

Color Segmentation Based Depth Adjustment for 3D Model Reconstruction from a Single Input Image

Vicky Sintunata and Terumasa Aoki

Abstract—In order to create a good 3D model reconstruction from an image, the shape information of the object of interest is needed. One such information that yields the shape information of an object from images is the shading information. This information can further be utilized using the Shape from Shading method. Unfortunately this method has some drawbacks, namely the depth ambiguity problems. Two points in the image which have the same intensity will have the same depth, although in reality they have different depth. Color also plays a huge effect to this ambiguity. Some color generates higher intensity than the others. This makes the system generate an incorrect reconstruction of the image. We propose a color segmentation based depth adjustment to overcome this problem. First we cluster the image into color clusters. Based on this color cluster we will adjust the depth of vertices belonging to the higher intensity cluster accordingly. The proposed method can reduce the effect of these depth ambiguity problems.

Index Terms—3D model reconstruction, shape from shading, color segmentation.

I. INTRODUCTION

3D reconstruction from image has long been studied in the field of computer vision. Based on the number of input images, 3D reconstruction can be separated into two types: a single input image and multiple input images. Multiple input images reconstruction can reconstruct object in a more detail results, but suffers from rectifying and correlation problems. Multiple images taken from multiple angles are also required to get a good reconstruction, which makes it inconvenient to use. On the other hand, a single input image will be much easier for the user to use. The challenge of a single input system is limited information regarding the overall shape of the object to be reconstructed.

Given an image of an object, human can determine the overall shape of the object. Sketch-based Interfaces for Modeling (SBIM) is one of the fields which exploit how people perceive an object from its silhouette. Like drawing on a paper, users can draw directly on the system and the system will generate the 3D model of the drawn object. One of the most well-known SBIM systems is Teddy [1]. The main input of the system is the silhouette of the object which is drawn by the users. Unfortunately, SBIM system usually draws a rounded 3D model due to the elevation method. A better solution is to use another method which depends not only to the silhouette of the object but another shape clues. One of the clues in shape perception can be acquired through shading

information. Shape from shading (SFS) uses the intensity information of an image to approximate the shape.

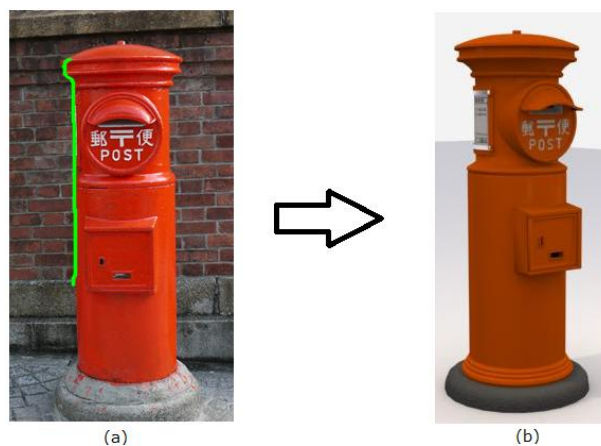


Fig. 1. User traces the contour in the image in green line(a) and the resulting 3D output (b).

Unfortunately, SFS has some drawbacks, which are the depth ambiguity problems. One of these problems is some parts of the object in the image with the same intensity value will have the same depth. Another problem is regarding the color intensity.

The main goal of our research is to combine the advantages of SBIM where user can decide which objects to be reconstructed and to solve the problems mentioned above (Fig. 1). We propose an algorithm based on the color segmentation method. First we divide the image into several color cluster based on its color intensity. Then, we adjust the depth of the surfaces belonging to the higher intensity cluster to reduce the color intensity influence. Finally in order to make the final output smoother, some simple smoothing operation is done.

The rest of this paper is organized as follows. Section II will briefly describe the related work on this field. Section III will describe problems description. Section IV will describe the proposed method in more details. Section V will show some of the results of the proposed method and some discussion. Section VI will conclude this paper.

II. RELATED WORK

As already mentioned in the previous part, human uses some clues in order to perceive the shape of an object. Shape from X method is a method which uses these clues to mime how people perceive an object and then reconstruct an object from the input image. Depending on the input of the algorithm, X can be anything from, but no limited to, texture, shading,

Manuscript received October 9, 2014; revised February 10, 2015.

