

Taking Consensus of Signed Distance Field for Complementing Unobservable Surface

Ryusuke Sagawa
Institute of Scientific and Industrial Research,
Osaka University
8-1 Mihogaoka, Ibaraki-shi, Osaka,
567-0047, JAPAN
sagawa@am.sanken.osaka-u.ac.jp

Katsushi Ikeuchi
Institute of Industrial Science,
University of Tokyo
4-6-1 Komaba Meguro-ku, Tokyo,
153-8505, JAPAN
ki@cvl.iis.u-tokyo.ac.jp

Abstract

When we use range finders to observe the shape of an object, many occluded areas may occur. These become holes and gaps in the model and make it undesirable to utilize the model for various applications. We propose a novel method to fill holes and gaps and complement such an incomplete model. We use a signed distance field (SDF) as an intermediate representation, which stores Euclidean signed distances from a voxel to the nearest point of the mesh model. Since the signs of a SDF become unstable around holes or gaps, we take a consensus of the signed distances of neighbor voxels by estimating the consistency of the SDF. Once we make the SDF consistent, we can efficiently fill holes and gaps.

1 Introduction

Recently, many researchers have studied modeling the shape of real world objects by scanning them using three dimensional digitizers, such as laser range finders [5, 12] and structured-light range finders [15]. They measure distances from the point of view to the surface of the object which can be seen from the sensors. Thus, to acquire the whole shape of an object, we have to scan it from various viewpoints by using those sensors. If the object has an intricate shape, many occluded areas may occur. Consequently, there are often unobserved surfaces, even when full use is made of the various kinds of sensors. However, it is too costly to cover every hole which is not observed by any range images by taking range images from various viewpoints. In the worst case scenario, we are not able to obtain the data needed to create a complete model. However, we want to fill such holes to make use of the constructed models in many different applications, such as creating a solid model and visualizing the model. Therefore, we propose a new method that enables us to complement the geomet-

ric models by estimating the neighborhood area of the holes and filling the holes and gaps. Since filling the holes of a model is a major issue in this field, several approaches have previously been proposed. The simplest approach is interpolation by triangulating the boundary vertices of a hole. If the hole is small and the topology is simple, the triangulation works well; however, triangulation becomes difficult if the surface is intricate and the hole is large. The second approach is to fit a mesh model around the hole, that is, a three-dimensional version of snakes [10]. In these methods [7, 3], a deformable surface moves iteratively to fit the model with satisfaction of the smoothness constraint. Since they determine the topology a priori, they are not suitable for intricately-shaped objects. The third approach is called “space carving.” Curless and Levoy [4] tag one of the states, unseen, empty and near the surface, to each voxel during the merging process. The hole filling is accomplished by generating a surface between voxels of unseen states and those of empty states. Since space carving methods do not consider viewing objects from other viewpoints, the result of rendering from different viewpoints may be far from acceptable. The fourth approach is interpolation by volumetric representation, such as level-set approaches [18, 19] and re-computation of the implicit surface [2, 6]. Davis et al. [6] re-compute the implicit surface by diffusing the signed distance function from the vicinity of the observed surface to the whole volume; however, this method sometimes fails in high curvature areas because it does not consider the consistency of the isosurface.

Our method is similar to the third and fourth approaches. In our framework, we compute the signed distance field (SDF) using multiple range images. Since we separate the entire volume into two manifolds by SDF, we can generate a closed surface by converting SDF to a mesh model. However, the sign of SDF becomes unstable around the holes. Thus, we cannot interpolate the holes to a suitable surface

by applying the merging method straightforwardly. In this paper, we thus propose a novel method to interpolate the holes of range images by taking a consensus of the SDF sign with those of the neighbor voxels.

2 Convert from Range Images to SDF

Some methods of merging range images [4, 17, 18, 9] use SDF as an intermediate representation of the surface, and there are several ways to compute a signed distance. For example, Curless [4] used the distance between a voxel and the point of a range image along the line of sight of the range image. Wheeler [17] used the closest point of a range image from the center of a voxel to compute the signed distance.

In this paper, we assume that the magnitude of a signed distance is the Euclidean distance from the center of a voxel to the closest point of range images. It is necessary to take a consensus of the signed distances of neighbor voxels. Our merging algorithm [13, 14] is based on the consensus surface algorithm [17], and it computes a signed distance by finding the closest point of range images by taking a consensus of range images and discarding outlier points. Since the magnitude of the signed distances of our method is Euclidean distances, we use the SDF computed by our merging algorithm as the input of the new method described in this paper. Naturally, we can apply our new method the SDF created by other methods if the SDF satisfies this assumption.

In our merging algorithm, the sign of a signed distance is determined by considering the normal vector \mathbf{n} of the closest point, which looks outside and is computed by averaging the normal vectors of the triangles to which it belongs, and the vector \mathbf{v} from the center of the voxel to the closest point. Thus, the signed distance d is computed as

$$d = \text{sgn}(-\mathbf{n} \cdot \mathbf{v})|\mathbf{v}|, \quad (1)$$

where $\text{sgn}(x)$ is 1 if x is positive and is -1 if x is negative.

