

DIRECT GEOMETRY PROCESSING FOR TELE-FABRICATION

Yong Chen

yongchen@usc.edu
Industrial and Systems Engineering
University of Southern California
Los Angeles, CA 90089

Kang Li, Xiaoping Qian

kli@iit.edu, qian@iit.edu
Mechanical, Materials and Aerospace Engineering
Illinois Institute of Technology
Chicago, IL60616

ABSTRACT

This paper presents a new approach for tele-fabrication where a physical object is scanned in one location and fabricated in another location. This approach integrates three-dimensional (3D) scanning, geometric processing of scanned data, and additive manufacturing technologies. In this paper, we focus on a set of direct geometric processing techniques that enable the tele-fabrication. In this approach, 3D scan data is directly sliced into layer-wise contours. Sacrificial supports are generated directly from the contours and digital mask images of the objects and the supports for Stereolithography Apparatus (SLA) processes are then automatically generated. The salient feature of this approach is that it does not involve any intermediate geometric models such as STL, polygons or non-uniform rational B-splines that are otherwise commonly used in prevalent approaches. The experimental results on a set of objects fabricated on several SLA machines confirm the effectiveness of the approach in faithfully tele-fabricating physical objects.

KEYWORDS

Additive manufacturing, 3D scanning, geometry processing, process planning, tele-fabrication

1 INTRODUCTION

This paper presents an end-to-end tele-fabrication approach where a three-dimensional (3D) physical object is scanned in one location and reproduced in another location. Due to the pervasive deployment of sensing and computing technologies, tele-fabrication has become increasingly feasible and important in a globally distributed supply-chain environment for product development.

Tele-fabrication can significantly reduce the product development cycle by enabling long-distance collaborations and maximizing the productivity of equipment resources [1]. The geometric information transfer of the input object among different locations is crucial in making decisions and determining parameters for the process planning tasks. Several geometric representations can be used to represent input objects

for fabrication purposes, such as STL, B-Rep, CSG and point data. Among these data types, STL is extensively and commonly used and has also seen quite a few tele-fabrication applications [1][2][3][4][5]. However, due to the inherent disadvantages of the STL format [6], such as redundant storage and approximation error, the accuracy and efficiency of the processing planning (e.g. slicing) become subject to the representation defects of STL format.

A key technical challenge in tele-fabrication is how to transfer from one location to another the geometric data of the 3D parts that is required for fabricating them in an efficient, accurate and flexible manner. In this paper, we propose a direct processing approach for tele-fabrication where the geometric information flows efficiently among different locations without sacrificing accuracy. More specifically, this fabrication process takes the digitized point form of physical objects in one location as input, directly slices the point data and transfers the obtained slices to another location, where sacrificial structures are built from the slices and the final part is built by means of Additive Manufacturing (AM).

The salient feature of our approach is that a set of geometric processing procedures have been developed that enable the direct generation of such geometric data for additive manufacturing, without using any intermediate surface models such as non-uniform rational B-spline (NURBS) surfaces, polygons or STL files. This approach thus bypasses many laborious procedures such as manual data segmentation, surface fitting and re-discretization into STL files, which are otherwise commonly associated with reverse engineering physical objects via an additive manufacturing process. Our approach builds on a set of geometric processing techniques that are developed at two institutions: *Illinois Institute of Technology* (IIT) and *University of Southern California* (USC). More specifically, this approach includes a new slicing approach that can automatically generate sliced contours from scanned raw data without the common NURBS surfaces and polygons and a new support generation method that can automatically generate the sacrificial structures for supporting the overhung structures in the layer-wise fabrication process. The slicing approach generates sliced contours are: 1) geometrically accurate, due to the use of the Moving Least-Square (MLS) point-set representation; 2) topologically robust, due to the use of Morse

complex for non-degenerate cases and the MLS-based Marching Cube contouring algorithm for complicated degenerated cases. The support generation method is developed by analyzing the layer-wise fabrication process in a systematic manner and using a set of point-based offset and Boolean operations to automatically generate the support structures.

Figure 1 gives a schematic illustration of the developed approach for tele-fabrication where a part is scanned in Chicago at IIT and fabricated in Los Angeles at USC. Geometric data that are adaptive to a specific additive manufacturing machine can then be automatically generated. Figure 2 gives an illustration of such geometric data flow for one machine that

can re-produce the part in live size. The scanned data is sliced into contours at IIT via the direct slicing procedure which extracts layered contours from the scan data via an implicitly defined moving-least squares (MLS) surface. The sliced shape model is then transferred to USC where a support structure is then automatically generated from the sliced shape model. Process plans for layer-wise fabrication are prepared based on the sliced model and the resulting support structures. A physical part is subsequently fabricated. Reproduction of the parts by different machines with different materials and scaling are shown in the later sections of this paper.

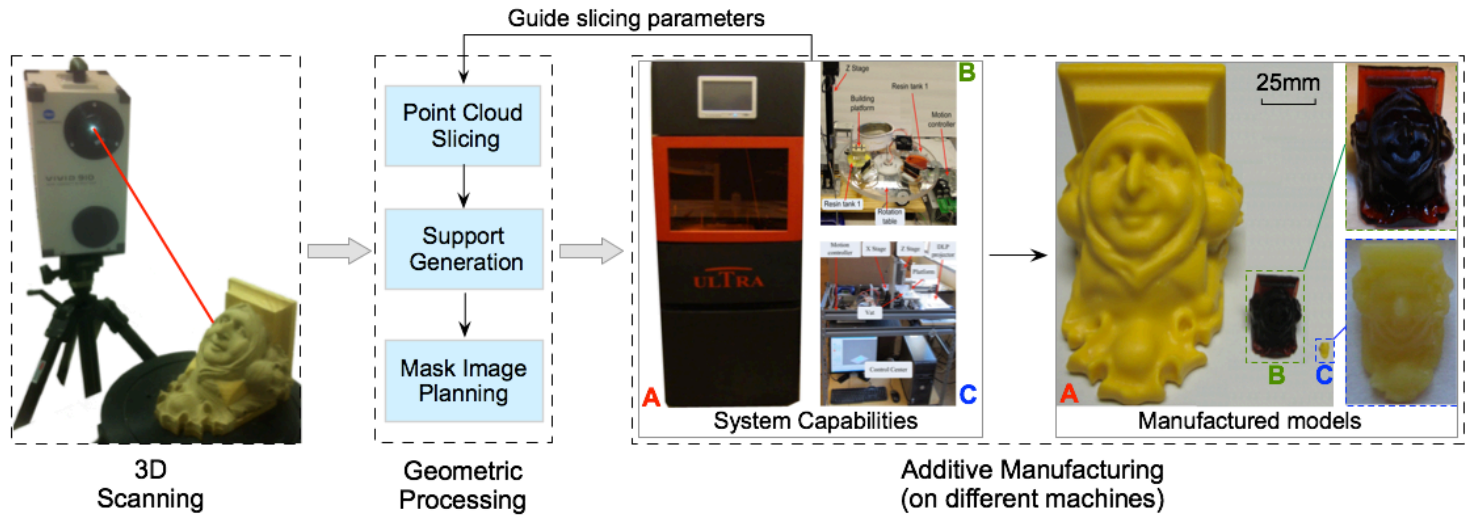


Figure 1. Overview of tele-fabrication: a physical object is scanned in one location and fabricated in another location. The scan data is processed according to the given machine specifications from micro- to meso-scales.