The authors are with the New Industry Hatchery Center (NICHe) of Tohoku University, Sendai, Miyagi Prefecture 9808579, Japan (e-mail: vicky@riec.tohoku.ac.jp, aoki@riec.tohoku.ac.jp).

etc.

SFS was first developed by Horn. SFS is used to derive 3D scene description from one or more 2D images [2]. The main input for SFS is the intensity value of the input images, which usually can be retrieved from the gray level of the input image. Lambertian surface, orthographic projection and known light source position are some of the assumptions usually used in solving SFS problems.

Reference [3] uses linear approximation in solving SFS problem and [4] uses triangular element surface model and linear approximation of the reflectance model. Reference [5] considers SFS as an optimization problem from some available information (shading) therefore they propose a Genetic Algorithm based method to solve it.

Reference [6] uses perspective projection assumption rather than the usual orthographic projection in order to improve the SFS algorithm. Reference [7] uses Oren-Nayer reflectance model rather than the usual Lambertian model. Similar with our method, [8] also uses color information in order to improve the SFS algorithm. First, they use a chromaticity-based specular reflection removal algorithm to make a pure diffuse image of the object of interest. Then they apply some color adjustment to generate an image without surface changes, which will be used as the input of the SFS algorithm. Interested reader can also refer to survey papers [2] and [9] for more details regarding SFS algorithms.

As we know, some color generates more intensity than the others. Most of the paper mentioned above does not consider color influence to the intensity values which actually has an effect to the reconstructed object. Therefore, we would like to address this problem.

III. PROBLEMS DESCRIPTION

As mentioned in part I, shape from shading suffers from some drawbacks. Consider Fig. 2. A through E are five points on the surface of the object (a sphere cut by a plane) and N_A , N_B , N_C , N_D , and N_E are the normal direction of each point. Assume that the light source position has the same position with the camera position.

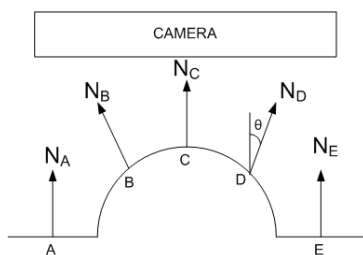


Fig. 2. Surface example of a sphere and a plane.

Since shape from shading assumes Lambertian surface model, the intensity captured by the camera will follow (1).

$$I_c = I_i \cos\theta_i . \quad (1)$$

I_c is the intensity captured by the camera, I_i is the intensity reflected by the surface at point i , and θ_i is the angle between the normal direction at point i and the light source.

As can be seen, point A, C, and E will have the same intensity value captured by the camera although they have

different depths. The reason is because they have the same normal direction, which will make (1) have exactly the same value. This will make point A, C, and E reconstructed with the same depth value or has many interpretation such as illustrates in Fig. 3.

Another depth ambiguity arises because of the color information. Some colors generate higher intensity compared to other colors. Consider Fig. 4. As can be seen, the words part of the input image, since they have yellow color, generate a higher intensity compared to the background (red). This characteristic will influence the depth approximation, which will make an incorrect result. Fig. 5 is the rendered image using only (1).

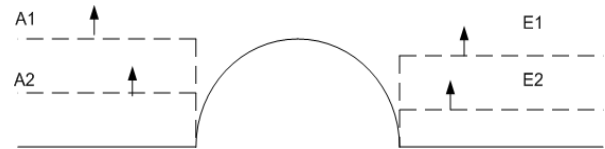


Fig. 3. Multiple interpretations of a surface.



Fig. 4. Input image (left) and intensity image of input (right).



Fig. 5. Rendering of input image: front view (left) and side view (right).

IV. PROPOSED METHOD

In this part, we will explain our proposed method. There are some steps in our method (refer to Fig. 6). The first step is to make base meshes. This is done in order to reduce the area to be computed and to make sure that the object of interest will be able to be reconstructed completely.

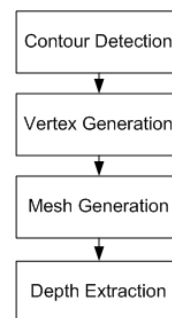


Fig. 6. General steps of proposed method.