Moreover, our merging algorithm proposed in [13] uses voxels of adaptive resolution in an octree manner. In an area which is far from range images, we sample the 3D space coarsely and use voxels of large size. If range images are nearby, we sample the 3D space more finely. The width of voxel W is determined by the magnitude of the signed distance:

$$W < \frac{2}{3\sqrt{3}}|d|. \quad (2)$$

An example of SDF in adaptive resolution is shown in Figure 2(d). Since the range images are near the green voxels, the resolution becomes coarse according to the distance from range images.

3 Instability of Computing Sign of SDF

In this section, we point out the problem of computing a signed distance. Often, a mistake occurs in computing the

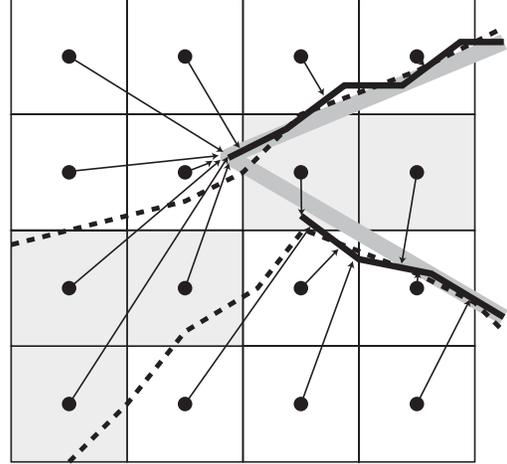


Figure 1. Corruption of the surface caused by wrong sign of SDF

sign of a signed distance. This especially happens where the surface of the object is highly curved. Figure 1 shows an example of computing signed distances when there is high curvature surface of an object. The gray line denotes the real surface of the object. The black solid lines indicate the range images. In this situation, we regard every point of the range images as having consensus. Each arrow is the vector from the center of a voxel to the nearest point. If a voxel is inside the object by considering normal vectors of range images, the sign is negative, and the voxel is filled by the color gray in Figure 1.

In a high curvature area, it is difficult to cover both sides of a corner with only a single range image. Thus, we need several range images to cover the entire corner; and to align both sides into a common coordinate system. Although the lower left area is obviously outside in Figure 1, the signs of those voxels are negative because they are considered to be inside by the normal vector of the nearest point. We use the marching cubes algorithm (MC) [11] to convert the SDF to a mesh model. The generated surface corrupts, as represented by the dotted lines. The signs of the signed distances become unstable to noise around the boundary of holes and gaps for the following three reasons:

1. The relative positions of range images after aligning each other involve errors caused by the errors of measurement.
2. It is difficult for range images to overlap one another at the point of sharp corners.
3. There are surface holes which are not covered by any range image.

First, since the signs of SDF can change unstably due to a small error of the position of a range image, the shape of