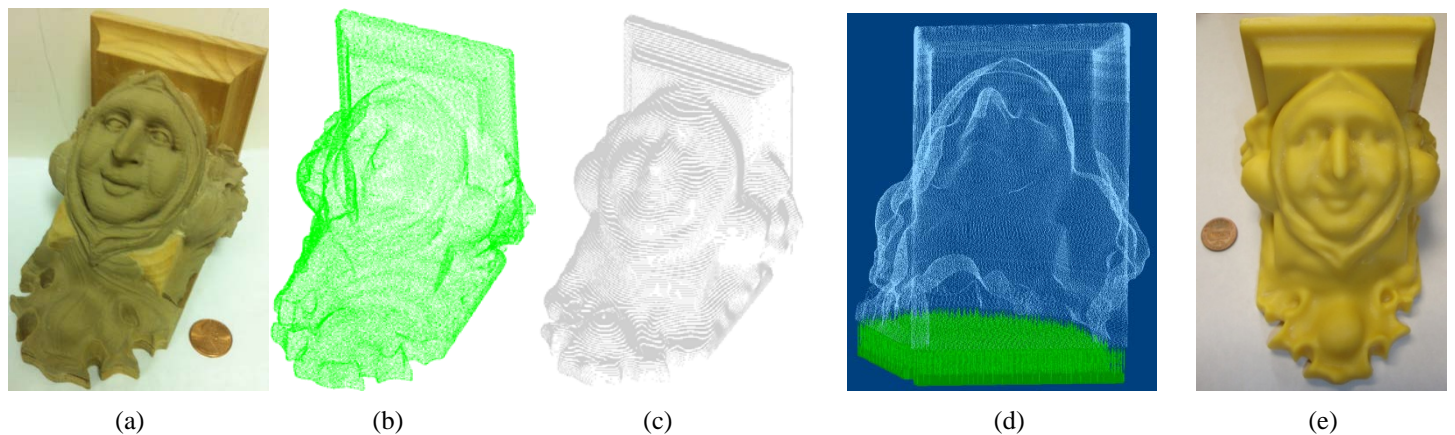


Figure 2. Direct geometric processing for tele-fabrication. The physical part in a) is scanned in Chicago and the scanned data cloud is shown in b). The data cloud is then sliced as shown in c). Upon transferring the data to Los Angeles, support structures (d) are automatically generated from the sliced model and a physical part is built as shown in e).

Such direct processing of raw scan data into forms that are directly usable in additive manufacturing preserves the data accuracy and is both efficient and flexible. The resulting tele-fabrication capability can be used in 3D copiers or 3D fax machines. The developed direct geometric processing methods thus have the potential to transform how shape information is processed and used in direct digital manufacturing. It can

potentially impact mass customization, part repair and service, and patient-specific bio-implant fabrication.

The remainder of the paper is organized as follows. The basic process of 3D digitization is presented in Section 2. The slicing procedure is shown in Section 3. Contour-based support generation is presented in Section 4 and mask generation for additive manufacturing is presented in Section 5. The experimental results of multiple test cases are presented in

Section 6. Finally conclusions and future work are given in Section 7.

2 3D DIGITIZATION OF PHYSICAL OBJECTS

As a preparatory step in our tele-fabrication approach, the physical object is first digitized by a 3D digitizer in Chicago, and the point cloud data is then obtained.

The digitizing system housed at IIT used in this paper is shown in Figure 3. It consists of three major components: digitizer, rotary stage and computer. The digitizer used here is the Non-contact 3D Digitizer Vivid 910 from Minolta Corporation. The rotary stage is equipped with a turntable where the 3D object is placed and laser-scanned by the digitizer; the turning functionality allows the object to be scanned on any desired viewpoint. The computer with the scanning software installed connects the digitizer and the rotary stage into an integrated system that performs the digitization task according to the user’s requirements.

The digitization process contains two stages: step scanning and registration. In the scanning stage, the object is scanned 6 times and it is turned 60 degrees at each step. The scanned images for the sculpture part are shown in Figure 4. The scanning software provided by the digitizer enables one to freely adjust the scanning parameters and the relative orientation between the lens and the object, so as to achieve a satisfactory scanning result. In the distance field plot for each scan, red means farther distance and blue closer. Useful parameters include the laser density, which controls the sampling density, and the step angle, which controls the incremental angle of the rotary stage. The scanned data from all the scanning steps are then aligned to form the entire model of the object in the registration stage.

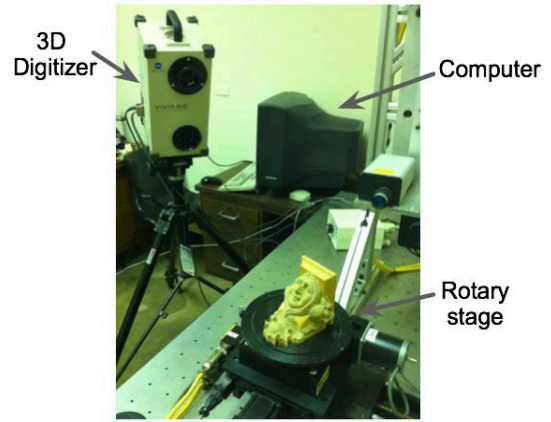


Figure 3. A digitization system and software system.

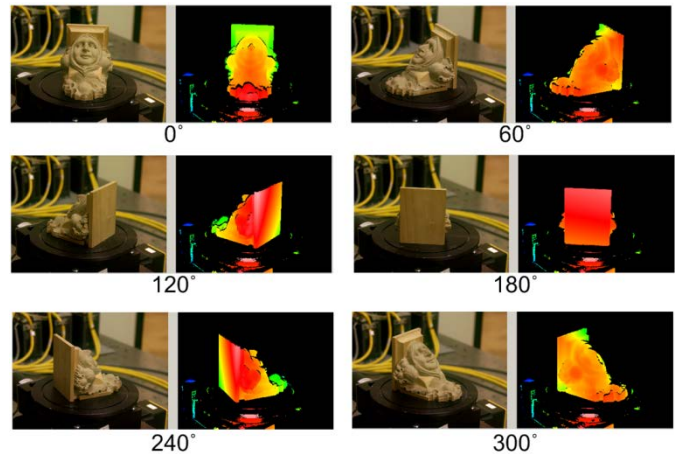


Figure 4. The 6 scanning steps with 60-degree step angle.

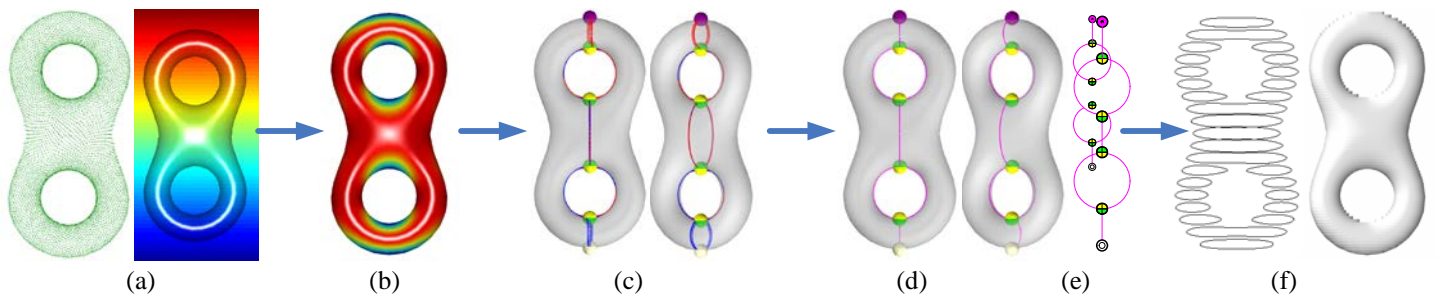


Figure 5. Overview of the Morse complex based point cloud slicing procedure. (a) point cloud. (b) Morse function on the MLS surface. (c) critical point generation. (d) Morse-Smale complex. (e) enhanced Reeb graph. (f) sliced model. The magenta, green & yellow and white Dots represent the maximum, (top and bottom) saddle and minimum critical points respectively.

3 POINT-CLOUD SLICING

Our slicing algorithm has evolved from the original single contour adaptive marching method [14] to a multiple contour slicing method with topology guarantee [13]. In this section we will briefly present this topology-guaranteed approach based on Morse complex and also a slicing approach based on the Marching Cube algorithm that can deal with degeneracy in a general way. The former deals with degeneracy-free shapes and special degeneracies; the latter works well with all kind of

geometries, especially ones with complicated topology. Both approaches generate the slice contour points by intersecting the slicing plane with the Moving Least Square (MLS) point set surface.

