

# An Optimal Model Subdivision Method for 3D FDM Printers

Satoshi Nakajima <sup>†</sup>, Mikio Shinya (member)<sup>††,†††</sup>, Takashi Yuasa <sup>††</sup>,  
Michio Shiraishi <sup>††</sup>

## Abstract

Although FDM 3D printers have been getting popular, it is still difficult for novice users to determine basic parameters such as deposition directions and model subdivision. This paper describes an optimization method that determines the optimal model subdivision and deposition direction. The cost function is based on the supporting structure volume and the number of regions to adhere, which can be computed from multi-layered depth maps on GPU. The optimization can be efficiently achieved by the dynamic programming. Experiments showed that the method can improve object surface smoothness and reduce working time of support removal.

**Key words:** 3D printer, optimization, FDM, dynamic programming.

## 1. Introduction

A 3D printer is a device that creates 3D physical objects from geometric models. In recent years, low-cost consumer-level 3D printers have been released and rapidly spread out. Many attractive applications have been developed, such as animatronic mold<sup>(1)2)</sup> and Kinect-base self-portrait printing<sup>(3)</sup>. This results in more opportunities for non-professional novice users to create 3D objects by 3D printers, and it is important to develop tools that assist untrained users to easily create high quality objects.

Zhou developed a structural analysis method that identifies structural mechanical problems<sup>(4)</sup>. Reduction of used materials was discussed by Wang<sup>(5)</sup>, who proposed to construct inner objects by truss scaffoldings and to apply nonlinear optimization. Prévost et. al., proposed a method balancing static models at rest<sup>(6)</sup>. However, many issues still remain open.

Most of low-cost systems adopt the fused deposition modeling (FDM) method, which makes physical objects by depositing melted materials such as ABS resins. FDM 3D printers build up 3D objects layer by layer. Therefore, the deposition direction has a large influence

on the production. When FDM 3D printers make objects containing holes or bridge-structures, they have to deposit supporting structures underneath the objects, as shown in **Fig. 1-a**. It is necessary to remove these supports by hand after deposition, which often damages the objects, as shown in Fig. 1-b. Therefore, it is very important to control support structures to obtain well-finished objects and to reduce work time.

The support volume can be reduced by changing the size or structure of supports<sup>(7)8)</sup>. However, it is possible to reduce support volumes by selecting an appropriate deposition direction, as well. We can also reduce it by dividing a geometric model into several parts, creating each part separately, and adhering them after the deposition, as shown in **Fig. 2**. For novice users, however, it is difficult to determine the appropriate direction and model subdivision. Therefore, it is important to develop a computation model that calculates optimal deposition directions and model subdivision.

Although a model subdivision technique based on concave edges has been reported<sup>(9)</sup>, it is restricted to simple geometric objects consisting of polygons parallel to each other and it is difficult to apply the method to complicated shapes such as figure dolls. To the authors' knowledge, optimization methods to determine model subdivision based on the cost function has not been discussed.

This paper describes an optimization method that determines the optimal model subdivision and deposition direction by utilizing CG techniques and the dynamic programming. The cost function we used in this paper

This paper reported in IWAIT2014, January 6 2014.

Received Month xx, 20xx; Revised Month xx, 20xx; Final received Month xx, 20xx; Accepted Month xx, 20xx

<sup>†</sup> Fujitsu Broad Solution & Consulting, Inc.

(Tokyo, Japan)

<sup>††</sup> Graduate School of Science, Toho University

(Chiba, Japan)

<sup>†††</sup> UEI Research

(Tokyo, Japan)

was defined by a sum of supporting structure volumes and the number of regions to adhere, although it can be easily introduce other factors such as the area of the regions and weighting functions of support structure length.

Supporting structure volumes can be efficiently computed from multi-layered depth maps on GPU by using the depth peeling method<sup>10)</sup>. The number of adhesion regions is also calculated from the layered depth maps using a filling algorithm. The optimization is also efficiently achieved by the dynamic programming. We restricted model subdivision planes to be perpendicular to the deposit direction, which allows us to apply the dynamic programming. By using the optimal subdivided object, even novice users can reduce the problems caused by the support removal.