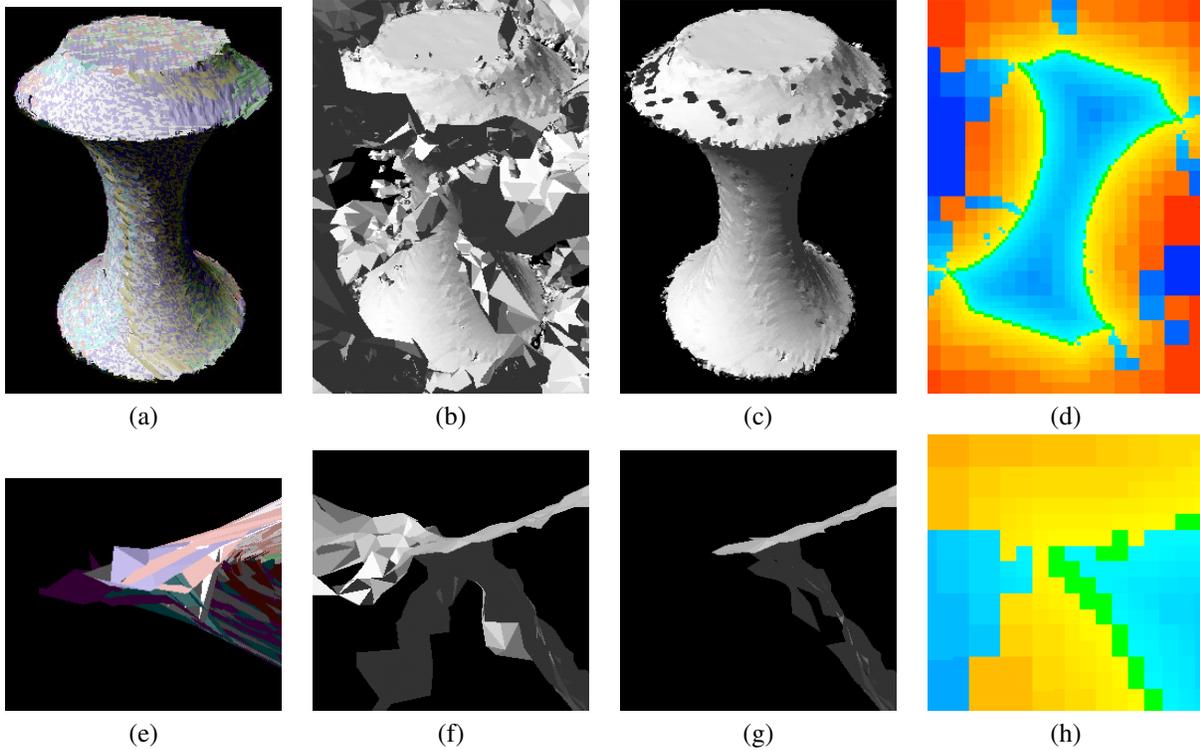


Figure 2. Merging results of an object with sharp corners

the generated mesh model is greatly affected by a change of signs of SDF. Second, because a single range image cannot cover both sides of a sharp corner, the sign of a voxel changes according to the range image which is used to compute SDF. Moreover, a gap between range images often occurs due to the insufficient sampling interval of each range image. Thus, it is difficult to determine whether the voxel is inside or outside of the surface. Third, it is often difficult to acquire range images to cover the whole surface in many cases. If a hole exists, the sign of SDF can be unstable around the hole.

Figure 2(a) shows the range images of an object which has sharp corners. Figure 2(e) is a zoom-up of one of the sharp corners. Due to the above reasons, it is hard to determine whether a voxel is inside or outside near the corner. Consequently, the result of converting SDF to a mesh model by MC is shown in Figure 2(b) and (f). Since the signs of SDF of some voxels are wrong, many vertices and triangles are generated outside of the object.

If the magnitude of a signed distance is smaller than the width of the voxel, the nearest point is inside the voxel. Thus, the generated vertices and triangles are not far from the real surface. Therefore, if we restrict the magnitude of SDF to apply the SDF to MC comparing the magnitude with the width of the voxels, erroneous meshes are not generated. If we use the voxels whose signed distances are smaller than

their width, the merging result becomes Figure 2(c) and (g). Erroneous meshes are removed; however, holes in the surface remain where the magnitude of the signed distances was larger than the width of voxels.

Figure 2(d) and (h) are 2D slice of the SDF. Yellow and red voxels indicate that they have positive signed distances, and blue voxels indicate negative ones. The magnitudes of the signed distances of red voxels are larger than those of yellow voxels. Also, blue voxels have larger magnitudes than light-blue voxels. And green voxels are near range images. When we generate a mesh model even if the iso-surface is between red and blue voxels, we get the former erroneous mesh model. On the other hand, if we generate only a mesh model near the green voxels, we get the latter mesh model with holes. To interpolate holes of the surface, a method is needed to determine the signs of SDF which are resistant to errors caused by the above reasons.

4 Consistency of the SDF

Our merging algorithm takes a consensus of range image points for outlier detection; however, it does not consider a consensus of the signed distances of adjacent voxels. In this section, we introduce a new criterion to estimate whether the signed distances of adjacent voxels are consistent with each other.

Figure 3 shows three examples of the signed distances of adjacent voxels. The width of the voxels is 1.0. In Figure

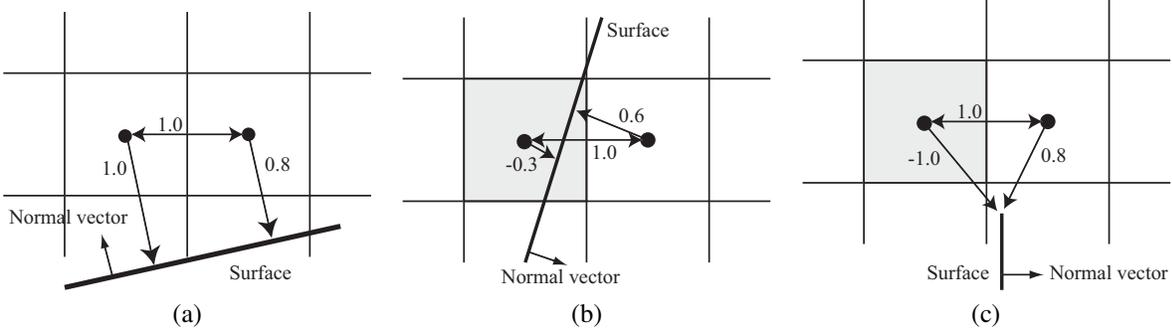


Figure 3. Three examples of signed distances of adjacent voxels

3(a), both voxels are outside of an object whose signed distances are 1.0 and 0.8. If the signs of the adjacent voxels are the same, the isosurface does not exist between them. Since the arrows from the center of the voxel to the surface represent the nearest neighbor points and the normal vector looks outside, it is possible that both signed distances are the same, as shown in Figure 3(a). Thus, they are considered to be consistent with each other.

In Figure 3(b), the signs of the adjacent voxels are different. The signed distances are -0.3 and 0.6 . The gray voxel means that it has a negative signed distance. An isosurface exists between two voxels. Since it is possible to compute the situation from a surface as shown in Figure 3(b), the signed distances of the two voxels are considered to be consistent.

