No-infill 3D Printing

X.R. Wei [1] and G.H. Geng[1] and Y.H. Zhang[1]

[1] Schoole of Information Science and Technology, Northwest University, China

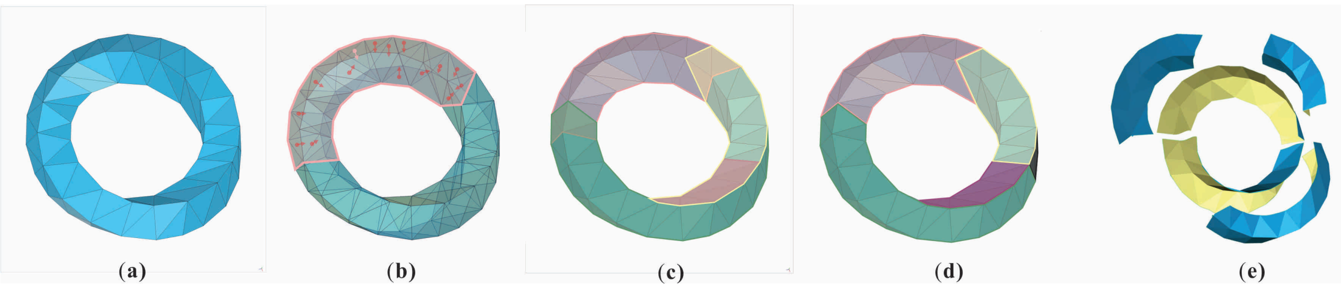

Figure 1: *Overview of the proposed algorithm.*

Abstract

This paper introduces a partition method for printing hollow objects without infill via Fused Deposition Modeling (FDM), which is one of the most widely used 3D-printing technology. We linked the partition problem to the exact cover problem (ECP). Then the hollow objects can be printed without any infill. The experimental results show that our solution can be applied to a variety of models, closed-hollowed or semi-closed, with or without holes, as evidenced by experiments and performance evaluation on our proposed algorithm.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: —

1. Purpose

The overview of the proposed algorithm is shown in Figure 1, which solves the problem of printing hollow objects. Most of hollow objects (e.g., hollow ball(as shown in Figure 2), vase, cupboard shapes) do not possess infill, which can not be printd since FDM has the intrinsical limitation that a hollow object cannot be printed directly without infill supporting it cause the filament can only be deposited on the top of an existing surface.

Although adding internal support can help solve the hollow model printing problem, it can not be removed and is decidedly not ideal as far as consumption of resources and time. Furthermore, in some applications, the internal support is detrimental since it impacts the accuracy of the inner surface once removed. For example, a test tube, which requires precise inner capacity.

2. Aim

- No Infill: The inside surface can be printed without support.
- Minimum: There are as few partition parts as possible.

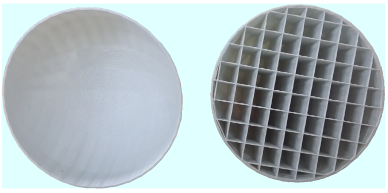

Figure 2: *A hollow ball that printed with infill.*

- Parts Uniform: The size of the partitions is uniform.

3. Method

The input to our algorithm is a 3D mesh model as shown in Figure 1(a), the output is an enumeration of model parts after segmentation as shown in Figure 1(e). Each step of the algorithm is described in details:

Step1: Candidate No-infill Subset (NIS). For each mesh, the in-

DOI: 10.2312/egp.20161057

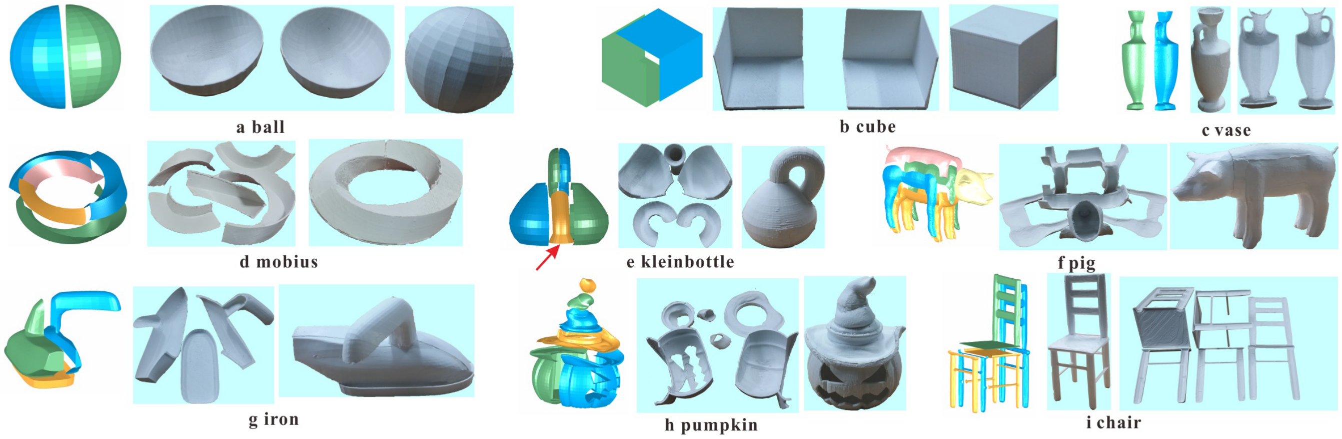

Figure 5: *Photographs of some physically printed and fabricated 3D models.*

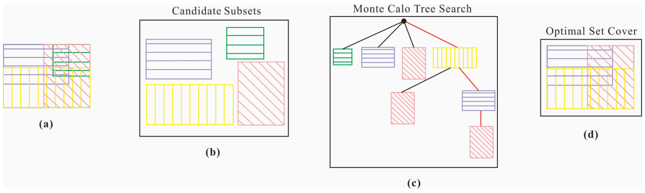

Figure 3: *Overview of the optimal set cover.*

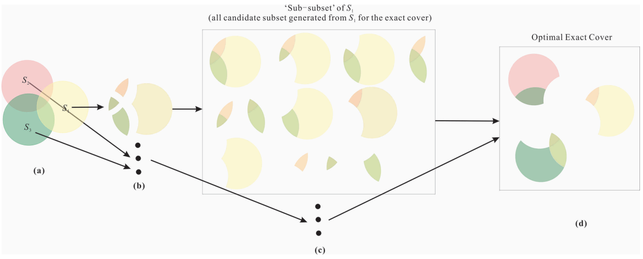

Figure 4: *Overview of the optimal exact cover.*

side surface can be printed without support in myriad oriented positions which constitute an oriented interval called the no-support oriented interval (NSOI). If there is a non-empty intersection among the NSOI of any adjacent meshes, these meshes can constitute a NIS. We used the region growing method to generate sufficiently large NISs. The red region in Figure 1(b) is a NIS region of the model.

Table 1: *Statistics for our algorithm's results.*

model	#*s* (mm^3)	#*t*(m)	#*m*(g)	#*pt*	#*pc*	#*c*(s)
ball	65*65*65	62	14	31%	42%	7.5
cube	77*65*75	190	30.2	34%	54%	1.6
iron	112*78*68	120	25.6	27%	44%	18.7
mobius	120*128*28	168	45.2	18%	14%	29.3
klein	94*85*105	220	45.9	30%	35%	20.2
pig	183*66*104	323	79.2	30%	33%	37.9
vase	35*100*35	74	16.9	14%	20%	15.4
pumpk	120*122*140	743	162.3	22%	35%	37
chair	200*80*76	330	77.6	5%	6%	23.8

Step2: Set Cover. We applied the Monte Carlo tree search algorithm, which searches the optimal set-cover by extensive simulation, to the set of candidate NIS, as shown in Figure 3. We also employed a function for evaluating the quality of subsets in the set cover, constructing the upper confidence found policy according to the function. We used the upper confidence bound for trees algorithm to search the Monte Carlo tree to ultimately find the optimal set cover. Figure 1(c) is the optimal set cover of the model.

Step3: Exact Cover. There are many intersections among the subsets in the optimal set cover, as shown in Figure 4. We first obtained the candidate subsets of the exact cover by analyzing the subsets and their intersections in the optimal set cover, then searched the optimal exact cover from these candidate sets using the depth-first tree search method with boundary limits. The partition parts were then obtained from the exact cover. Figure 1(d) is the optimal exact cover of the model.

4. Results and Conclusion

Figure 5 shows a gallery of partition results produced by our algorithm for a various shapes that were closed or semi-closed, smooth or coarse surfaces, with or without holes, symmetrical or asymmetrical, all of which demonstrate the generality of our approach.

Table 1 provides the Statistics for experiments in Figure 5. We report the number of partition part(#*y*), print size(#*s*), print time(#*t*), material cost(#*m*), the percentage reduction of the print time(#*pt*), the percentage reduction of the material cost(#*pc*) and the Execute time(#*c*). Experiments show that our algorithm can effectively reduce the print time and material cost. The reduction is most obvious in the models with a large hollow volume, such as the cube, the ball, the iron and the Klein bottle.