from equation (4), we obtain several sets of NURBS surface data for different shapes of the "eye" and "nose." Reconstructing these data into a surface with triangulated meshes prior and then connecting it with the other surface meshes modeled with the rest of the range data will cause new difficulty in connecting the patches at boundary due to modeling distortion. Moreover, it easily leads to surface seam and collision for animation.

In our method, the original data within the feature region are displaced by the NURBS surface data. Therefore, a blended data set for a redesigned model is newly reacquired. For the doll, the new data of "left eye" represented in Figure 7.a is used to replace the old one, and so is the new "nose" data (cf. Figure 7.b), as shown in Figure 8. Of cause, the NURBS data is in correct position.

The blended data set is reconstructed into a subdivision surface by the method of Hoppe et al. As a result, Figure 9.b shows an enlarged model that focuses on the "eye" and "nose." Figure 10 shows two complete models that present different shapes.

5 Texture Mapping

The reconstructed model is a complete subdivision surface comprised of complex triangulated meshes. Texture mapping can be used to improve the visual

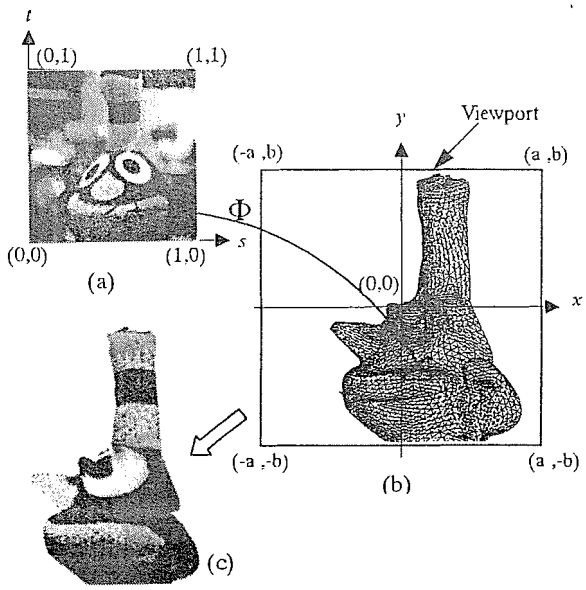


Figure 11. (a) The original texture image (b) The model constructed with triangulated meshes(rotated) (c) The textured model

richness of the surface. Although the use of texture mapping to complex surface has been widely discussed in the literature [24][25], our study presents a new technique challenge because the shape of model is strongly changed so as to make it more difficult to map texture to the surface without distortion. It is not our intent in this section to describe such texture mapping in detail. As an example, we make use of the original image shown in Figure 2.a to map the designed model with some distortions. The new generation of an object shows the same appearance with a different shape.

With regard to texture mapping, texture coordinates ($s,t \in [0.0,1.0]$) are assigned to the vertices of the triangle patches by using a mapping function Φ, as illustrated in Figure 11. Since the 3D surface is put in correspondence with a 2D image, we can define the function by simply considering the horizontal and vertical coordinates ($x \in [-a,a], y \in [-b,b]$) of model's vertices: $s = (x+\alpha a)/2\alpha a$, $t = (y+\beta b)/2\beta b$, α and β are used to modify the one-to-one correspondence between texture coordinate and world coordinate, which are determined according to the perspective and transformation in the view volume. The textured model is shown in Figure 11.c.

6 Conclusions and Future Work

Our study in the combined application of image segmentation and NURBS surface for transforming range data has been extremely positive. As a means to search the features of a real object, the image processing is efficient and simple. It is rather easy to determine the range data domain by mapping the 2D image edge to the

3D edge. The use of NURBS surface leads to the desired transformation of feature range data computationally and affords reliable results. Modeling an entire set of transformed range data with a subdivision surface not only produces excellent shape but also prevents seams and collisions between connected surfaces.

This algorithm can be used to design a new type of model from a real object for virtual reality and design manufacturable parts for CAD. Moreover, it can be used by the utility software of an LRF.

Because a variety of models can be generated from just one real object and their geometric shape is tremendously different from the original, mapping the texture captured from the real object to these models results in the considerable amount of distortion. Seeking a valid method for the non-distortion texture mapping is the future work.

References

[1] Y. Yanagida, S. Tachi: "Virtual Environment Construction Method Using Class Library (Japanese)," *Trans. IEE of Japan*, Vol.115-C,No.2, pp.236-244, February 1995.