In Figure 3(c), the signs of the adjacent voxels are different, similar to Figure 3(b). The difference from Figure 3(b) is that the signed distances are -1.0 and 0.8 . By considering the SDF of these two voxels, an isosurface exists between them; however, if the original surface existed between them, the sum of the magnitude of the two signed distances would be smaller than the width of the voxel. Thus, the original surface does not exist between the two voxels in this case, and the situation is considered to be inconsistent to generate meshes between them. The inconsistent situation such as Figure 3(c) occurs for the reasons described in Section 3. If the signed distances of two adjacent voxels satisfy the following inequality, we define the situation as inconsistent:

$$|d - d'| > W, \quad (3)$$

where d and d' are the signed distances of two adjacent voxels. Consequently, if there is an inconsistency of SDF, the sign of the signed distance is doubtful, and we examine the situation to generate a mesh model which is consistent with the original surface.

5 Flip Sign of SDF

Our new method proposed in this paper takes a consensus of the signed distances of adjacent voxels. We have already introduced a criterion by (3). We compared a voxel

with its adjacent voxels. Since our merging method uses voxels of adaptive resolution in an octree manner, the width of the adjacent voxel can be different. Thus, (3) is modified to

$$|d - d'| > \alpha \frac{W + W'}{2}, \quad (4)$$

where W is the width of the current voxel, W' is the width of the adjacent voxel and α is a parameter defined by the user. We usually use $\alpha = 1.0$.

We compare a voxel with all adjacent voxels. If the sizes of adjacent voxels are same, the number of adjacent voxels in a 3D space is 26. The simple solution to determine whether the voxel is inconsistent or not is to count the number of adjacent voxels N which satisfies (4). If N is sufficiently large compared with the total number of the adjacent voxels, we regard that the voxel is inconsistent. However, we empirically determined that the convergence is too fast and the surface remains unstable when we use this criterion. Therefore, instead of counting N , we count the following numbers of adjacent voxels N_1, N_2, N_3, N_4 which satisfies

$$\begin{aligned} N_1 : & |d - d'| \leq \alpha \bar{W} \\ N_2 : & |d - d'| > \alpha \bar{W} \\ N_3 : & |-d - d'| \leq \alpha \bar{W} \\ N_4 : & |-d - d'| > \alpha \bar{W} \end{aligned} \quad (5)$$

where $\bar{W} = (W + W')/2$. If these numbers become

$$N_2 + N_3 > \beta(N_1 + N_2 + N_3 + N_4), \quad (6)$$

We consider that the sign of the signed distance of the voxel is inconsistent. Then, we flip its sign to make it consistent with the adjacent voxels. We usually use $\beta = 0.5$. We compute the consistency as for all voxels, and flip its sign if it is inconsistent with its adjacent voxels. We iterate this process until all voxels are consistent.

In some situations, since it is not always $|d - d'| > \alpha \bar{W}$ when $|-d - d'| \leq \alpha \bar{W}$, the signs of the flipping voxels may oscillate and the number of flipping voxels may not converge to 0. In such cases, we increase α or β and relax the condition during iteration if the convergence becomes slow.

After one iteration, we can restrict the voxels to examine their signs. If the signs of a voxel and its adjacent voxels did not flip in the previous iteration, we do not have to examine it in the current iteration. Thus, we record the voxels whose signs were flipped, and use the database to determine whether we need to examine a voxel. For details, refer to [14].

6 Analysis with Level Set Formulation

In this section, we analyze our method with level set formulation [16]. In the level set formulation, the zero level isosurface $f(\mathbf{x}) = 0$ evolves according to the following partial differential equation:

$$\frac{\partial f}{\partial t} + F|\nabla f| = 0, \quad (7)$$

where F is the speed function. Then, the approximation using a discrete voxel indexed by (i, j, k) at time n becomes

$$\frac{f_{ijk}^{n+1} - f_{ijk}^n}{\Delta t} + F|\nabla_{ijk} f_{ijk}^n| = 0, \quad (8)$$

where Δt is the time step. If we interpret our method to one of the level set method, our method can be represented by using the following speed function F :

$$F = \begin{cases} -\frac{2f_{ijk}^n}{|\nabla_{ijk} f_{ijk}^n| \Delta t} & (6) \text{ is satisfied} \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

We consider only the derivatives of f_{ijk} where the SDF satisfies (6) because we make only unstable surfaces propagate.

If we consider the curvature κ of the isosurface, which is computed by $\kappa = \nabla \cdot \frac{\nabla f}{|\nabla f|}$, the above condition of F is identical to

$$F = \begin{cases} \text{sgn}(\kappa) \frac{2f_{ijk}^n}{|\nabla_{ijk} f_{ijk}^n| \Delta t} & |\kappa| > \epsilon \\ 0 & \text{otherwise} \end{cases} \quad (10)$$

where ϵ is a parameter, because κ and f_{ijk}^n have the same sign when (6) is satisfied. Figure 4 shows two examples of the situations. In this figure, the characters $+$ and $-$ denote the sign of SDF. We can see the sign of the signed distance of the center voxel is same with the curvature κ .