Fig. 7. Input image examples.

Before explaining the steps in more detail, there are some

assumptions used in this paper. First is the input image for the proposed method. The object of interest of the input images used in this paper is assumed to have been extracted from the background. This assumption is used mainly for the automatic contour detection method (explained below). An example of the input image can be seen in Fig. 7. We also assume that the light source has the same position as the camera position. Intensity value will also be considered to be the depth of the surfaces.

A. Contour Detection (Tracing)

Contour detection (tracing) can be done in two ways. The first way is manually done by the user, while the second way is to use an automatic contour detection. This is done by using Canny Edge detection. Some parts might be disconnected during the extraction therefore we also add dilation method to fill in the missing part. Next, we keep only the external contour data. Fig. 8 illustrates this step.

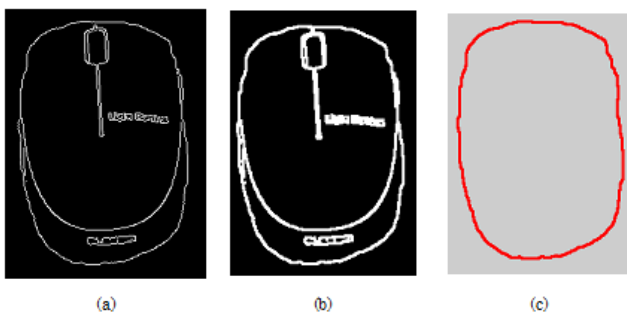


Fig. 8. Contour tracing (automatic) step: (a & d) output of canny edge detection, (b & e) output of dilation process, (c & f) external contour.

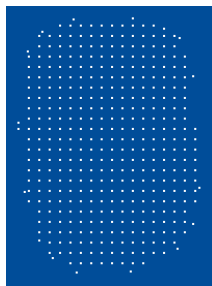


Fig. 9. Output of vertex generation step.

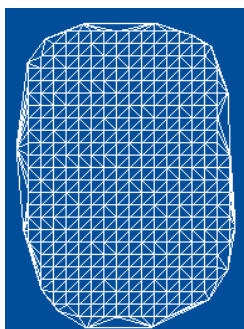


Fig. 10. Result of mesh generation.

B. Vertex Generation

Contour data from the previous step is a set of points. In order to simplify the contour data which will be the base for the mesh generation step, we use the Douglas Peucker algorithm. We also add vertices inside the contour by using the grid method. The distance between one point with the others on the grid is 10. Fig. 9 illustrates the output of this

step.

C. Mesh Generation

In order to generate the mesh, similar with [4], triangular mesh is created. This is done using Constrained Delaunay triangulation. The output vertices of the Douglas Peucker algorithm will be considered to be the constraint. This means, no triangles will be generated if it lies outside the external contours. The resulting output of this step can be seen in Fig. 10.

D. Depth Extraction

This step is the heart of our proposed method. In order to extract the depth information, intensity value of the input image is used. Intensity value might be acquired from the grey value of the input image (RGB), but this will raise a problem mentioned in part III. Therefore, we use the grey level image taken from HSV color space, particularly the V as the intensity value, rather than the grey level of RGB color space.

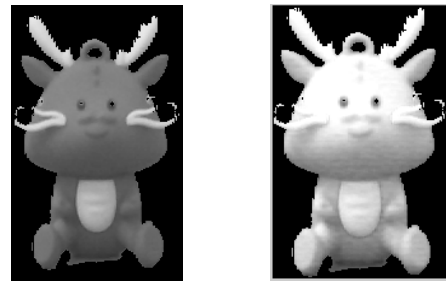


Fig. 11. Grey level of RGB color space (left) and HSV color space (right).

Fig. 11 compares the grey image of the input image in RGB color space and HSV color space. As can be seen, horns, whiskers and stomach part of the object, which have yellow colour, generate higher intensity in RGB color space. On the contrary, HSV color space produces better output in terms of intensity (horns part and face part are blend equally).

Two steps are done in depth extraction step: the flat region search and the cutting algorithm (depth adjustment).