3.1 MLS surface

The MLS surface defines a smooth surface from a discrete data set. Levin [15][16] defined an MLS surface M as the stationary set of a projection operator. Such projection based MLS surfaces are referred to as projection MLS surfaces,

which can be explicitly defined as the local minima of an energy function along the directions given by a normal vector field $\mathbf{n}(\mathbf{x})$.

The normal vector $\mathbf{n}(\mathbf{x})$ at a location \mathbf{x} in space is just the average of the normals \mathbf{v}_i associated with all the points \mathbf{q}_i of point cloud \mathbf{Q} , weighted by a Gaussian weighting function θ of the distance between \mathbf{q}_i and \mathbf{x} :

$$\mathbf{n}(\mathbf{x}) = \frac{\sum_{\mathbf{q}_i \in \mathbf{Q}} \mathbf{v}_i \theta(\mathbf{x}, \mathbf{q}_i)}{\left\| \sum_{\mathbf{q}_i \in \mathbf{Q}} \mathbf{v}_i \theta(\mathbf{x}, \mathbf{q}_i) \right\|}$$

The energy function $e(\mathbf{y}, \mathbf{a})$ at location \mathbf{y} along direction \mathbf{a} is the average of the distance between \mathbf{q}_i and the plane that passes \mathbf{y} and takes \mathbf{a} as the planar normal, weighted by the same weighting function:

$$e(\mathbf{y}, \mathbf{a}) = \sum_{\mathbf{q}_i \in \mathbf{Q}} \left[(\mathbf{y} - \mathbf{q}_i)^T \mathbf{a} \right]^2 \theta(\mathbf{y}, \mathbf{q}_i)$$

Since the MLS surface is the loci where the directional derivative of the energy function along the normal vector direction vanishes, the MLS surface could be explicitly defined by:

$$g(\mathbf{x}) \equiv \mathbf{n}(\mathbf{x})^T \left(\frac{\partial e(\mathbf{y}, \mathbf{n}(\mathbf{x}))}{\partial \mathbf{y}} \Big|_{\mathbf{y}=\mathbf{x}} \right) = 0$$

3.2 Morse complex based slicing of MLS surface

The overall process of this slicing algorithm is shown in Figure 5. The idea of topological guarantee of slice contours will be discussed, and the main steps: critical points generation, Morse-Smale complex construction, enhanced Reeb graph building and adaptive marching will be briefly described. Detailed information could be found in [13][14].

3.2.1 Critical points generation

The concept of critical points is crucial for the topological analysis using Morse theory. Mathematically, the critical points for a given differentiable Morse function are those locations on the geometry where the gradient of the Morse function vanishes. For the slicing of point cloud, the Morse function chosen is the height function constrained by the point-set surface, which is represented by a Lagrangian function described in [14]. There are three types of critical points: *maximum*, *minimum* and *saddle*. The critical points of a torus is shown in Figure 6, where the saddles can be further categorized into *top saddle* and *bottom saddle* depending on their orientation, with normal pointing either opposite or along the slicing direction \mathbf{n}_H . The critical points are directly related to the topology change of the slicing contours as slicing proceeds along \mathbf{n}_H . The minimum, bottom saddle, top saddle and maximum correspond to creating, splitting, merging and destroying of slicing contour(s) respectively.

The critical points of the MLS surface could be obtained following the steps below:

1). Project input points onto the MLS surface defined by input point cloud with normals.

- 2). Calculate the normals of the projected points on the MLS surface and sift out those points with normals \mathbf{n}_S that span a sufficiently small angle with \mathbf{n}_H ;
- 3). Refine by further minimizing the angle in order to obtain the real critical points; the refinement is realized by a constrained optimization process.

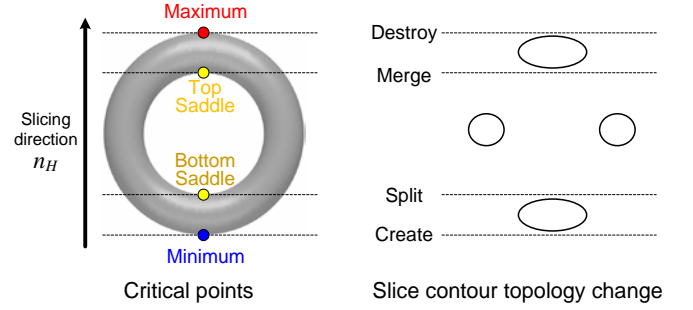


Figure 6. Critical points and slicing topology change.

3.2.2 Morse-Smale complex construction

After the critical points are identified and classified, two ascending and two descending integral lines can be traced from each saddle to maximums and minimums respectively, and all the integral lines constitute the Morse-Smale complex. Such complex of a mechanical part example with degeneracies is shown in Figure 7.

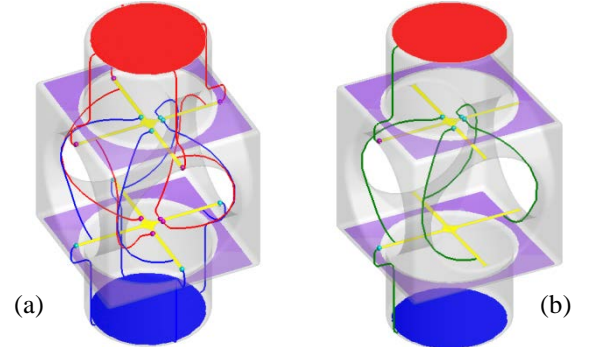


Figure 7. Morse-Smale complex and enhanced Reeb graph of a mechanical part. (a) Morse-Smale complex with ascending and descending integral lines; (b) Enhanced Reeb graph extracted from the complex.

3.2.3 Enhanced Reeb Graph building

Starting from the complex just generated, the enhanced Reeb graph is calculated by following the rules below:

- (1) Maximum or bottom saddle: if it has more than one downward arcs, keep the arc between this critical point and its nearest lower neighbors and prune the other arcs by trimming them at the height of the adjacent point and re-linking them to it as shown in Figure 8a;
- (2) Top saddle: first classify all its lower neighbors into two groups, and then do pruning (Figure 8b) and/or grouping (Figure 8c) to remove unwanted arcs in the complex;
- (3) Minimum: remove all downward arcs.

As an example, the enhanced Reeb graph extracted for the mechanical part is displayed in Figure 7b.

3.2.4 Curvature-Adaptive Contour Marching

The point data is ready for slicing after the enhanced Reeb graph is available. The topology of the slice, i.e. the number of contours, is immediately known from the intersection points between the slicing plane and the enhanced Reeb graph. Still taking the part example as shown in Figure 9, intersecting with the enhanced Reeb graph at two different slicing layer heights will reveal different topologies of the slice contour(s). In this case, the top row has 1 contour and the bottom row has 4 contours, depending on the number of intersection points.

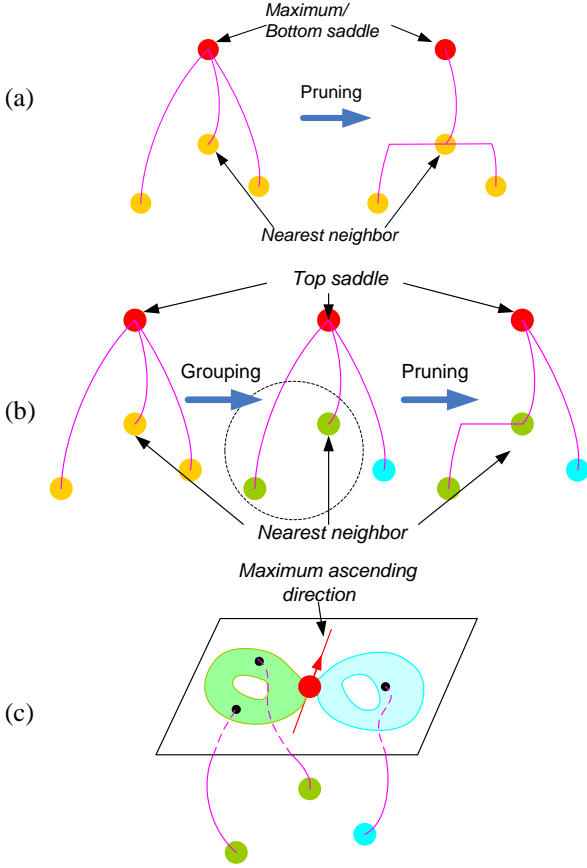


Figure 8. Rules of generating an enhanced Reeb graph from the Morse-Smale complex. (a) rule 1 for maximum and bottom saddle; (b) rule 2 for top saddle; (c) grouping the lower neighbor points in rule 2.