If the speed function is $F = -\kappa$, which is used in the heat equation, the highly curved surface is smoothed out and the surface gets closer to a sphere. At this point, since the sign of F used in our method is same with that of $F = -\kappa$, we can say that the surface evolves to be a uniformly curved surface compared with adjacent voxels with keeping smoothness on the boundary of holes by our method. Since a voxel far from a surface is larger than one near a surface, the curvature far from the original surface gets smaller than that near a surface.

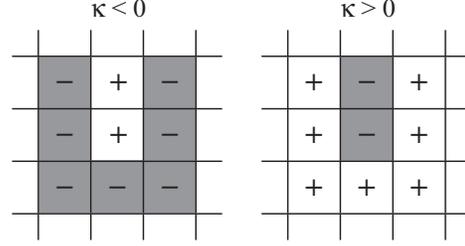


Figure 4. The sign of the signed distance is same with the curvature κ .

As described in the previous section, the SDF may oscillate. The reason is regarded to be that the magnitude of F does not depend on the curvature. Thus, it is necessary to adjust parameters α and ϵ to stop oscillation. In our practical method, we adjust β instead of ϵ . If $\beta = 0.5$, we accept only a completely planar surface around the voxel, and if we increase β , it is namely that we accept a curved surface.

7 Experiments

We experimented having our proposed method implemented using Intel Xeon 2.4GHz processor with 1GB memory. We first tested our algorithm to take the consistency of signed distances with adjacent voxels using a small object. In Figure 2, the signed distances around sharp corners are unstable to noises. We applied our new method to the SDF of the object, and obtained the model shown in Figure 5. Since we took the consistency of the signed distances of the adjacent voxels, we removed erroneous vertices and triangles from the generated mesh model at the same time as we filled holes on the original surface (see Figure 5(a),(c) and (d)). We can see that our method successfully removed the wrong signs from Figure 5(b) and (e), which are 2D slice of the SDF after applying our method.

Now, we compare our method with Davis’s method [6]. First, we converted the mesh model of this object shown in Figure 2(c), which has a lot of holes, to the volumetric representation of Davis’s method by a tool provided by Allen [1]. We adjusted the parameter so that the size of voxels after conversion was the same as the original one. Figure 6 shows the result of hole filling by the method. Figure 6(a) and (b) are the resulting mesh model after 100 and 300 iterations, respectively. There are improbable surfaces around the sharp corners even after 300 iterations. Figure 6(d) is a slice of the volumetric representation of (b). Since it is obvious that the implicit surface around the top-left corner is collapsed, the correct mesh model is not generated. We suppose that the reason for this is that the method does not consider the consistency of the isosurface.

The SDF used by our method is divided to $128 \times 128 \times 128 (= 2^{21})$ voxels in the finest resolution; however, since we use voxels of adaptive resolution, the total number of

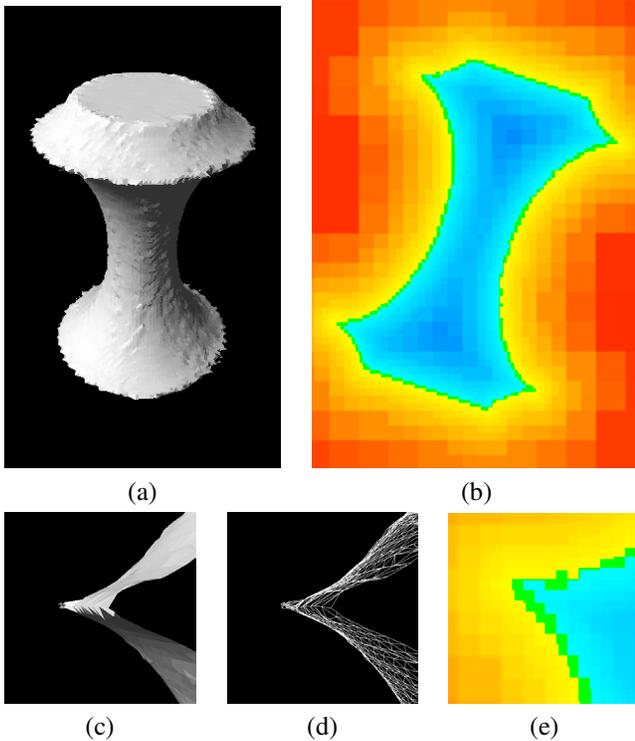


Figure 5. Result of taking consistency of the SDF

voxels is 130,521. We constructed a consistent SDF by 20 iterations. In this experiment, we set α to 1, and increased β according to $\beta_{n+1} = 1.01\beta_n$ at time n , if the number of flipped voxels did not decrease. We started with $\beta = 0.5$ and finally β became 0.52. The number of flipped voxels was reduced quickly at the beginning, and it was much smaller than the total number of voxels. Thus, the reduction of computation by using the database of flipped voxels worked effectively. The total time of our method was 5.72 seconds for 20 iterations, while Davis’s method took 51.4 seconds for 300 iterations.