1) Flat region search

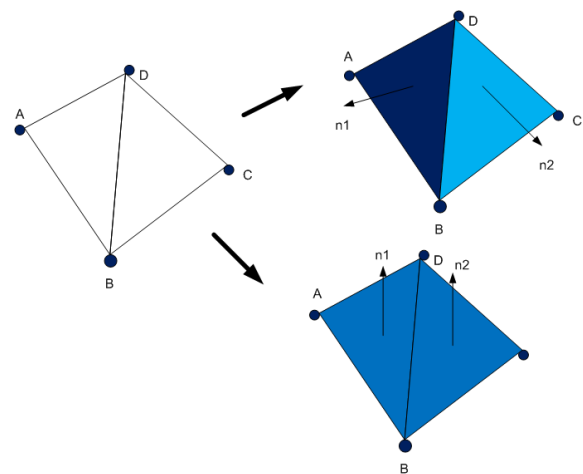


Fig. 12. Flat and non-flat surfaces.

This is done in order to reduce the effect of the depth ambiguity. First, we determine whether a triangle mesh represents a flat surface or not. This is done by checking each vertex of the triangles whether they have the same intensity

value or different intensity value. A triangle which has three identical intensity values at each vertex is considered to represent a flat area. See Fig. 12.

Let triangle ABD and BCD be two adjacent triangles. There are two interpretations of these triangles. One can be seen on the upper part of Fig. 12 and another on the bottom part of Fig. 12. n_1 and n_2 are the normal directions of each triangle respectively. By looking at the normal direction of the triangles, we can determine whether these two triangles combine into a flat surface or non-flat surface. Again, based on (1) the intensity value of a triangle will depend on its normal direction, and therefore, by searching for adjacent triangles which have the same intensity will give us an intuition whether they create a flat surface or not.

Then, we group vertices which have the same intensity. We then find vertices which were connected directly with the other vertices that do not have the same intensity.

This step can be summarized in the following algorithm:

- 1) For every vertex do:
 - Check the neighbors
 - If neighbor has the same intensity, do nothing
 - Else, keep the neighbor's intensity value
- 2) If all vertices have been examined, calculate the average of the intensity values that were kept.
- 3) Replace all vertices' depth belonging to the same intensity group with the average intensity values that were kept.

2) Cutting algorithm (depth adjustment)

Before doing the cutting algorithm, we divide the input image into 3 color clusters. This is done using k-means clustering method with k value equals 3. This step is required for the next step, where we will reduce the depth of each vertex according to its color clusters. Fig. 13 illustrates the output of k-means color clustering.

We then generate first the vertices lying on the medium intensity value color cluster. Next, we reduce the depth (intensity) of each vertex belonging to the higher intensity color cluster. As mentioned before, yellow color has higher intensity than red color. Therefore, we will reduce vertices' depth belonging to the yellow color cluster. This is done by firstly calculating the average difference of intensity between the vertices on the higher intensity group (yellow) and the other vertices which have direct connections with the higher intensity group. Fig. 14 illustrates this step.

Point A, B, E, and F belong to the lower intensity cluster. Point C and D belong to the higher intensity cluster. Δ can be defined as follows:

$$\Delta = \frac{\sum I_{ij}}{n} \quad (2)$$

where:

$$I_{ij} = \begin{cases} |I_i - I_j| & \text{if there is connection between } i \text{ and } j \\ 0 & \text{otherwise} \end{cases}$$

n = number of paired ij vertices connected with each other

It should be noted that (2) only applies to pairs with different clusters. For example, point A and point B will not be considered for calculating Δ even though there is a

connection from point A to point B (Fig. 14, illustrated with line connecting those points) and so do point C and D, point E and F. Only B-C pair and D-E pair will be considered when calculating Δ .

After calculating the average difference, every vertex belonging to the higher intensity cluster will be reduced with Δ . Continuous line in Fig. 14 illustrates output generated from this step. See that the depth of point C and D are reduced with Δ .

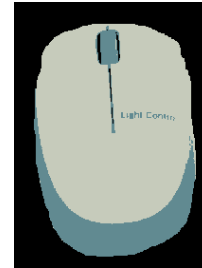


Fig. 13. Clustering output.