For each contour at a layer, the contour points could be generated by a marching procedure featuring adaptive step length controlled by MLS surface curvature [14]. With the above curvature control, the adaptive contour points could be generated without difficulty. A single marching step is illustrated by Figure 10a, and detailed implementation follows the steps below:

- 1). Identify starting point c_0 as the intersection point between the slicing plane and the enhanced Reeb graph;
- 2). Compute all the contour points $\{P_i\}(i=0,1,\dots)$:

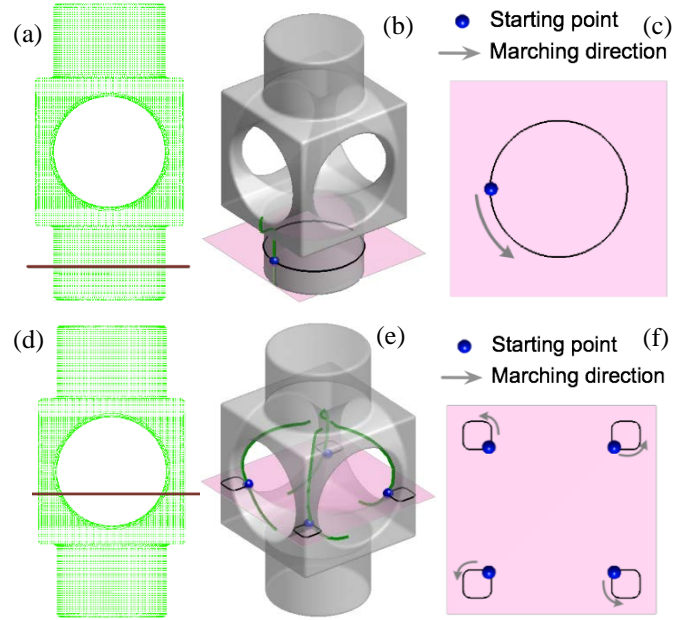


Figure 9. Identify the slice's topology by intersecting the slicing plane with the enhanced Reeb graph. (a)(d) slicing planes; (b)(e) intersecting with the enhanced Reeb graph; (c)(f) adaptive contour marching.

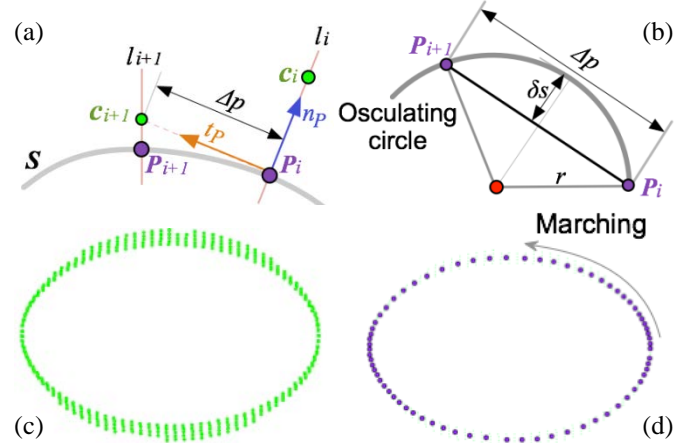


Figure 10. Adaptive contour marching. (a) intersection-based marching with adaptive step length; (b) step length determined by osculating circle radius; (c) partial point cloud near the slicing plane; (d) contour points generated by marching with curvature-adaptive step lengths.

- a). Calculate the normal direction \mathbf{n}_i at c_i , and an intersecting line \mathbf{l}_i could be determined;
- b). Find the intersection point P_i between \mathbf{l}_i and the MLS surface S ;
- c). Get the tangent direction \mathbf{t}_i and curvature κ at P_i ;
- d). Calculate step length Δp at P_i by

$$\Delta p = 2\sqrt{r^2 - (r - \delta_s)^2} = 2\sqrt{2r\delta_s - \delta_s^2}$$

where radius of the osculating circle is $r = 1/|\kappa|$, and δ_s is

the allowed approximation error, and the geometric relation between this error and the circle radius is demonstrated in Figure 10b;

3). March to next point by $\mathbf{c}_{i+1} = \mathbf{P}_i + \Delta p \cdot \mathbf{t}_i$;

4). Set $\mathbf{c}_i = \mathbf{c}_{i+1}$, and go to step 2.1 until \mathbf{P}_i is close enough to \mathbf{P}_0 judging by a user-specified threshold.

Given the point cloud in Figure 10c (only a portion near the slicing plane is plotted to show the rough shape of the intersection curve with the embedded MLS surface), the above algorithm will generate the sequential contour points in Figure 10d, which will be then connected then to form the contour previously shown in Figure 9c and f.

3.3 Marching Cube based slicing of MLS surface

Since Morse theory is only applicable to non-degenerate cases, in order to deal with topological complexity in a more general way, we here also developed a Marching Cube based slicing approach. The idea is simple: as the MLS surface is an implicit surface defined by $g(\mathbf{x}) = 0$, it's possible to extract the surface by locating the zero-value contours of the $g(\mathbf{x})$ scalar field. In the context of slicing, the surface extraction problem reduces to the 2D contour extraction from the $g(\mathbf{x})$ scalar field on the slicing plane, which makes the slicing even more efficient.

As an example in Figure 11a, point cloud of the sculpture shown in green is sliced by the slicing plane shown in yellow, and the red point cloud subset within a small distance to the slicing plane is intended to show the relative position of the plane. The Marching Cube based slicing algorithm goes in the following steps for a given slicing plane:

1). Build a regular grid large enough to encompass the entire projected point cloud on the slicing plane Figure 11b. Here a 350×350 grid is chosen in the area that marginally bounds red point cloud subset. Figure 11c gives a zoom-in view of how the grid looks like.

2). Compute the $g(\mathbf{x})$ value at all the grid points, which yields the $g(\mathbf{x})$ scalar field displayed in Figure 11d. Note, grid points that are far away from the red subset points will not be computed for speed concern. As a result, these areas are shown in blank white.

3). Extract zero-value contours from the scalar field, and pick from all candidate contours those ones close enough to the MLS surface. Note there are spurious contours nearby, and we only pick the contour that is close to the projected MLS surface. The closeness from an extracted contour with points $\{\mathbf{P}_i\}$ to the MLS surface is measured by the mean distance from each contour point \mathbf{P}_i to its projection onto the MLS $MLS(\mathbf{P}_i)$, and the picking criterion could be written by

$$d = \frac{\sum_{i=1}^n \mathbf{P}_i - MLS(\mathbf{P}_i)}{n} < d^*$$

where MLS is projection operator and d^* is the distance threshold set to be half the Gaussian kernel.