Next, we experimented hole filling of the unobserved surface of the Great Buddha of Kamakura. Before taking the consistency of the SDF, we got models shown in Figure 8. Figure 8(a) is a 2D slice of the SDF, and (b) and (c) are rendering results of the SDF directly by volume rendering [8]. Since the range images have a lot of holes, which are shown in Figure 8(d),(e) and (g), undesirable surfaces are generated if we use all inconsistent voxels (Figure 8(f) and (h)).

After taking the consistency of the SDF, we obtained the model shown in Figure 9. Figure 9(a) is a 2D slice of the SDF and (b) and (c) are results of volume rendering. We could now successfully fill holes of the model by converting the SDF to a mesh model. Figure 9(d),(f) and (h) were rendered with triangles, and (e) and (g) are the wire-frame representations. Because we used voxels in adaptive resolu-

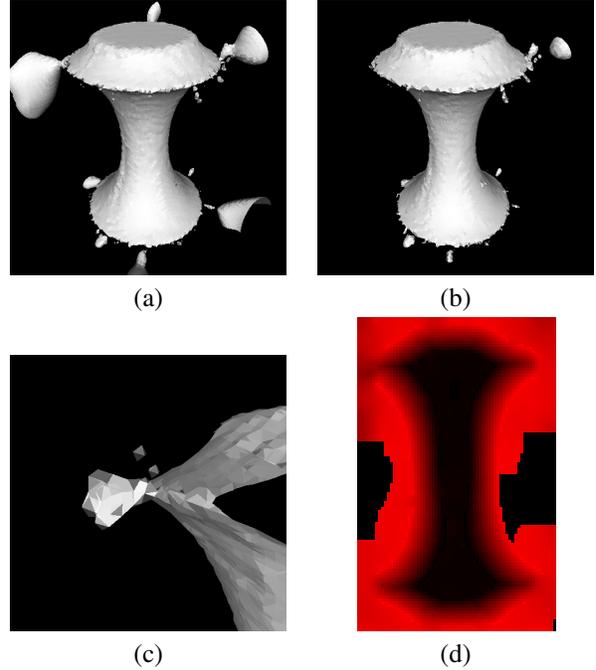


Figure 6. Result of hole filling by Davis’s method

tion, large triangles were generated to fill large holes using marching cubes with voxels of adaptive size [13].

The SDF of the Buddha model has $1024 \times 1024 \times 1024$ voxels in the finest resolution while the actual number of voxels is 17,024,273. Figure 7 shows the relationship between the number of flipped voxels and the time for each iteration, α and β ; the scales of the number of voxels and computational time are logarithmic. After the first iteration, the computational time was drastically reduced because we restricted voxels to be examined by the method described in Section 5. In this experiment, we increased both α and β according to $\alpha_{n+1} = 1.05\alpha_n$ if $N_n \geq 0.95N_{n-1}$, and $\beta_{n+1} = 1.01\beta_n$ if $N_n \geq N_{n-1}$, where N_n is the number of flipped voxels at time n . We gradually relaxed the condition by increasing α because the number of flipped voxels would be reduced much too fast if we increased only β . Flipping signs was iterated 97 times and the total computational time was about 14 minutes. The memory was used about 550MB at maximum. The number of triangles and surface area after complementing the model was 5383549 and 402.75m^2 while those of the original model were 5241486 and 329.23m^2 . (The height of the Buddha is 11.3m.) Thus, our algorithm efficiently fill holes with little additional triangles.

8 Summary

We have proposed a novel method to fill holes of a 3D model. Though the signed distance field, which is computed from an incomplete mesh model, has inconsistencies

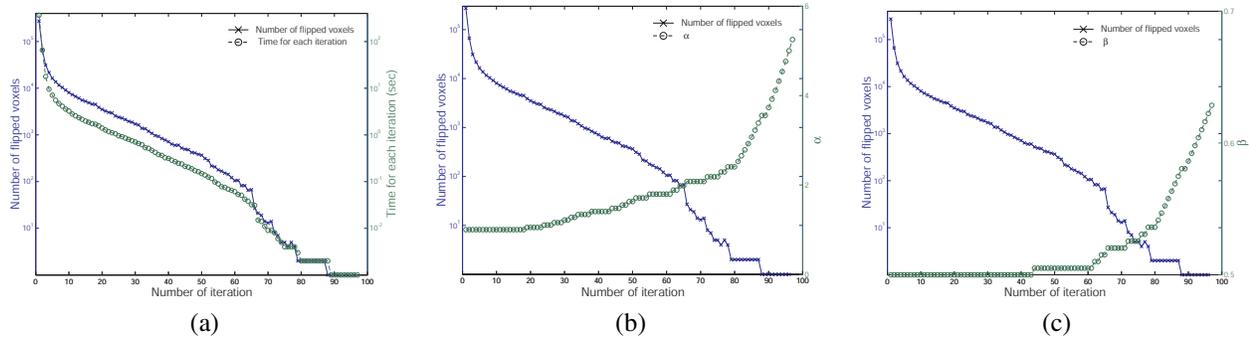


Figure 7. The relationship between the number of flipped voxels and the computational time, α and β . The scales of the number of voxels and computational time are logarithmic.

for the signs of neighbor voxels because it becomes unstable around the holes of range images, our new method takes a consensus of the sign distances by estimating the consistency of those of the adjacent voxels. Consequently, the final SDF becomes consistent and we efficiently complemented an incomplete model by hole filling using the consistent SDF.