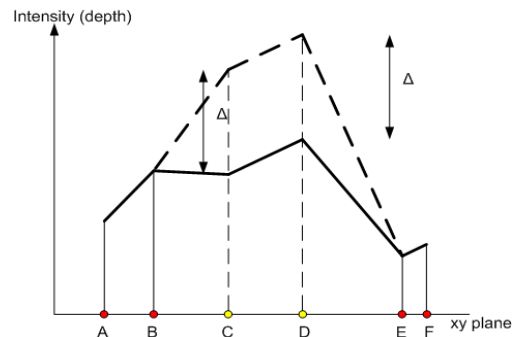


Fig. 14. Depth adjustment method.

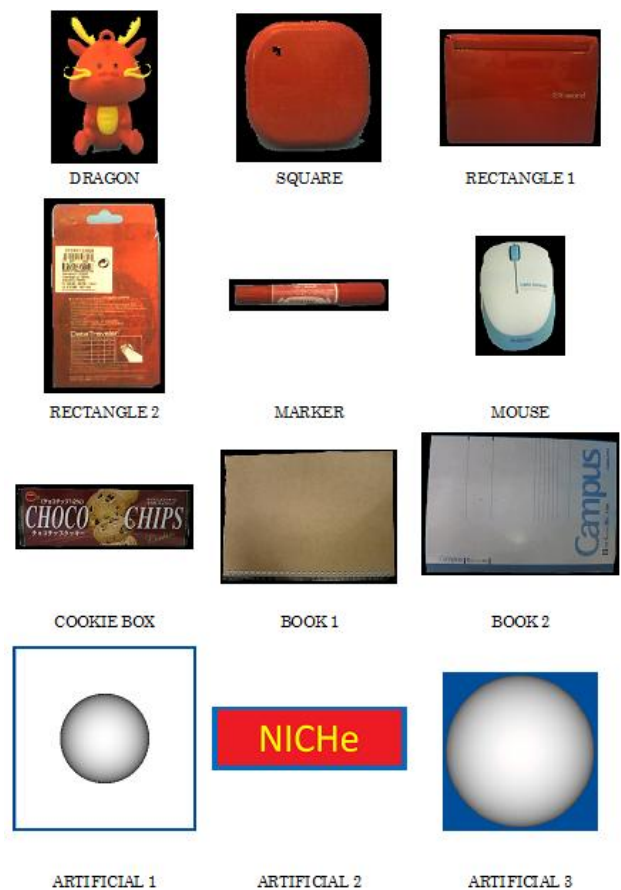


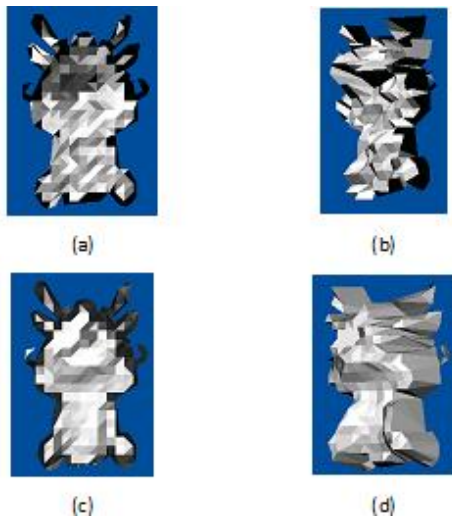
Fig. 15. Input images.

Final step in this method is to smooth the last reconstruction. This is done by applying the median filter. A search circle is applied to every vertex, where the vertex of interest will be the center of the search circle. Every vertex lying inside the radius of the circle will be considered to be chosen as the final depth for the vertex of interest. The median value of the vertices lying inside the circle will be the final depth for the vertex of interest.