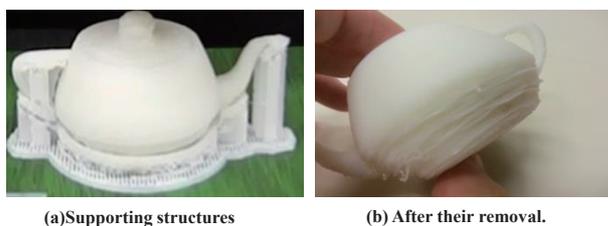


Fig. 1 An example of supporting structures.

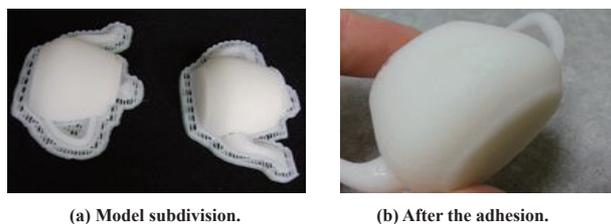


Fig. 2 Model subdivision and adhesion.

## 2. 3D Printer

### 2.1 3D printing systems

A 3D printer is a device creating 3D physical objects from geometric models. There are several types of 3D printing systems, such as the stereo-lithography, the selective laser sintering (SLS) and the fused deposition modeling (FDM). Stereo-lithography 3D printers scan a high power laser beam and draw the cross-sections of 3D physical objects onto the surface of the pool of light curing resin, repeating it layer by layer. SLS 3D printers create 3D physical object by fusing small particles such as nylon using a high power laser. FDM 3D printers create objects by depositing thermoplas-

tic resin such as ABS resins or polycarbonates, by depositing the melted materials layer by layer. The stand on which objects are placed horizontally moves to construct object cross-sections like a plotter and vertically steps down after the completion of a layer. The FDM system has enabled low-cost products and is most popular in the consumer-level market.

### 2.2 Supporting Structures

When FDM 3D printers create objects with holes or bridge-structures, they have to deposit supporting structures underneath the objects (Fig. 1-a). After the deposition, supporting structures have to be removed by hand, which often damages the objects. Therefore, it is very important to control support structures to obtain well-finished objects and to reduce work time.

### 2.3 Cost function

To determine appropriate directions and model subdivision, we have to consider the following factors.

- (1) Object qualities: The object is often damaged by removing supporting structures. It can be expected that less support structures decrease the damage.
- (2) Support removal time: Support removal by hand is a labor intensive task. This factor also depends on the amount of supporting structures.
- (3) Adhering time: It is necessary to adhere divided parts, which also requires manual operations. This increases with respect to the number of subdivision planes.

The volume of supporting structures and the adhering region can be considered to have dominant influences on the factors (1), (2) and (3). Thus, we adopted a cost function consisting of two terms: the volume of supporting structures and the number of regions to adhere after the deposition. Note that the volume term monotonously decreases with the number of subdivision while the number of regions to adhere increases.

The area of adhering regions can be another important factor. It will be also shown that this factor can be also integrated to our cost function. Design issues on the cost function will be discussed in Section 5.4.

## 3. Computation of Supporting Structure Volumes

A supporting structure volume can be regarded as a volume of shadow regions when light rays come in the deposit direction and can be calculated by using depth values as **Fig. 3**. Let us assume model subdivision planes to be perpendicular to the deposit direction

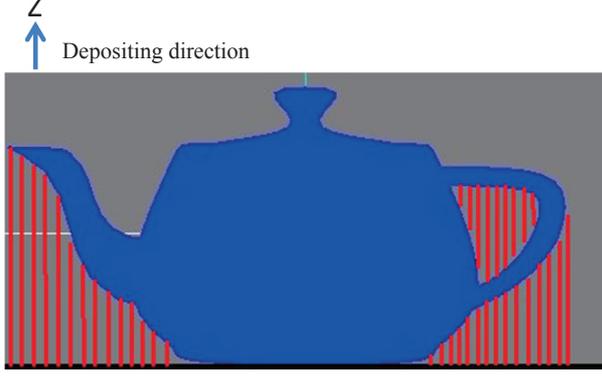


Fig. 3 Supporting structures.