Acknowledgments

This research was, in part, supported by JST under CREST Ikeuchi Project.

References

- [1] Brett Allen. <http://grail.cs.washington.edu/software-data/ply2vri/>, 2002.
- [2] J.C. Carr, R.K. Beaton, J.B. Cherrie, T.J. Mitchell, W.R. Fright, B.C. McCallum, and T.R. Evans. Reconstruction and representation of 3d objects with radial basis functions. In *Proc. SIGGRAPH 2001*, pages 67–76. ACM, 2001.
- [3] Y. Chen and G. Medioni. Description of complex objects from multiple range images using an inflating balloon model. *Computer Vision and Image Understanding: CVIU*, 61(3):325–334, 1995.
- [4] B. Curless and M. Levoy. A volumetric method for building complex models from range images. In *Proc. SIGGRAPH'96*, pages 303–312. ACM, 1996.
- [5] Cyra Technologies, Inc. CyraX 2500. <http://www.cyra.com>.
- [6] J. Davis, S.R. Marschner, M. Garr, and M. Levoy. Filling holes in complex surfaces using volumetric diffusion. In *Proc. First International Symposium on 3D Data Processing, Visualization, and Transmission*, 2002.
- [7] H. Delingette, M. Hebert, and K. Ikeuchi. Shape representation and image segmentation using deformable surfaces. *Image and vision computing*, 10(3):132–144, April 1992.
- [8] J.D. Foley, A. van Dam, S.K. Feiner, and J. F. Hughes. *Computer Graphics: Principles and Practice in C*. Addison Wesley Professional, 2nd edition, 1995. ISBN:0-201-84840-6.
- [9] H. Hoppe, T. DeRose, T. Duchamp, J.A. McDonald, and W. Stuetzle. Surface reconstruction from unorganized points. In *Proc. SIGGRAPH'92*, pages 71–78. ACM, 1992.
- [10] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1988.
- [11] W. Lorensen and H. Cline. Marching cubes: a high resolution 3d surface construction algorithm. In *Proc. SIGGRAPH'87*, pages 163–170. ACM, 1987.
- [12] MINOLTA Co. Ltd. Vivid 900 non-contact digitizer. <http://www.minoltausa.com/vivid/>.
- [13] R. Sagawa, K. Nishino, and K. Ikeuchi. Robust and adaptive integration of multiple range images with photometric attributes. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2001*, volume 2, pages 172–179, December 2001.
- [14] Ryusuke Sagawa. *Geometric and Photometric Merging for Large-Scale Objects*. PhD thesis, Graduate School of Engineering, The University of Tokyo, 2003.
- [15] I. Sato, Y. Sato, and K. Ikeuchi. Acquiring a radiance distribution to superimpose virtual objects onto a real scene. *IEEE Trans. on Visualization and Computer Graphics*, 5(1):1–12, 1999.
- [16] J.A. Sethian. *Level Set Methods*. Cambridge University Press, 1996.
- [17] M.D. Wheeler, Y. Sato, and K. Ikeuchi. Consensus surfaces for modeling 3d objects from multiple range images. In *Proc. International Conference on Computer Vision*, January 1998.
- [18] R.T Whitaker. A level-set approach to 3d reconstruction from range data. *International Journal of Computer Vision*, 29(3):203–231, October 1998.
- [19] H.-K. Zhao, S. Osher, and R. Fedkiw. Fast surface reconstruction using the level set method. In *Proc. First IEEE Workshop on Variational and Level Set Methods, in conjunction with Proc. ICCV '01*, pages 194–202. IEEE, 2001.