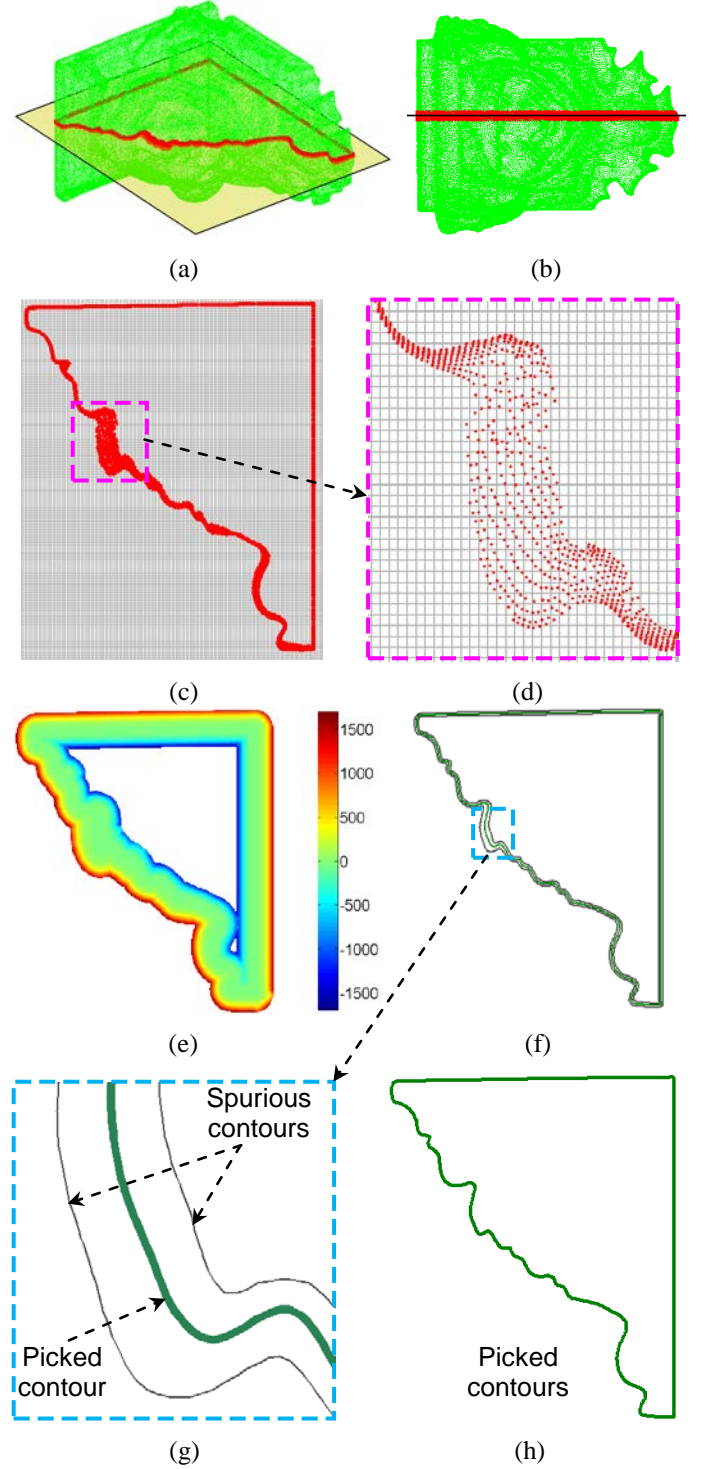


Figure 11. Marching cube based MLS surface slicing. (a) slicing plane (yellow) and points (red) close to the plane; (b) front view; (c) marching cube grid on the slicing plane; (d) zoom-in view; (e) $g(\mathbf{x})$ field; (f) 3 extracted zero-value contours; (g) zoom-in view only the middle contour (green, thick) is taken as the slicing contour; (h) output contour for the current slicing plane.

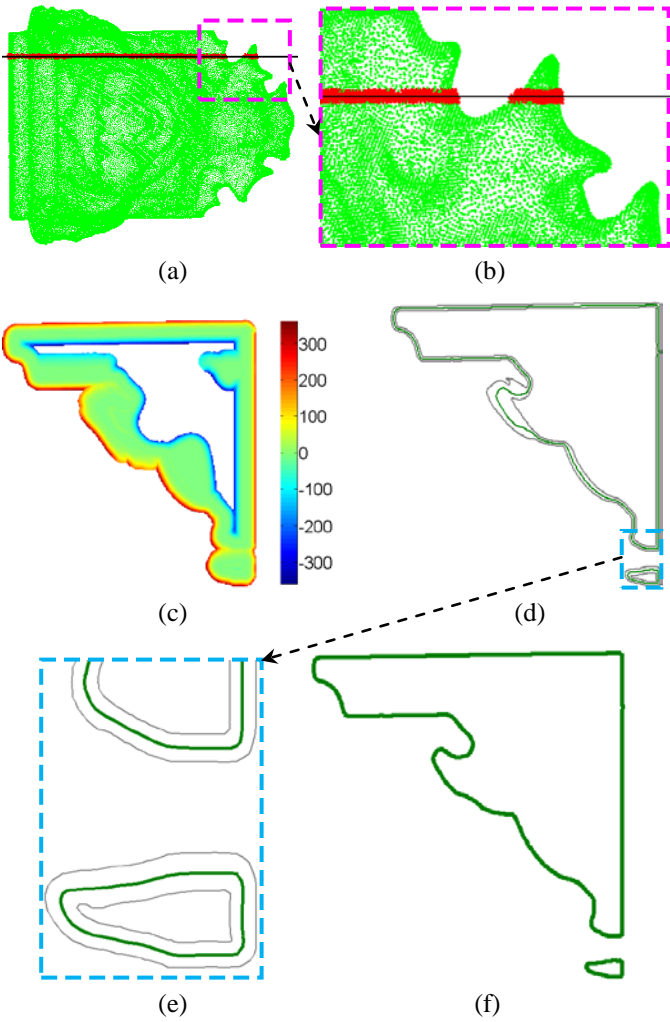


Figure 12. Slicing with topological guarantee. (a) another slicing plane where 2 contours are expected; (b) zoom-in view; (c) scalar field; (d) 6 contours extracted; (e) contours zoom-in; (f) 2 contours picked as actual slice contours.

Here the Marching Cube based slicing directly produces the correct number of contours by the combination of $g(x)$ scalar field's zero-value criterion and the closeness to MLS surface contour picking criterion. This way of guaranteeing the topology of slices proves very effective and robust in practice.

If we consider another slicing plane shown in Figure 12a where 2 contours are expected, initially 6 contours will be extracted from the scalar field, and later only the 2 middle contours will be identified as the ultimate contours to output by the distance to MLS surface picking criterion.

4 CONTOUR-BASED SUPPORT GENERATION

Layer-based additive manufacturing (AM) is a collection of techniques for fabricating solid objects by the sequential delivery of energy and/or material to specified points in space to produce that solid [21]. For the direct processing of scanning data without generating STL models, we explore the AM process planning methods based on a digital model that is defined in slicing contours.

In the paper, a contour-based support generation method and related algorithms are presented for the mask-image-projection-based Stereolithography (MIP-SL) process. Supports are similar to fixtures in machining or scaffolds in construction. Support structures are typically required to facilitate the building process in the Stereolithography Apparatus (SLA) process. During the building process, a cured portion needs to be anchored such that it will not sink to the bottom of the liquid resin tank, or float away when a blade is used to flatten the liquid surface. Appropriate anchor positions and structures for a given geometry are critical in ensuring the success of the building process. At the same time, the added supports are a waste of material and leave undesired marks on the contacting building surface. It is desired to add a minimum amount of supports that can be easily removed after the building process.

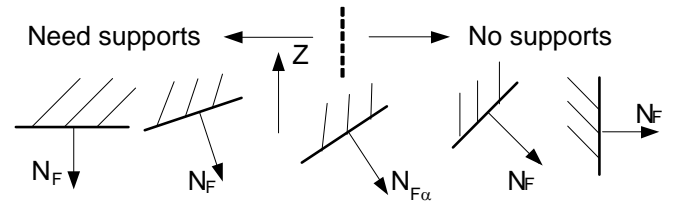


Figure 13. An illustration of the angle-based support generation method.

In the previous work on support generation for AM processes, most approaches are developed for STL models [22][23][24][25]. The most common approach developed for support generation is to judge all the triangles in a STL model by comparing their orientation angle (relative to the Z axis) with a minimum supporting angle (α) that is specified by a user. As shown in Figure 13, supports will be created for a triangle whose face normal N_F satisfies $N_F \cdot Z < N_{F\alpha} \cdot Z$, in which the face normal related to the given minimum supporting angle is $N_{F\alpha}$. The approach has been widely used in commercially available systems such as Lightyear from *3D Systems Inc.* (Rock Hill, SC) and Magics RP from *Materialise* (Leuven, Belgium).

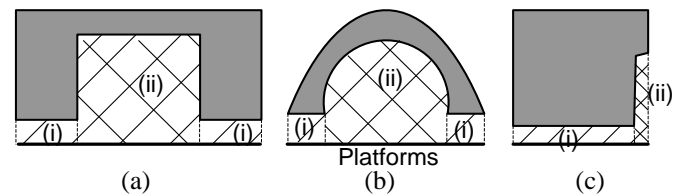


Figure 14. An illustration of various geometries and related supports. (a) Cantilever; (b) Vaulted overhang; (c) Small overhang.

