

Smoothing Method for Shape Measurement by Using a Low-Cost 3D Scanner^{*)}

Hiroshi MARUO, Ayumu SAITOH¹⁾, Tomohisa ISHIBASHI¹⁾ and Yuichi TAMURA

Konan University, 8-9-1 Okamoto, Higashinada-ku, Kobe, Hyogo 658-8501, Japan

¹⁾*University of Hyogo, 2167 Shosha, Himeji, Hyogo 671-2280, Japan*

(Received 6 December 2010 / Accepted 18 March 2011)

A 3D scanner is a device that measures 3D objects in the real world and converts them to 3D digital data. However, much noise and several loss areas (holes) are observed in the outputted data, especially when using a low-cost 3D scanner. The purpose of this study is to propose a method for noise removal, burying holes, and outputting a smooth 3D model. We propose a smoothing method employing the multi-level partition of unity implicit (MPU) method. This method can compute a continuous surface by using implicit functions, enabling noise and loss area removal. However, the output 3D data generated by this method is not often accurate because the MPU method is sensitive to noise. Therefore, before applying the MPU method, we applied a noise reduction process. In this study, we propose a system that can grasp smooth 3D object data by using a low-cost 3D scanner.

© 2011 The Japan Society of Plasma Science and Nuclear Fusion Research

Keywords: 3D scanner, implicit function, computer graphics, 3D restoration, virtual reality

DOI: 10.1585/pfr.6.2406112

1. Introduction

A 3D scanner is a device that measures objects in the real world and outputs their 3D shape digital data. Many 3D scanners have been proposed and commercialized; however, high-precision 3D scanners [1] having laser range sensors are very expensive. In this study, we employ the DAVID system [2] that can acquire 3D shape data by using a generic laser pointer and web camera. However, two major problems that arise in using the data obtained by this 3D scanner for designing are the noise generated (measurement errors) and the loss areas (holes) in the outputted results. The second error occurs because of the materials and the placement of the web camera and the object being measured. These errors cannot be avoided because the reflection ratios of materials and colors are different, and we cannot constantly keep rearranging the web camera position.

Some methods have been proposed to solve these problems. Studies for compensating the area using the similarity of objects [3–5] and for noise reduction have been conducted [6–8]. The DAVID system also has these functions, for example, “smooth average” and “smooth media”. However these functions do not consider the entire shape of a scanned object and changes correctly scanned data. Moreover the DAVID system has a function “fusion” which enables to remove overwrapped surfaces and points. However, this method is easily affected by noises. In this study, we propose a method for smoothing and compensating the lost area by applying noise removal process and

curved surfaces that are expressed using implicit functions. This smoothing method was originally applied to noiseless digital data, and the result obtained using this method is also affected by noises. Therefore, a large portion of the noise from the raw data outputted is removed before smoothing. The purpose of this study is to not obtain very accurate 3D data, but to obtain natural 3D data for use in designing prototypes. Moreover, this proposed method enables to change density of polygons in a 3D model since the entire 3D model is determined by implicit functions.

2. Procedure

Figure 1 demonstrates the processing procedure of our method. First, the shape is measured using a low-cost 3D scanner. Next, noise in the measured 3D model is removed using the feature of the polygon. Finally, implicit function curved surfaces are applied to the output model, and thus, the 3D shape model is obtained.

In this study, our apparatus comprises a PC (Intel Core2Duo E6850, 2.99 GHz, 2 GB RAM), a USB web

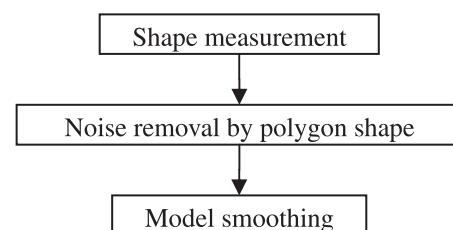


Fig. 1 Processing procedure.

author's e-mail: mn024015@center.konan-u.ac.jp

^{*)} This article is based on the presentation at the 20th International Toki Conference (ITC20).

camera (DIGITAL COWBOY/DC-NCR20U) with VGA resolution, a laser pointer (LE650-5-3-F-S) which emits a red laser line, and a robot arm (ELEKIT/MR-999CP) for moving the laser pointer.

2.1 Shape measurement

In this study, we employ the DAVID system for obtaining 3D digital data. Figure 2 shows how the 3D digital data was obtained using this system. Before measuring, it is necessary to calibrate the web camera. The measuring space consists of two boards that should be arranged vertically. Several markers are printed on these boards and the size and position of the markers are already known. The web camera is calibrated by scanning the positions of these markers. By employing this procedure, the position and posture in this coordinates is determined and the position of the boards and the web camera should be fixed.