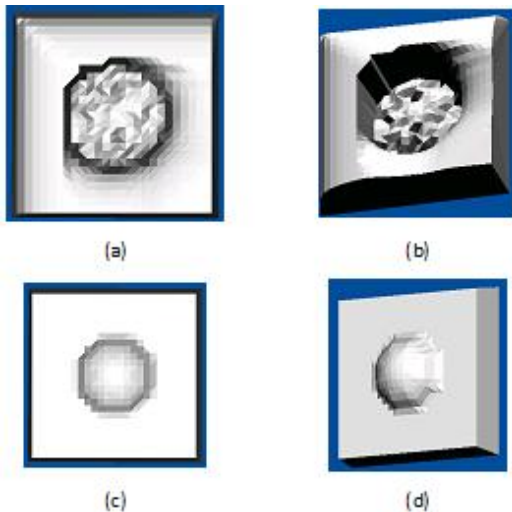
V. RESULT

The proposed method was implemented in C language, OpenCV library support, OpenGL library support, and Triangle [10] program for applying the constrained Delaunay Triangulation during mesh generation. Size of the images used is 640×480 . Twelve input images are used in this paper. See Fig. 15. Three of the input images used are artificial images and the rest are natural images. The intensity value of each vertex ranged from 0 to 1, therefore we multiply the value with 100. 50 and 150 are used as the parameters for canny threshold respectively. As for the median filtering, we use 15 for the radius of the circle.

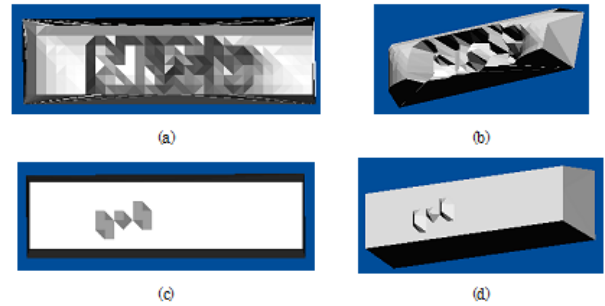
A limitation in our current algorithm is regarding the number of objects to be reconstructed. Currently only one object can be reconstructed using this algorithm. Fully black-colored object will not also be able to be reconstructed correctly.



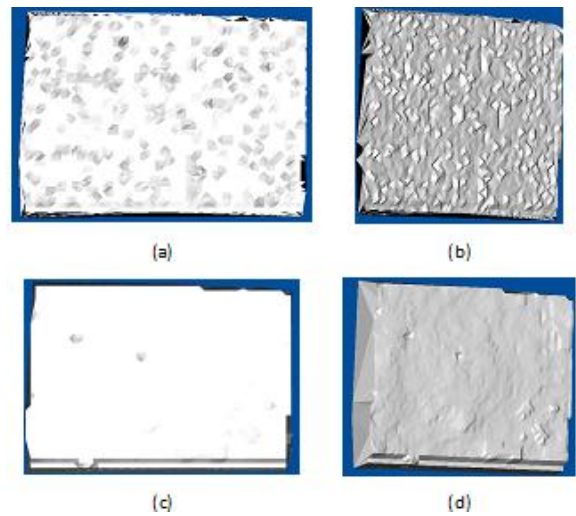
SFS output: (a) front view, (b) side view; Proposed Method output: (c) front view, (d) side view
Fig. 16. Dragon comparison.



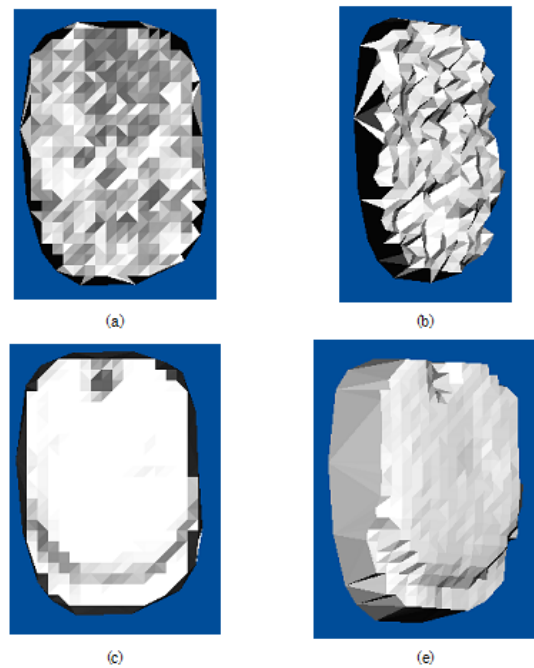
SFS output: (a) front view, (b) side view; Proposed Method output: (c) front view, (d) side view
Fig. 17. Artificial 1 comparison.