Chalasaniet *al.* [26] presented a support generation method for the Fused Deposition Modeling (FDM) process based on two-dimensional sliced geometry. It computed supports by determining the shadow of all the layers with respect to the build direction Z. Figure 14 shows some examples, in which regions (i) and (ii) are the required supports. Such supports can be used in the FDM process since two different materials including a water-solvable support material are used. However, such supports cannot be used in

the SLA process since a single material is used for both part and supports.

In addition, the aforementioned support generation methods do not take full advantage of the self-supporting property of geometric features in the SLA process. That is, for a vaulted overhand or a small overhang as shown in region (ii) of Figure 14b and Figure 14c respectively, no supports are needed since all the layers of such geometric features can be built based on the previously built layers. The self-supportness of a geometric feature will be fully utilized in our contour-based support generation method that is based on a layer-wise analyzing approach.

4.1 Principle

The additive manufacturing processes build physical objects layer-by-layer. Accordingly our support generation approach is to analyze the sliced contours layer-by-layer for determining where to add supports that are critical for the building process. A simple 1-dimensional example is shown in Figure 15a to illustrate our contour-based support generation approach.

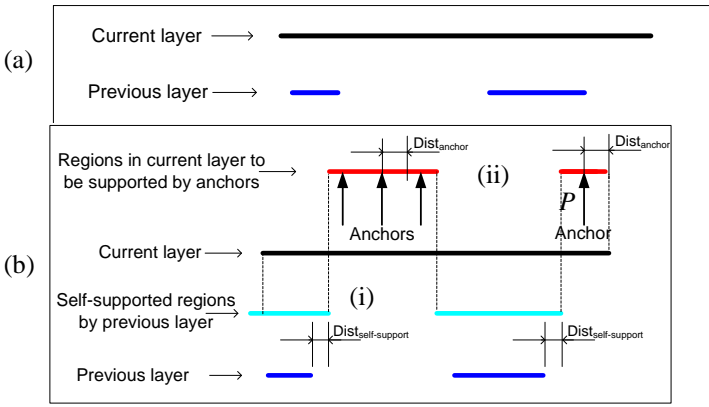


Figure 15. An illustration of the contour-based support generation method. (a) Given layers; (b) Layer analysis result.

Suppose the relative sizes and positions of a current layer and its previous layer are known. All the previous layers have been built when the current layer is to be built. Obviously the portions of the current layer that directly contact the previous layer have been supported. In addition, certain neighboring areas are also supported by the previous layer (refer to Figure 15b, assume by enlarging the previous layer by $Dist_{Self-support}$). The value $Dist_{Self-support}$ of a region of the previous layer may be determined by its size. A larger $Dist_{Self-support}$ value can be set for a bigger region. For the remaining portions of the current layer that have not been anchored by the previous layer (i.e. regions ii), supports need to be added under them in order for them to be anchored. In this paper, we define the additional pins that are added under a given part as *anchor supports*. Such anchors have been fixed on the building platform or previously built layers when the current layer is to be built. Assume when an anchor is added at position P , the circular region centered at P with a radius of $Dist_{Anchor}$ can be safely built (refer to Figure 15b). Accordingly all the regions (ii) are fully supported after a certain number of anchors have been added. There are many ways of defining

the shapes and sizes of the added anchors. The value of $Dist_{Anchor}$ can be set accordingly. For example, if rigid anchors are used in a given region, a large $Dist_{Anchor}$ value can be set for the anchors. Hence less supports will be required; however, such strong anchors will also be hard to remove and will leave larger marks on the contacting surfaces.

The principle of our contour-based support generation approach is to systematically analyze the shape of each layer related to its previous layer; accordingly the added anchor supports can be determined such that the layer can be fully fixed either by the previous layer or the anchor supports. After all the layers of a 3D model have been analyzed, related supports including bases and reinforcements are constructed to ensure the building process will be successful. The contour-based support generation approach can be mathematically defined as follows.

```

For layers of  $L_{Cur}, L_{Prev}$ ,
For each loop  $L_{Cur_i}$ ,
 $L_{Attached_i} = L_{Prev} \cap L_{Cur_i}$ ;
 $L_{Offset_i} = L_{Attached_i} \uparrow^{Dist_{self-support_i}}$ ;
 $L_{self-support_i} = (L_{Offset_i} \cap L_{Cur_i})$ 
 $L_{To-support} = L_{Cur} - L_{self-support_i}$ 
For each loop  $L_{To-support_i}$ ,
Cover the region with anchors based on  $Dist_{anchor}$ .

```

In the equations, \uparrow^r is the offset operation of a 2D region by growing it a distance r [27]. \cap and $-$ are the intersection and subtraction operations of two 2D regions, respectively.

The contour-based support generation approach is general that can handle various types of overhangs including the self-supported cases as shown in Figure 14. The two main parameters used in our approach, $Dist_{self-support}$ and $Dist_{anchors}$ have clear association with the building process. In comparison, the aforementioned STL-based support generation method is mainly based on geometric analysis. The main parameter used in the approach is the minimum support angle α , which has no direct association with the layer-wise fabrication process. In addition, the anchoring forces required in the SLA process may be different depending on the types of resins, and the building process settings such as sweeping speed, curing styles, etc. For different manufacturing settings, the values of $Dist_{self-support}$ and $Dist_{anchors}$ can be experimentally determined. Accordingly such values can be provided by the SLA system developers and used in our approach.

4.2 Main Algorithms

There are three main 2-D operations in the aforementioned contour-based support generation method including offsetting, Boolean, and region covering. For a 3D model defined in a set of contours, the operations need to be robust and general. The algorithms used for such geometric operations are discussed as follows.

(1) Offsetting operation.

Offsetting a solid S by a distance r into a grown or shrunk version of S has been precisely defined for point sets in *Euclidean space* E^2 or E^3 . Although the offsetting operation is mathematically well defined, computing an offset model for a

given solid has proven to be difficult. We developed a novel offsetting method based on a point representation named the *Layered Depth-Normal Image* (LDNI) for an input polygonal model and an offset distance [27]. The offsetting method is based on: (1) directly computing offset boundary, (2) converting the boundary into structurally sampled points, (3) accordingly filtering the sampling points, and (4) reconstructing offset contour from the filtered points. The key benefits of the method are that the related geometric operation can be general, robust, and efficient. Various types of self-intersections in the offset boundary can be trimmed such that a valid offset boundary can be constructed.

Figure 16 presents an example taken from the sliced model of the scanning points as shown in Figure 2. Two consecutive layers (Layers 106 and 107) are shown in Figure 16a and Figure 16b respectively. The computed offset contours of Layer #106 (i.e. $L_{Attached}^{\uparrow Dist_{self_support}}$) are shown in Figure 16c.

(2) Boolean operations.

As one of the most fundamental geometric operations in computer-aided design and manufacturing (CAD/CAM), a Boolean operation, such as union, intersection or difference, is well-defined based on set operations. In order to achieve robust computation, we also convert the continuous boundary representation into sampling points and then compute their Boolean result [29]. Based on the LDNI representation, the Boolean operations are straightforward and easy to implement. This is, a LDNI model consists of a set of well-organized one-dimensional (1D) volumes defined by the even number of depth-normal samples stored in each pixel of the LDNI model. Consequently, the Boolean operations on LDNI models are converted into the Boolean operations on 1D segment. After Boolean operations, the computed LDNI model is an implicit representation of a solid defined by the geometric operation. A contouring method [30] can then be used in reconstructing a polygonal model from the LDNI model.

An example is shown in Figure 16. Based on the computed offset contours, the subtraction between the current layer and the offset contour is computed. The Boolean operation result is the region to be supported by additional anchor supports.

(3) Region covering operation.