After that an object is positioned between the two boards, as shown in Fig. 2. The laser pointer that is attached to the front of the robot arm (Fig. 3) emits a laser beam. This laser beam is projected on the object and both the boards. If there is no object in the measuring space, the two lines reflected from the two boards are projected on the web camera. On the other hand, if there is an object in this space, the rays are reflected from the surface of the object, from the boards, and are then projected on the web camera.

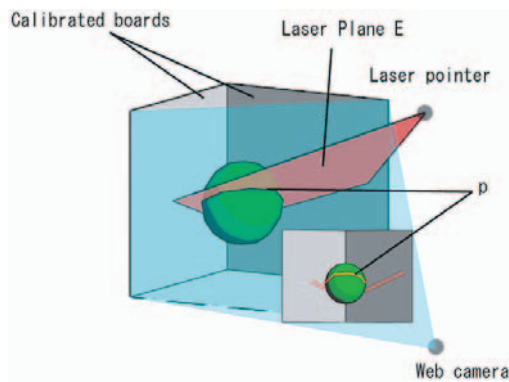


Fig. 2 System configuration of the DAVID system.

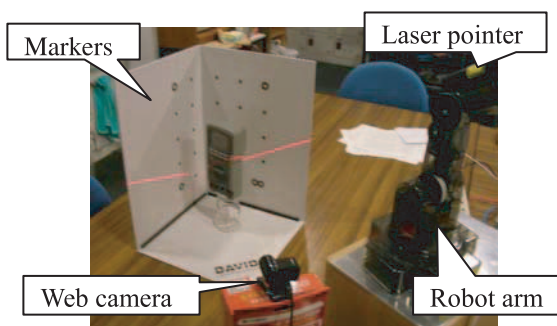


Fig. 3 Scanning environment.

In the aforementioned calibration process, the position and posture of the web camera is already known. Thus, the laser plane E (Fig. 2) is determined. Once the laser plane E is determined, the absolute 3D position of point p can also be determined from the web camera image. If we measure the entire area of the objects by this process, it is necessary to measure from various angles and positions by moving the camera and the object. In our research, we used the robotic arm for moving the laser pointer. This is because the wavering of the user's hand greatly influenced the measurement results. By repeating this process changing the angle of the object, the whole shape can be measured with not only the 3D positions of the vertices but also the polygons and normal vectors.

2.2 Noise removal by polygon shape

Figure 4 shows an example of the measurement results, which primarily includes much noise (Fig. 4 (c)). It is impossible to perform the smoothing process because the noise influences the final shape. Therefore, much of the noise is removed before smoothing. Figure 4 (a) shows the result of correctly scanning the 3D data and Fig. 4 (b) shows typical noise obtained by the David system. The feature of this noise is that one point of a polygon, including the noise, is apart from two other points and the shape resembles that of an isosceles triangle having long edges. For this noise removal, the following processing was applied.

The triangle consists of 3 points p_1 , p_2 , and p_3 . The edges are determined by Eq. (1), where L_i is the distance

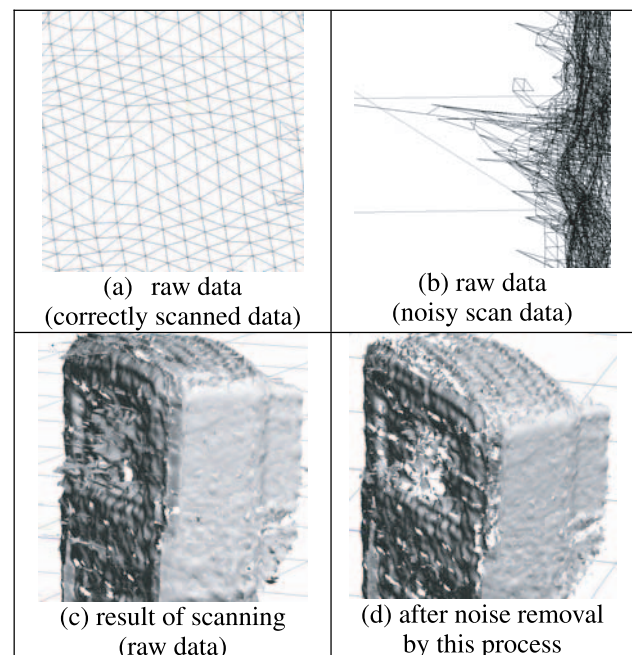


Fig. 4 Scanned raw data and the results of the noise removal process. ((a), (b), (c) are output data from DAVID and (d) is the model after noise removal process.)

of the opposite edge from point p_i .

$$\begin{cases} L_1 = \|p_2 - p_3\| \\ L_2 = \|p_1 - p_3\| \\ L_3 = \|p_1 - p_2\| \end{cases} \quad (1)$$