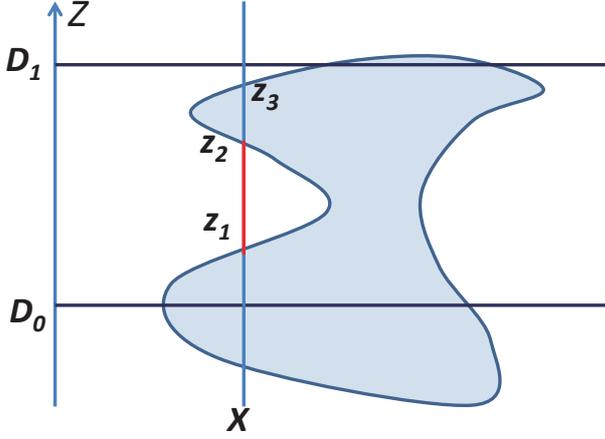


Fig. 4 Length of supporting structure at  $X$ .

$z$ . In **Fig. 4**, values  $z_i (i = 1, \dots, n)$  indicate the depth value of the object surface at  $X$ , and  $D_1$  and  $D_0$  are the depth values of the upper subdivision plane and the lower plane:

$$D_0 < z_1 < z_{l+1} < \dots < z_h < D_1.$$

In this example, we have to set a support between  $z_1$  and  $z_2$  at  $X$  and its length is  $z_2 - z_1$ . By summing up these support length, the total support length at  $X$ ,  $h_s(X)$ , can be calculated by:

$$h_s(X) = \sum_{\text{outside object}} L_i, \quad (1)$$

$$L_0 = z_l - D_0, \quad (2)$$

$$L_i = z_{l+i+1} - z_{l+i}, \quad (3)$$

$$L_h = 0. \quad (4)$$

Note that  $L_h$  is zero because no support is necessary above the top surface. The support volume in this subdivision section can be calculated by the sum of  $h_s(X)$ :

$$H'_s(D_0, D_1) = \sum_X h_s(X). \quad (5)$$

Depth values  $z_i(X)$  can be stored in multi-layered depth

maps. Referring to these maps, the supporting structure volume can be efficiently calculated.

In 3D printing, it is also possible to deposit objects in the reverse direction as well. Therefore, we set the supporting structures volume  $H_s$  is the smaller volumes in the direction  $s$  and  $-s$ , as:

$$H_s(D_0, D_1) = \min(H'_s, H'_{-s}). \quad (6)$$

The area of adhering regions,  $H_a$ , can be also calculated in a similar way. By setting  $L_i = 1$ , the area  $H_a$  is given by counting the number of points as:

$$H_a = \sum_X \sum_{\text{outside object}} 1 \quad (7)$$

The number of regions to adhere,  $M(z)$ , can be calculated from the cross-sections of the objects and subdivision planes. A bit map image of the cross-section  $C_D(x, y)$  can be constructed from the multi-layered depth map. The number of regions is then counted through simple region filling.

#### 4. Optimization Method

The cost function  $g_s$  is defined by the summation of the supporting structure volume  $H_s(Z_i, Z_{i+1})$  and the number of regions to adhere  $M(Z_i)$  as:

$$g_s(Z_i, Z_{i+1}) = H_s(Z_i, Z_{i+1}) + \alpha M(Z_i), \quad (8)$$

where  $\alpha$  denotes a weight constant balancing the two terms.

Selecting the optimal  $\alpha$  value is generally a difficult task. However, it can be roughly estimated by regarding it as a balancing factor between the dynamic ranges of a decrease in the support volume  $H_s$  and the number of regions to adhere  $N$ . Given the expected decrease in the volume  $\delta H$  at the maximum allowable number of the planes  $N_m$ ,  $\alpha$  can be estimated by  $\delta H / N_m$ . In the case of **Fig. 9**,  $\delta H$  is about 0.15. If we set  $N_m = 15$ ,  $\alpha = 0.15 / 15 = 0.01$  would be a good value.

##### 4.1 Optimization of subdivision planes

Assume that an object is subdivided by  $N$  planes perpendicular to the deposition direction  $s$ , as shown in **Fig. 5**. Let  $Z_i (i = 1, \dots, N)$  be their depth values, and the minimum and maximum depth value of the object be  $Z_o$  and  $Z_e = Z_{N+1}$ . We quantize  $z$ -value in  $n_z$  levels. The total cost function,  $G$ , is defined by the sum of the cost values between neighboring subdivision planes,  $g_s(Z_i, Z_{i+1})$ , as:

$$G = \sum_{i=0}^N g_s(Z_i, Z_{i+1}). \quad (9)$$

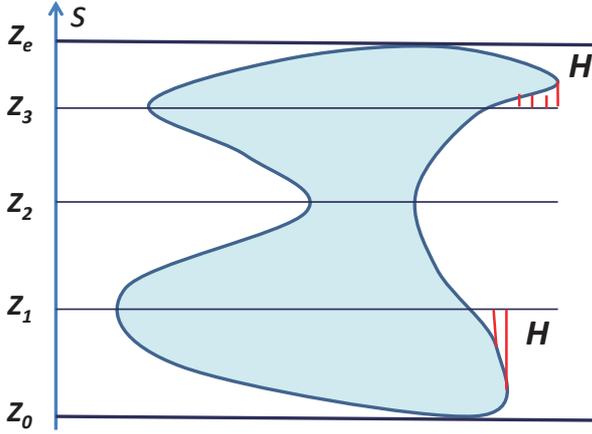


Fig. 5 Supporting structure volume.

The minimization of  $G$  can be efficiently achieved by the dynamic programming at an  $O(n_z^2)$  computation cost, while the cost for the exhaustive search is  $O(n_z^N)$ . First, we calculate  $g_s(z, z')$  and store the values into a table. Next, we calculate  $E_j(z)$  defined by

$$\begin{aligned} E_0(Z_1) &= g_s(Z_0, Z_1), \\ E_j(Z_j) &= \min_{Z_{j-1} < Z_j} (E_{j-1}(Z_{j-1}) + g_s(Z_{j-1}, Z_j)). \end{aligned} \quad (10)$$

and also stored as DP-tables. The minimum values  $G_{min}$  for given  $N$  and  $S$  can be found by a linear search, as:

$$\begin{aligned} G_{min}(N; s) &= \min_{Z_1, \dots, Z_N} G \\ &= \min_{Z_N} (E_{N-1}(Z_N) + g_s(Z_N, Z_e)). \end{aligned} \quad (11)$$

#### 4.2 Optimization of number of subdivision

The optimization represented by Eq. (11) is conducted for each number of subdivision plane  $N = 1, 2, \dots, N_{max}$ . We adopt  $N$  that provides the smallest cost value as the optimal number of planes.

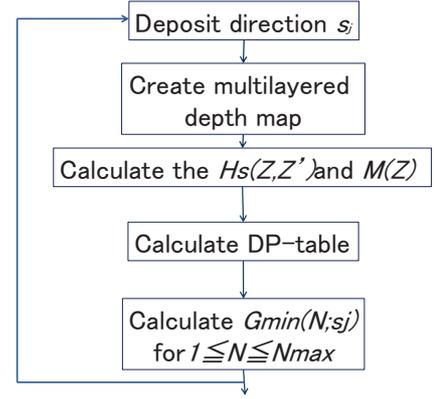
#### 4.3 Optimizing of deposit direction

The cost function Eq. (9) is non-linear with respect to the deposit direction. We decided to simply sample deposition directions and calculate the minimal cost in each direction. We adopt the direction that provides the smallest cost value as the optimal deposit direction.

### 5. Experiments

#### 5.1 Procedure and implementation

Fig. 6 outlines the proposed method. We sample deposit direction  $s_j$  and calculate multi-layered depth maps  $z_i[x][y]$  by depth peeling method<sup>10)</sup>. By referring to the maps, we calculated support volumes  $H_s(Z, Z')$ .



Optimized deposit direction and subdivision planes

Fig. 6 Outline of the proposed method

We also construct a bitmap image of cross-sections  $C_z(x, y)$  and count the number of adhesion,  $M(Z)$ .

Using  $H_s(Z, Z')$  and  $M(Z)$ , the DP-table  $E_k$  is calculated by Eq. (11). Minimal cost values  $G_{min}(N; s_j)$  are then calculated from Eq. (11) for each number of subdivision planes  $N$ . We adopt  $N$  that provides the smallest cost value. We execute this process for all sample directions and the optimal deposition direction is the sample direction in which the cost value is smallest.

The object is clipped by the optimal subdivision planes and the clipped polygons are triangulated. We used Delaunay triangulation and implemented it using CGAL<sup>11)</sup>.

In the experiments, we used multi-layered depth maps at  $64 \times 64$  resolution. We quantized depth values by 64 levels. We sampled 19 deposition directions as shown in Fig. 7.