SFS output: (a) front view, (b) side view; Proposed Method output: (c) front view, (d) side view
Fig. 18. Artificial 2 comparison.



SFS output: (a) front view, (b) side view; Proposed Method output: (c) front view, (d) side view
Fig. 19. Book 2 comparison.



SFS output: (a) front view, (b) side view; Proposed method output: (c) front view, (d) side view
Fig. 20. Mouse comparison.

The result of our proposed method (depth extraction method) is compared with the SFS algorithm as in [2]. Three comparisons are presented as in Fig. 16–Fig. 20. As can be seen the proposed method generates a better result, especially for the artificial 2 example (Fig. 17). Depth ambiguity in this example can be reduced using the proposed method. Also can be seen from Fig. 17, depth ambiguity due to color effect can be reduced.

VI. CONCLUSION

In this paper we propose an algorithm to reduce the effect of depth ambiguity in shape from shading. By using our proposed method, depth ambiguity of SFS can be reduced as can be seen in part V. In this proposed method, a global depth adjustment is used. A further research regarding the local depth adjustment will be our main focus.

REFERENCES

- [1] T. Igarashi, S. Matsuoka, and H. Tanaka, "Teddy: A sketching interface for 3d freeform design," in *Proc. the 26th Annual Conference on Computer Graphics and Interactive Techniques*, 1999, pp. 409-416.
- [2] R. Zhang, P. S. Tsai, J. E. Cryer, and M. Shah, "Shape from shading: a survey," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 21, issue 8, pp. 690–706, 1999.
- [3] P. S. Tsai and M. Shah, "Shape from shading using linear approximation," *Image and Vision Computing*, vol. 12, issue 8, pp. 487-498, 1994.
- [4] K. M. Lee and C. C. J. Kuo, "Shape from shading with a linear triangular element surface model," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 15, no. 8, pp. 815-822, 1993.
- [5] H. Saito and K. Usami, "Shape from shading using genetic algorithm," *Industrial Electronics, Control, and Instrumentation*, vol. 3, pp. 1620-1625, 1993.
- [6] E. Prados and O. Faugeras. "Shape from shading: a well-posed problem?" *Computer Vision and Pattern Recognition*, vol. 2, pp. 870-877, 2005.
- [7] G. Wang, S. Liu, J. Han, and X. Zhang, "A novel shape from shading algorithm for non-lambertian surfaces," *Measuring Technology and Mechatronics Automation*, vol. 1, pp. 222-225, 2011.
- [8] X. Zhang, Y. Gao, and T. Caelli, "Colour adjustment and specular removal for non-uniform shape from shading," *Digital Image Computing Techniques and Applications*, pp. 563- 568, 2010.
- [9] J. D. Durou, M. Falcone, and M. Sagona, "Numerical methods for shape from shading: a new survey with benchmarks," *Computer Vision and Image Understanding*, vol. 109, issue 1, pp. 22-43, 2008.
- [10] J. R. Shewchuk, "Delaunay refinement algorithm for triangular mesh generation," *Computational Geometry: Theory and Applications*, vol. 22, pp. 21-74, 2002.



Vicky Sintunata was born in Bandung in 1988. He received his bachelor degree in industrial engineering from Parahyangan Catholic University, Bandung, Indonesia in 2011. He received his master degree in information science from Tohoku University, Sendai, Miyagi Prefecture, Japan in 2014. Currently he is pursuing his doctorate degree at Tohoku University.



Terumasa Aoki is an associate professor at NICHe (New Industry Creation Hatchery Center), Tohoku University, Japan. He received his B.E, M.E, and Ph.D degrees from University of Tokyo in 1993, 1995, and 1998 respectively. He has received various academic excellent awards such as Young Scientist Award from MEXT (the Ministry of Education, Culture, Sports, Science and Technology in Japan) in 2007, Yamashita Award and the Best Education Award from IPSJ (Information Processing Society of Japan) in 2001 and 2007 respectively, two Best Paper Awards from IIEEJ (the Institute of Image Electronics Engineers of Japan) in 2004 and 2009 etc. He is well known as the developer of DMD (digital movie director) and NeoPoster etc. His current research topics are digital content technology, especially image processing, computer vision, and computer graphics.