In our noise removal process, if Eq. (2) holds, the point p_j is considered to be noise.

$$\max(L_k) > S \min(L_j) \quad \begin{cases} k = 1, 2, 3 \\ j = 1, 2, 3 \end{cases}, \quad (2)$$

where S is a scale parameter for determining whether it is noise or not. In this research, we defined parameter $S = 2.0$. This parameter was obtained by conducting exploratory experiments. If this parameter is too big, noise removal becomes impossible; and if it is too small, the correct measured points are removed. There is some noise generated in the post-processed model; however, removing too much noise makes the holes too big; therefore, this residual noise is removed in the next smoothing process. Figure 3 (d) shows the result after this process.

2.3 Smoothing by MPU method

We employ the multi-level partition of unity (MPU) [9] method for smoothing the object. The MPU method is one method for computing 3D surface from point data using implicit function curved surfaces. The implicit function is determined as follows. There are n points with coordinates (x_1, y_1, z_1) , $(x_2, y_2, z_2) \dots$, and (x_n, y_n, z_n) ; these points are on the surface D. If Eq. (3) holds for function f , then

$$f(x_i, y_i, z_i) = 0 \quad (i = 1, 2, \dots, n). \quad (3)$$

A curved surface C: $f(x, y, z) = 0$ is an implicit function of the surface D.

It is possible to express an implicit function with all of the vertices; however, it takes a long time to determine a suitable surface. We then employ the MPU method that divides a point data model into two or more spherical cells, computes its suitable implicit function, and then connects these functions continuously. This enables us to shorten the processing time. If there is more number of point data than the threshold in one cell, the cell is divided. We use the implicit function, as expressed by Eq. (4).

$$f(x) = \sum_{i=1}^M \phi_i(\|x - c_i\|)Q_i(x), \quad (4)$$

where $Q_i(x)$ and $\phi_k(x)$ were determined by Eqs. (5) and (6), and M is the number of cells.

$$Q_i(x) = \begin{cases} x^T A x + b^T x + c & (\min(n_A \cdot n) \leq 0) \\ w - (a'u^2 + b'uv + c'v^2 + d'u + e'v + f') & (\min(n_A \cdot n) > 0) \end{cases} \quad (5)$$

$$\phi_k(\|x - c_i\|) = \begin{cases} \left(1 - \frac{\|x - c_i\|}{R}\right)^2 & (0 \leq \|x - c_i\| \leq R) \\ 0 & (\|x - c_i\| > R) \end{cases} \quad (6)$$

where n_A is the average vector of the normal vectors of n points in one cell; R is the radius of the cell and c_i is the center point of the i -th cell; and w, v and u are the axes of the local coordinate system in one cell. For computing the optimal solution, we minimize a variable ε (Taubin distance [10]) defined by Eq. (7). This shows the approximation error between the point on the implicit function surface and the point data obtained from measurements. By minimizing ε , parameters $A, b, c, a', b', c', d', e'$ and f' are calculated.

$$\varepsilon = \max_{\|p_N \in p_b\|} \frac{Q(p_b)}{\|\nabla Q(p_b)\|}. \quad (7)$$

Finally, the implicit function surface is determined and the smoothed 3D shape data is obtained.