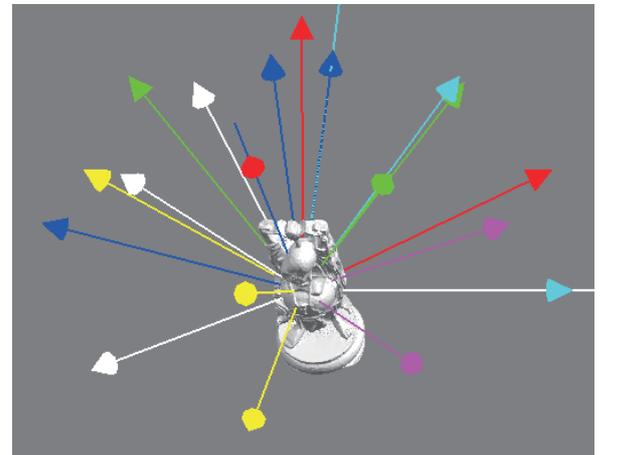


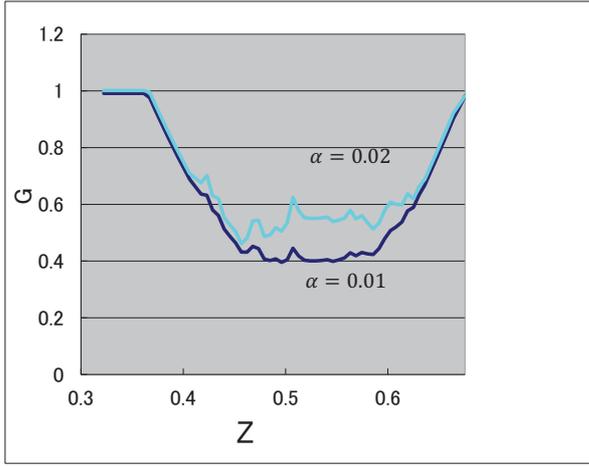
Fig. 7 Sampling direction.

#### 5.2 Optimization

We applied the proposed method to the Stanford

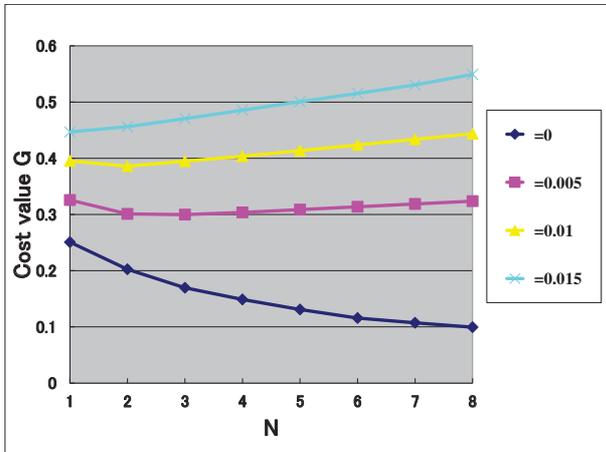
Buddha model<sup>12)</sup> and made physical objects. In this experiment, we fixed the deposition direction and the number of subdivision  $N = 1$  to show the basic properties.

**Fig. 8** shows the cost value  $G$  when we changed the weighting factor  $\alpha$  in Eq. (8). The supporting structure volume is normalized by the volume of the object. As shown in the figure,  $G$  has the minimum value near the center of the object.



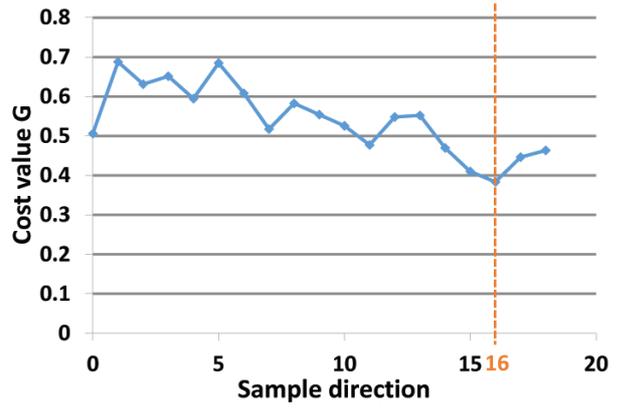
**Fig. 8** Support structure volume, the number of regions to adhere and the number of subdivision.