Based on the computed regions that are to be supported by anchors, we formulate the anchor layout problem into a 2D region covering problem. Suppose an anchor S_i that is to be added in a 2D region at position $P(x_i, y_i)$ to secure a small region that is centered around P with a radius $Dist_{anchor}$. A set of anchors need to be identified such that the whole region can be fully secured by the added anchor supports. That is, for any point P in the region, the closest anchor support S_k should satisfy $\|PS_k\| \leq Dist_{anchor}$. In addition, it is desired to uniformly distribute the anchors in the to-be-supported regions such that a minimum number of anchors will be used. In the commercially available support generation systems such as Lightyear and Magics RP, a set of pre-defined grid lines are used. Although being intuitive, such an approach will lead to

anchor supports with varying density for irregular 2D regions, especially for the anchors that are close to the region boundary. In comparison, the distribution of anchor supports can be more uniform by converting the support layout problem into a region covering problem.

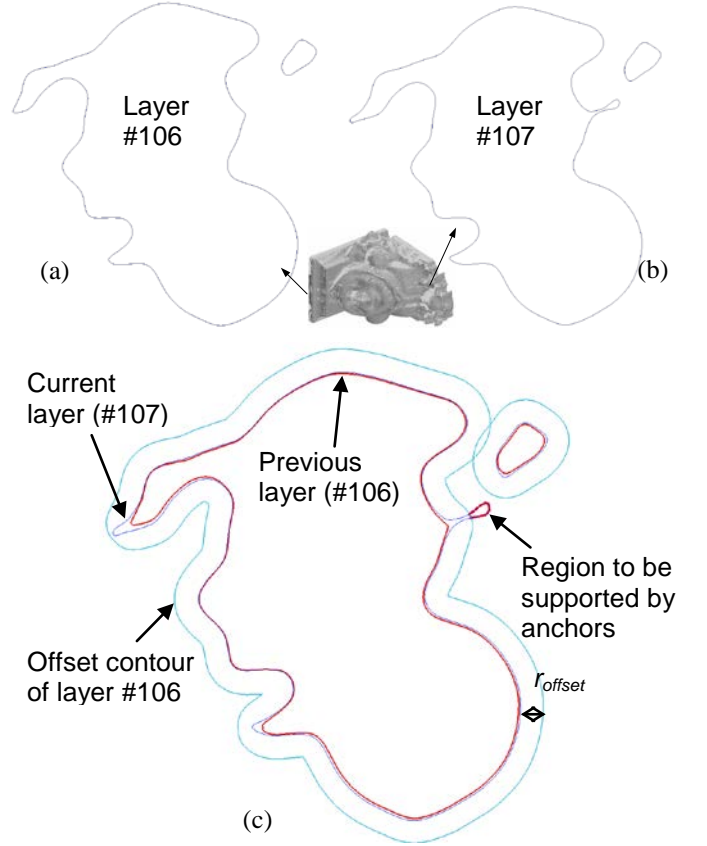


Figure 16. An example of layer 107. (a) Previous layer; (b) current layer; (c) computing result.

The region covering problem has been well studied [31][32][33], We develop a modified Delaunay triangulation method for finding a small amount of anchors that can sufficiently cover an arbitrarily given 2D region. The main algorithm of the method is given as follow.

- (1) Triangulate a contour L_i into a 2D region R_i ;
- (2) Compute an initial number of supports based on $Num_{anchor} = A_{R_i} / A_{anchor}$;
- (3) Randomly place N_{anchor} points at Pos_{anchor} in R_i ;
- (4) Create Delaunay triangulation of R_i based on current Pos_{anchor} ;
- (5) Find the centroid points of the related Voronoi partition;
- (6) Move Pos_{anchor} to the computed centroid points;
- (7) Repeat from (4) until Pos_{anchor} converges to centroid points of the Voronoi partition;

Check the length of all the partition edges. If any edge length is bigger than $Dist_{anchors}$, increase N_{anchor} to a larger value and repeat from (3).

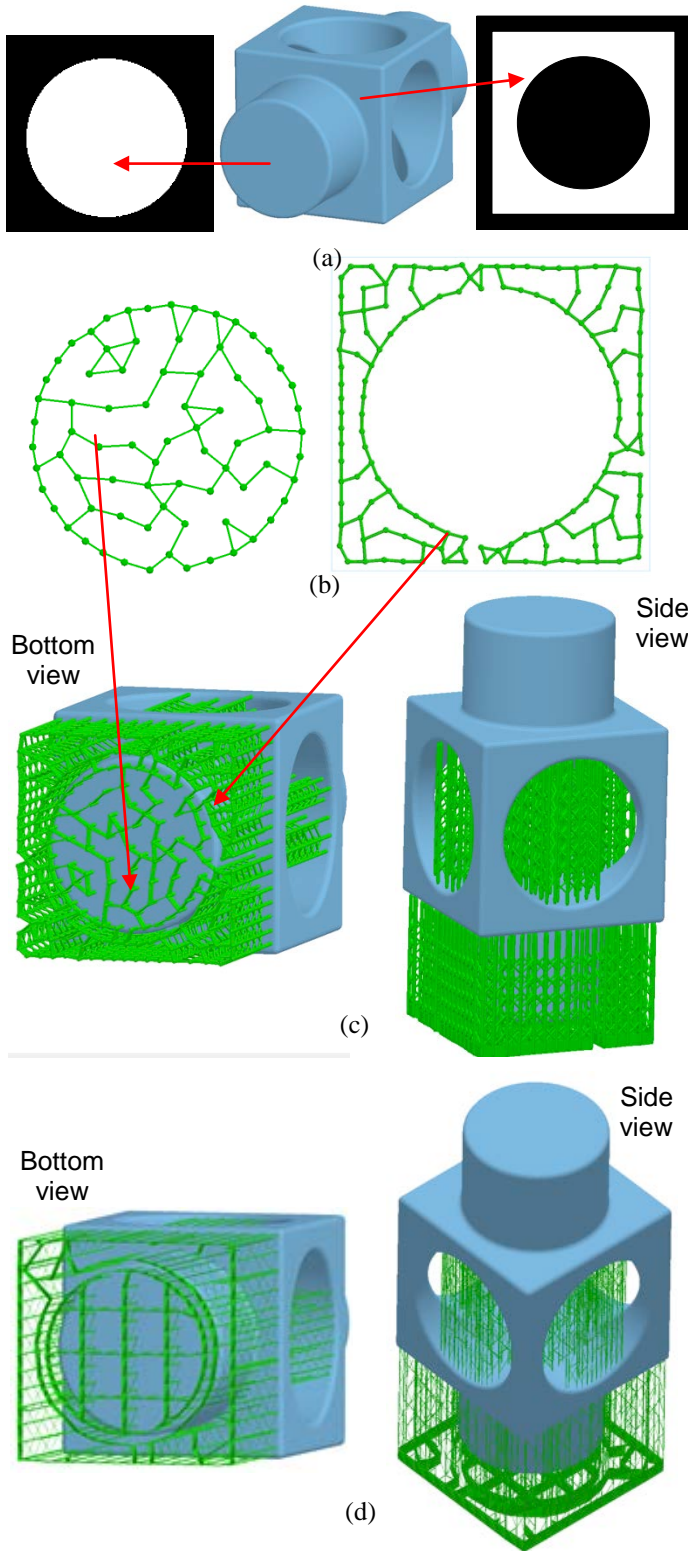


Figure 17. An example of support layout based on region covering. (a) Input regions to be supported; (b) computed support layouts; (c) CAD model of generated supports; (d) CAD model of generated supports by the Lightyear system.

In the algorithm, the computation of a Delaunay triangulation and related Voronoi partition for a given set of

points can be found in [31]. The Centroidal Voronoi Tessellation (CVT) method has been well studied as a powerful computational technique for minimizing an energy function defined by a density function [32][33] and proven to be convergent.

The computed N_{anchor} and related Pos_{anchor} can be used as the layout for adding related anchor supports. Many different support types have been developed before for the SLA process such as the *Fine point* supports developed by *3D Systems Inc.* Anchor supports used in our research are shown in Figure 17c. The neighboring anchor supports need to be connected together for an increased stability. In addition, the bottom portions of the anchor supports are connected into a base such that they can be securely attached to the building platform.