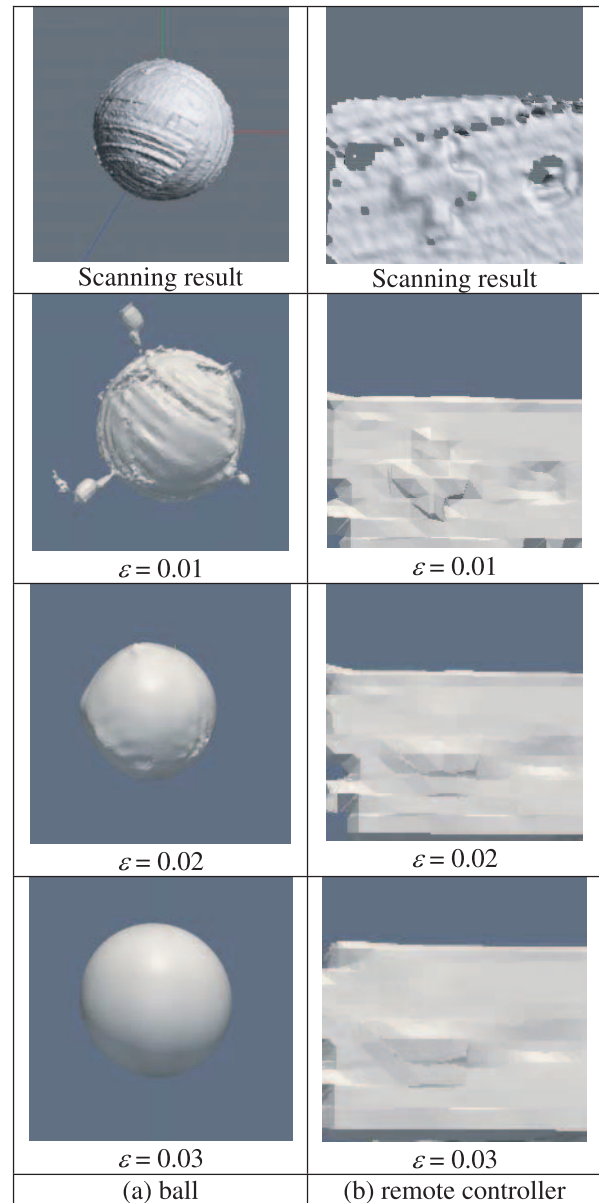


Fig. 5 Processing results.

Table 1 Processing time.

Model	Ball	Remote controller
Number of vertices of raw data	95436	23371
Number of vertices after noise removal	16098	12888
Number of polygons of raw data	187230	45185
Number of polygons after noise removal	32192	25776
Processing time for noise removal	58.13 s	2.65 s
Processing time for smoothing by MPU method	70.00 s	45.36 s

3. Estimations

The main aim of our study is not to obtain highly accurate 3D data, but to obtain a 3D model that can be used in a virtual environment or in concept design. Therefore, we did not estimate from the viewpoint of the accuracy of the outputted data. In this section, we show the results of estimation in terms of the processing time and their dependency on the estimation ε . In this estimation, we use two models. One model is a ball (Fig. 5 (a)) and the second is a remote controller device, which includes buttons and some ruggedness (Fig. 5 (b)).

Table 1 lists the number of vertices and polygons of the measured data and the processing time of these models. Figure 5 shows the raw data obtained from the DAVID system and the processing results when the approximation error margin ε is varied. In the ball model, when ε is small, there are some small objects outside the ball object. This occurs when some noise exists apart from the ball model. In such situations, one implicit function is made in a local area and this function is connected continuously to the ball model. Because of this, a number of small balls are generated. On the other hand, if ε is large, the implicit surface is not made from points in a local area. If the error factor is too large, the implicit function is generated from the surface point of the ball and a point apart from the ball surface. This decreases the accuracy of the output model but can make a natural model.

On the other hand, in the controller model, if the error factor is increased, the shape is too smooth and the features

of the model are lost. As a result of this estimation, this process can make the smooth surface by adjusting the error factor; however, this depends on the original features of the 3D model.

4. Summary

In this study, we propose a method for measuring 3D shape data by using a low-cost 3D scanner, smoothing the 3D shape data, and outputting natural 3D data. There is much noise in the output from a low-cost 3D scanner. It is difficult to smooth the surface of the obtained data by means of a general smoothing method. Therefore, we roughly remove the noise and then smooth the surface of the 3D object by employing the MPU method.

In our estimation, the processing time of the model, which consists of approximately 100,000 vertices, is approximately 2 minutes. This shows the possibility of the practical use of this method. And from a result of estimations, by adjusting the error factor, we can obtain a natural 3D output by using this method. It is unfit to be used in a detailed design process; however, it can be used as a 3D object in a virtual environment or in the concept design process.

- [1] <https://www.nextengine.com/>
- [2] S. Winkelbach, S. Molkenstruck and F.M. WahlSven, Lecture Notes in Computer Science: Pattern Recognition **4174**, 718 (2006).
- [3] N. Kawai, T. Sato and N. Yokoya, Proc. IEEE Inc. ICIP, 1532 (2008).
- [4] V. Kraevoy and A. Sheffer, Proc. Eurographics Symp. on Geometry Processing, 13 (2005).
- [5] M. Pauly, N.J. Mitra, J. Giesen, M. Gross and L. Guibas, Proc. Eurographics Symp. on Geometry Processing **255**, 23 (2005).
- [6] M. Desbrun, M. Meyer, P. Schröder and A.H. Barr, Proc. SIGGRAPH, 317 (1999).
- [7] C. Öztireli, G. Guennebaud and M. Gross, Comput. Graph. Forum **28**, 493 (2009).
- [8] T.R. Jones, F. Durand and M. Desbrun, TOG Proceedings of ACM SIGGRAPH **22**, 943 (2003).
- [9] Y. Ohtake, A. Belyaev, M. Alexa, G. Turk and H.-P. Seidel, Proc. SIGGRAPH, 173 (2005).
- [10] G. Taubin, IEEE Trans. Pattern Anal. Mach. Intell. **13**, 1115 (1991).