Fig. 9 shows the minimal cost value  $G_{min}(N)$  with respect to the number of subdivision  $N$ . When the weighting factor  $\alpha$  is zero, the cost value monotonously decreases with the number of subdivision. As the weighting factor  $\alpha$  increases, the minimum cost value is obtained at lower  $N$  value.

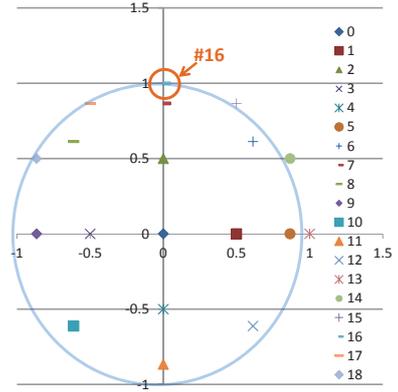


**Fig. 9** The number of subdivision plane and cost value.

**Fig. 10-a** shows  $G$  values with respect to each sam-  
Paper



(a) Cost value and deposition direction.



(b) Sampled deposition direction.

**Fig. 10** Deposition direction and cost value.

pled deposit directions. In this experiment,  $G$  took the minimum value at direction #16. The optimal number of subdivision planes was 2. **Fig. 11** shows the subdivided objects. The processing time for 19 directions was 9.2 second on a Windows PC (Intel Core i7-3770K @3.50GHz, GeForce GTX 690). **Table 1** shows the measured processing time for one direction. It took a few hours to print the objects and we think our optimization process is fast enough.

Calculation of depth map $hs(z)$	Calculation of $Hs(Z, Z')$ and $M(Z)$	Calculation of DP-table
0.18 s	0.27 s	<0.001 s

### 5.3 Examples

We created physical objects from the Stanford Buddha, Dragon, and Armadillo models<sup>12)</sup> using the proposed optimization method. The created objects are shown in **Fig. 12**, **Fig. 13** and **Fig. 14**. In the experiments, we used an FDM 3D printer, named “UP! Plus 3D Printer”<sup>13)</sup>. We set the parameter  $\alpha = 0.01$  in this experiment. As shown in Fig. 12-a, Fig. 13-a and Fig. 14-a, when objects were not subdivided, support-

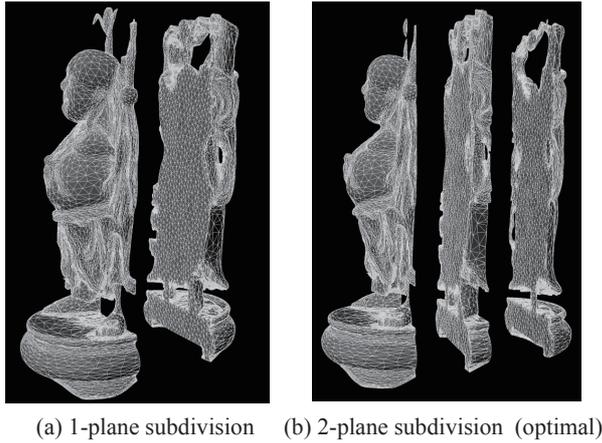


Fig. 11 Divided 3D CG object.

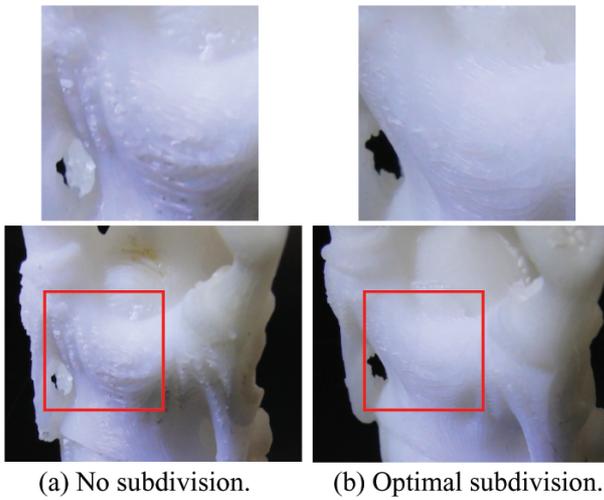


Fig. 12 Experimental results of the Stanford Buddha.