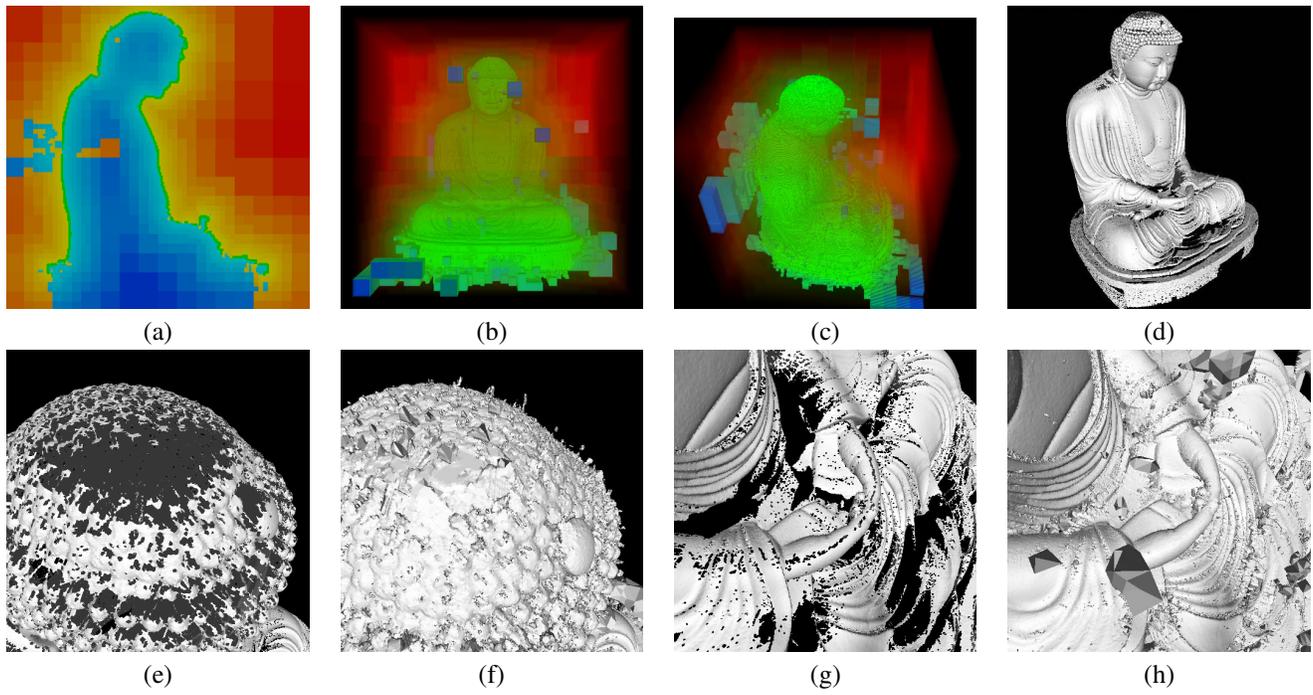


Figure 8. The SDF and mesh models of the Great Buddha of Kamakura before taking consistency of SDF

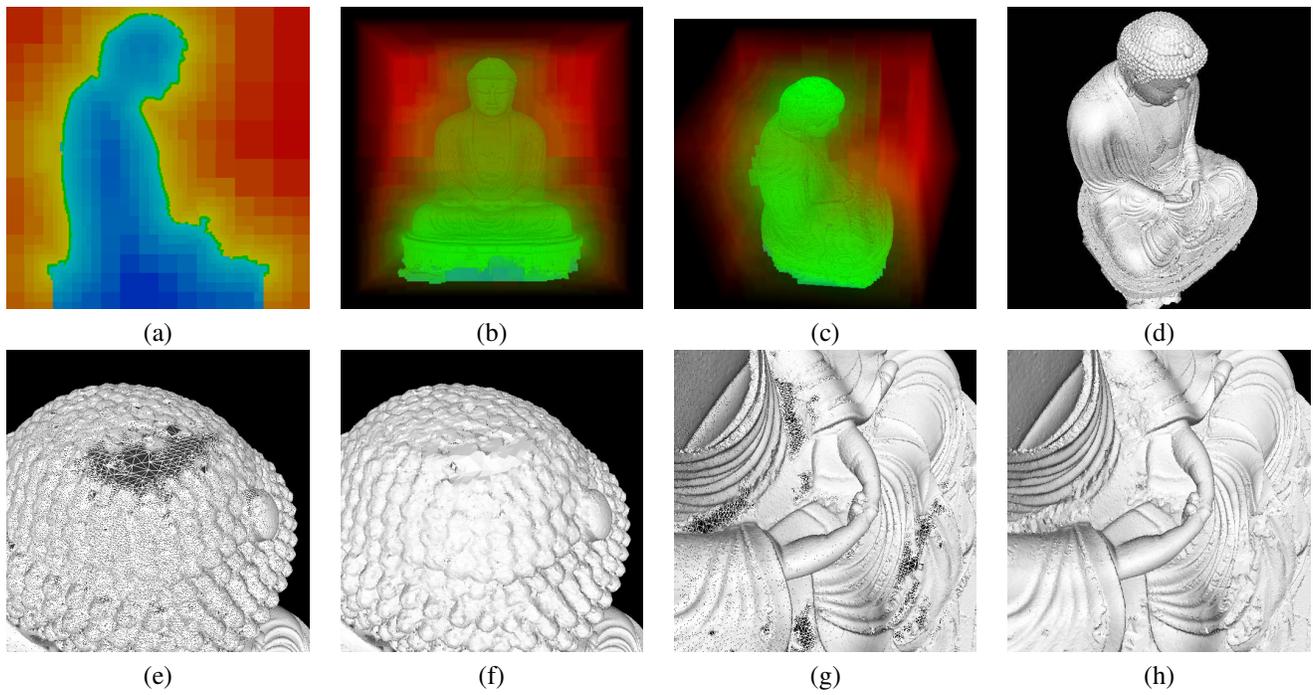


Figure 9. The models after taking consistency of the SDF of the Buddha