[2] K. Ikeuchi, Y. Sato, Ko Nishino, I. Sato: "Modeling from Reality: Photometric Aspect (Japanese)," *Transactions of the Virtual Reality Society of Japan*, Vol.4, No.4, pp.623-630, 1999.

[3] L. Nyland, D. McAllister, et al: "The Impact of Dense Range Data on Computer Graphics," *http://www.cs.unc.edu/~ibr/pubs/nylandmview99/range_impact/*

[4] K. Pulli, M. Cohen, et al.: "View-based Rendering: Visualizing Real Objects from Scanned Range and Color data," *Proceedings of 8th Eurographics Workshop on Rendering*, pp. 23-34, June 1997.

[5] *http://www.minolta.com/japan/rio/vivid/index.html*

[6] J. Shade, S. Gortler, et al: "Layered Depth Images," *Computer Graphics (SIGGRAPH'98 Proceedings)*, pp.231-242, 1998.

[7] M. M. Oliveira, G Bishop: "Image-Based Objects," *Proceedings of 1999 ACM Symposium on Interactive 3D Graphics*, pp.191-198, April 1999.

[8] V. Blanz, T. Vetter: "A Morphable Model for the Synthesis of 3D Faces," *Computer Graphics (SIGGRAPH'99 Proceedings)*, pp.187-194, 1999.

[9] A. W. F. Lee, D. Dobkin, et al.: "Multiresolution Mesh Morphing," *Computer Graphics (SIGGRAPH' 99 Proceedings)*, pp.343-350, 1999.

[10]G. Turk, J. F.O' Brien: "Shape Transformation Using Variational Implicit Functions," *Computer Graphics (SIGGRAPH'99 Proceedings)*, pp.335-342, 1999.

[11]V. Krishnamurthy, M. Levoy: "Fitting Smooth Surface to Dense Polygon Meshes," *Computer Graphics (SIGGRAPH'96 Proceedings)*, pp.313-324, 1996.

[12]E. Catmull, J. Clark: "Recursively generated B-spline Surfaces on Arbitrary Topological Meshes," *Computer Aided Design*, 10(6), pp.350- 355, 1978.

[13]T. Derose, M. Kass, T. Truong: "Subdivision Surfaces in Character Animation," *Computer Graphics (SIGGRAPH'98 Proceedings)*, pp.85-94, 1998.

[14]A. Levin: "Interpolating Nets of Curves by Smooth Subdivision Surfaces," *Computer Graphics (SIGGRAPH'99 Proceedings)*, pp.57-64, 1999.

[15]*http://www.softimage.com/*

[16]*http://www.aliaswavefront.com/pages/home/index.html*

[17]S. Kumar, D. Manocha: "Interactive Display of Large NURBS Models," *IEEE Trans. on Visualization and Computer Graphics*, Vol. 2, No. 4, pp.323-336, December 1996.

[18]H.Hoppe, T. DeRpse, et al.: "Piecewise Smooth Surface Reconstruction," *Computer Graphics (SIGGRAPH'94 Proceedings)*, pp.295-302, 1994.

[19]D. H. Ballard, C. M. Brown: "Computer Vision," Prentice Hall Inc., Englewood Cliffs, New Jersey.

[20]P. V. Henstock, D. M. Chelberg: "Automatic Gradient Threshold Determination for Edge Detection," *IEEE Trans. on Image Processing*, Vol. 5, No. 5, pp.784-787, May 1996.

[21]S. S. Lee, S. J. Horng et al.: "Entropy Thresholding and Its Parallel Algorithm on the Reconfigurable Array of Processors with Wider Bus Networks," *IEEE Trans. on Image Processing*, Vol. 8, No. 9, pp.1229-1242, September 1999.

[22]L. Piegl: "On NURBS: A Survey," *IEEE Computer Graphics and Applications*, Vol. 11, No. 1, pp.55-71, January 1991.

[23]L. Piegl, W. Tiller: "The NURBS book," Springer-Verlag, New York, Second Edition, 1997.

[24]B. Lévy, J M. Mallet: "Non-Distorted Texture Mapping for Sheared Triangulated Meshes," *Computer Graphics (SIGGRAPH'98 Proceedings)*, pp.343-352, 1998.

[25]D. R. Peachey: "Solid texturing of complex surfaces," *Computer Graphics (SIGGRAPH'85 Proceedings)*, pp.287-296, 1985.

(2000年4月3日受付)