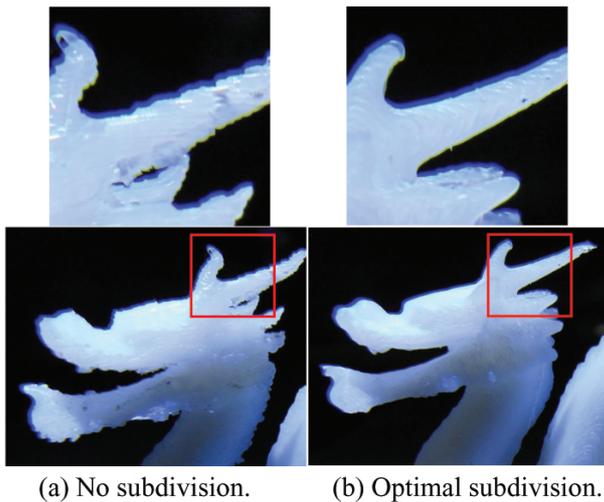


Fig. 13 Experimental results of the Stanford Dragon.

ing structures could not be completely removed and the object surface remained rough. In Fig. 12-b, Fig. 13-b and Fig. 14-b, on the other hand, the optimization resulted in smoother object surfaces and improved quality of object finish.

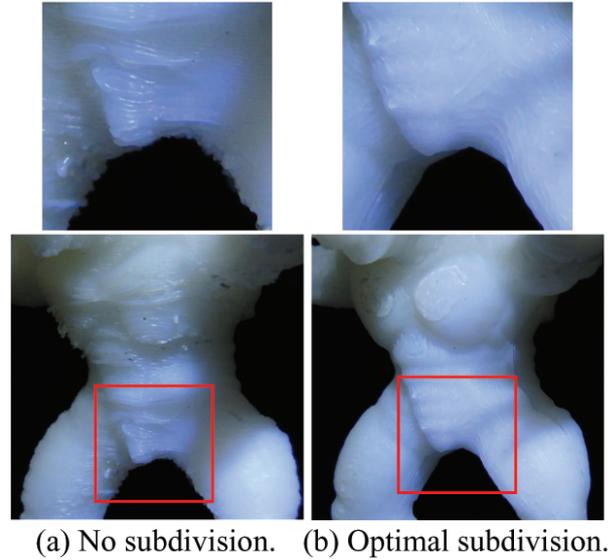


Fig. 14 Experimental results of the Stanford Armadillo.

Table 2 SUPPORT REMOVE TIME AND ADHERING TIME.

	Number of subdivision	Support remove (min)	Adhering (min)	Print time (min)
Buddha	0	19	0	210
	2(optimal)	9	5	215
Dragon	0	19	0	70
	1(optimal)	1	1	72
Armadillo	0	3	0	127
	3(optimal)	2	13	111

Table 2 shows printing time and working time required in removing supports and adhering subdivided parts. The table indicates that the optimization largely decreased support removing time. Subdivision introduces adhering time, which was rather modest with the Dragon and the Buddha. However, the adhering time with the Armadillo was relatively large because the subdivided model contained many small regions to adhere. It was suggested to refine the cost function to avoid such small subdivided area. The differences in printing time were not significant.

#### 5.4 Discussion

Through the experiments, we noticed that the cost function defined by Eq. (8) may not necessarily be the best measure of quality and working time but that the area of adhering regions and the length of support structures can be important factors as well. From these considerations, we have examined some other cost functions based on region areas, support length, and their combination. Although we have made several pilot experiments using different cost functions and various balancing values ( $\alpha$ ), we cannot obtain a significant im-

provement in terms of working time and quality of finish so far. We realized that more systematic investigations are necessary to refine the cost function.

The proposed methods digitizes the depth of dividing planes and the deposit directions in a fixed number of levels. With respect to the direction, the cost function cannot be represented by an analytic function in general and it is hard to handle it in a continuous space. However, it is possible to apply a coarse-to-fine subdivision approach to obtain a good trade-off between the resolution and cost. We can further subdivide the positions/directions near the minimum values at a higher sampling rate so that we can search better values in a depth first fashion.

The proposed method adopts the depth peeling method and dynamic programming. To make comparisons with more naive methods, we first implemented a scan-conversion program that creates sorted lists of depth values from which a multi-layered depth image is made. Surprisingly, the execution time to build a multi-layered depth image of the Buddha model at  $64 \times 64$  resolution was 0.031 sec, much smaller than that for the depth peeling method, 0.18 sec. However, we would still prefer the depth peeling method to the software scan-conversion because of simplicity of implementation. We also implemented an optimization code based on an exhaustive search. The execution time rapidly grows up with respect to the maximum number of dividing planes  $N$  and was 13 seconds and 119 seconds for  $N = 7, 8$ , respectively.

## 6. Conclusion

Although FDM 3D printers have been getting popular, it is still difficult for novice users to determine basic parameters such as deposition directions and model subdivision. This paper describes an optimization method that determines the optimal model subdivision and deposition direction by utilizing CG techniques and the dynamic programming. The cost function we used is defined by a sum of supporting structure volumes and the number of regions to adhere.

Supporting structure volumes can be efficiently computed from multi-layered depth maps on GPU. The optimization can be efficiently achieved by the dynamic programming. Using the optimization method, we created physical objects with an FDM 3D printer. The optimization resulted in smoother object surfaces and improved quality of object finish. It also reduced working time of support removal. The experiments also sug-

gested that more refinement of the cost function is necessary to reduce adhering time more effectively. Future work includes systematic studies on the cost functions.

## References

- 1) ZHU, L., XU, W., SNYDER, J., LIU, Y., WANG, G., AND GUO, B. 2012. Motion-guided mechanical toy modeling. *ACM Trans. Graph.* 31, 6, 127.
- 2) BICKEL, B., BACHER, M., OTADUY, M. A., LEE, H. R., PFISTER, H., GROSS, M., AND MATUSIK, W. 2010. Design and fabrication of materials with desired deformation behavior. *ACM Trans. Graph.* 29, 4.
- 3) LI, H., VOUGA, E., GUDYM, A., LUO, L., BARRON, J. T., AND GUSEV, G. 2013. 3D self-portraits. *ACM Trans. Graph.* 32, 6 (November), 187.
- 4) Q. Zhou, J. Panetta, D. Zorin, Worst-case structural analysis, *ACM Trans. on Graphics*, vol. 32, No.4, 137, 2013.
- 5) WANG, W., WANG, T. Y., YANG, Z., LIU, L., TONG, X., TONG, W., DENG, J., CHEN, F., AND LIU, X. 2013. Cost-effective printing of 3D objects with skin-frame structures. *ACM Trans. Graph.* 32, 6, 177:1-177:10.
- 6) Prévost, R., Whiting, E., Lefebvre, S., and Sorkinehornung, O. 2013. Make it stand: balancing shapes for 3D fabrication. *ACM Trans. Graph.* 32, 4 (July), 81:1-81:10.
- 7) HUANG, X., YE, C., MO, J., AND LIU, H. 2009. Slice data based support generation algorithm for fused deposition modeling. *Tsinghua Science and Technology* 14, S1, 223-228.
- 8) HEIDE, E., 2011. Method for generating and building support structures with deposition-based digital manufacturing systems, 07. US Patent 20110178621 A1.
- 9) Z. Hu, K. Lee Concave edge-based decomposition for hybrid rapid prototyping, *International Journal of Machine Tools and Manufacture*, vol. 45, No. 1, pp. 35-42,1, January 2005.
- 10) C. Everitt, Interactive order-independent transparency, Technical report, NVIDIA Corporation, 2001.
- 11) CGAL, Computational Geometry Algorithms Library, <http://www.cgal.org>
- 12) The Stanford 3D Scanning Repository, <http://graphics.stanford.edu/data/3Dscanrep/>
- 13) UP! Plus 3D Printer, <http://3dprintingsystems.com/products/up-plus-3d-printer/>



**Satoshi Nakajima** received a B.S. and an M.S degree from Toho University in 2012 and 2014. He currently works with Fujitsu Broad Solution & Consulting, Inc.



**Mikio Shinya** is currently a Professor in the Department of Information Science, Toho University. He received a BSc in 1979, a MS in 1981, and a PhD in 1990 from Waseda University. He joined NTT Laboratories in 1981, and moved to Toho University in 2001. He was a visiting scientist at the University of Toronto in 1988-1989. His research interests include computer graphics and visual science.



**Takashi Yuasa** received a B.S. degree from Toho University in 2014. He is a master student at Graduate School of Science, Toho University.



**Michio Shiraishi** received the B.A., M.A., and Ph.D. degrees from The University of Tokyo in 1997, 1999, 2003, respectively. He is an associate professor in the Department of Information Science, Toho University. His current research interests are non-photorealistic rendering and visualization.