Figure 17 shows an example, in which two of the to-be-support regions (in white) are shown in Figure 17a. The computed anchor supports for a given $Dist_{anchor}$ are shown in Figure 17b. The related anchor supports including bases and reinforcements between neighboring pins are shown in Figure 17c. In a comparison, the supports generated by the Lightyear system are shown in Figure 17d. By using a set of pre-defined grid lines, the supports are distributed non-uniformly in the regions; in addition, much more supports are required than the supports used in our approach (105 vs. 78 and 174 vs. 136 for the two regions as shown in Figure 17b).

5 MASK IMAGE PLANNING FOR THE PROJECTION-BASED STEREO LITHOGRAPHY

Based on the computed anchor supports for a given set of sliced contours, the mask images that can be used in the building process will be prepared. A specified exposure time will be assigned for each layer. In the section, the scanning point model as shown in Figure 2 will be used as an illustration example. Assume a different part orientation has been picked by the user for achieving a smaller Z height. As shown in Figure 20, the sliced model based on the given Z axis can be computed using the aforementioned point-cloud slicing approach; the anchor supports based on the contour-based support generation method can also be determined. Accordingly, mask projection images can be prepared for building the physical object.


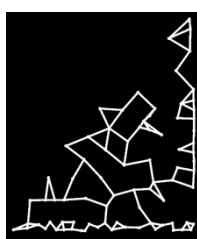
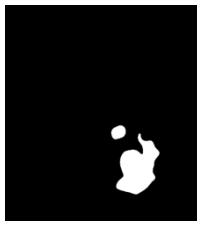
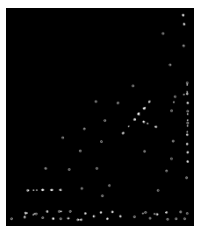
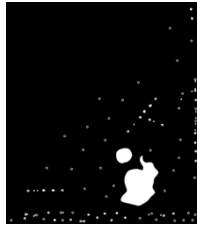

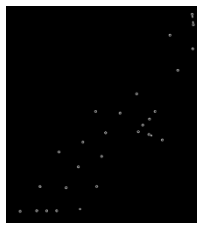



5.1 Part and Support Images

Each contour in the sliced model defines the layer boundary. They can easily be converted into a black and white mask image such that all the pixels inside the contour is white (i.e. grayscale value=255) and all other pixels are black (i.e. grayscale value=0). However, as shown in [34], such a binary image will lead to aliasing effect due to the sampling error in converting the continuous contour boundary into discrete pixel values. In addition, the energy distribution of a pixel follows a *Gaussian* distribution and spreads to its neighboring pixels. The light intensity of a pixel also varies for different grayscale levels and can be calibrated [35]. Hence the projection mask image for a given contour can be computed by modeling the interaction between all the pixels such that the projected light can be controlled to achieve a desired accuracy and resolution. An *optimized pixel blending* method [34] has been developed,

in which a grayscale mask image will be computed for achieving the desired pixel blending effect required by given slicing data. In addition, various exposure patterns can be used in defining the internal pixels such that the curing process will lead to less shrinkage and related deformation in the built object [36].

The mask image of the supports can be computed by directly slicing the geometric elements related to the computed anchor supports. The exposure time of anchor supports can be different from that of the parts. In addition, the bottom few layers require an expose time that is much longer than other layers such that they can be fully bonded to the building platform. Based on the mask images of parts and related anchor supports, a final projection image can be generated for the building process by simply merging the related two mask images. Table 1 shows the mask images of some sampled layers related to the part as shown in Figure 20.

Table 1. Mask projection images of some sampled layers.

Layer #	Mask image of part	Mask image of supports	Projection mask image
1	None		
100			
200			
300		None	

5.2 Building Process

Based on the technology of *Digital Micromirror Device* (DMD), a mask image can be projected onto a photocurable resin surface to selectively cure liquid resin into a layer of the object. Consequently, the MIP-SL process can be much faster than the laser-based SLA process. During the building process, the computed mask images are sent to the DMD, which are then exposed to the resin surface for a specified time period. Liquid resin is recoated after a layer is cured. By repeating the layer-by-layer building process, a 3D object can be fabricated on the building platform. After all the layers are finished, the platform is raised up for the built object to be removed. The added supports are peeled away from the attached part surface and discarded. Finally the built part is cleaned.

Three MIP-SL systems with different platform sizes and resolutions are used in the tests including a commercially available system *A* (Ultra machine from EnvisionTec Inc.), and two internally developed systems *B* and *C* [37][38]. Two types of photocurable resins are used in the tests including Perfactory™ SI500 and R5, both from EnvisionTec Inc. The exposure time of each layer is 9, 0.4, and 0.5 seconds for *A*, *B*, and *C* respectively. The capabilities of the three tested MIP-SL systems are shown in Table 2. Based on them, an integrated tele-fabrication system may have various fabrication capabilities in reproducing existing physical objects in different sizes.

6 EXPERIMENTAL RESULTS AND ANALYSIS

Two test cases are presented as follows to demonstrate the presented direct geometry processing approach for tele-fabrication of physical objects. The scanning and slicing of existing physical objects were performed in Chicago at IIT. The process planning and fabrication based on MIP-SL systems were performed in Los Angeles at USC. The experimental results have demonstrated the potential applications of 3D copying and 3D faxing of a physical object. The presented geometry processing methods can be automated and integrated into a seamless tele-fabrication system.

6.1 Fabrication of Two Test Models

6.1.1 Test 1: fertility model

The first test is a statue model called “fertility” from AIM@SHAPE repository [39]. Since the model has already been digitized by repository contributors, the digitization is not needed and the triangle mesh vertices have been used as the input point data for slicing shown in Figure 18a. The point data is then directly sliced based on a selected building direction and a given layer thickness. The Morse complex construction result and enhanced Reeb graph are shown in Figure 18b and Figure 18c respectively. Based on them, the sliced contours are shown in Figure 18d. After support generation and mask image preparation, the built physical object before being taken out from the platform is shown in Figure 18e. Hence the physical object is successfully duplicated after the supports are removed and the part is cleaned.

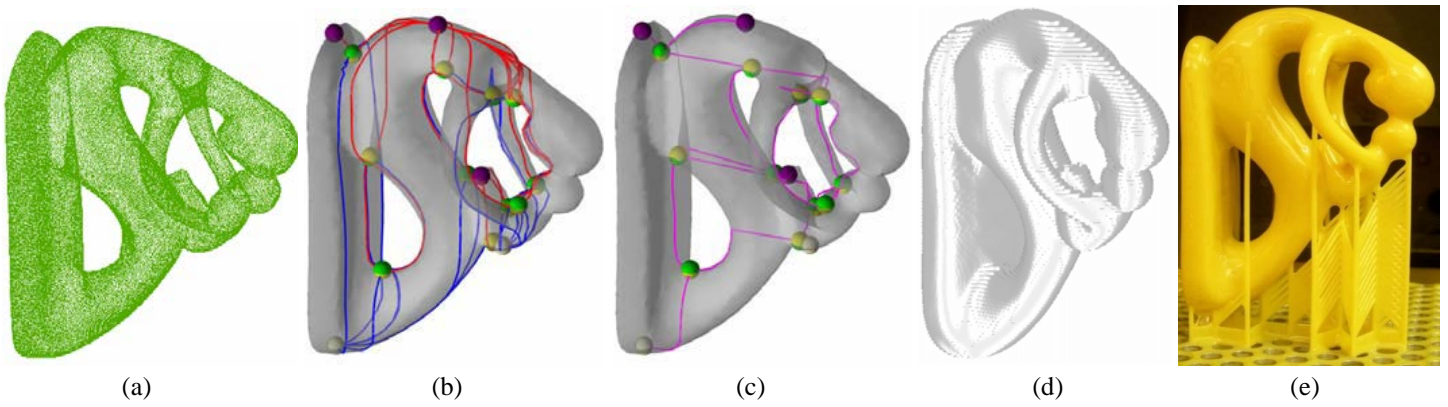


Figure 18. Tele-fabrication test 1: fertility model. (a) Scanned point-cloud data; (b) critical points and Morse complex; (c) enhanced Reeb graph; (d) sliced model; (e) built part.

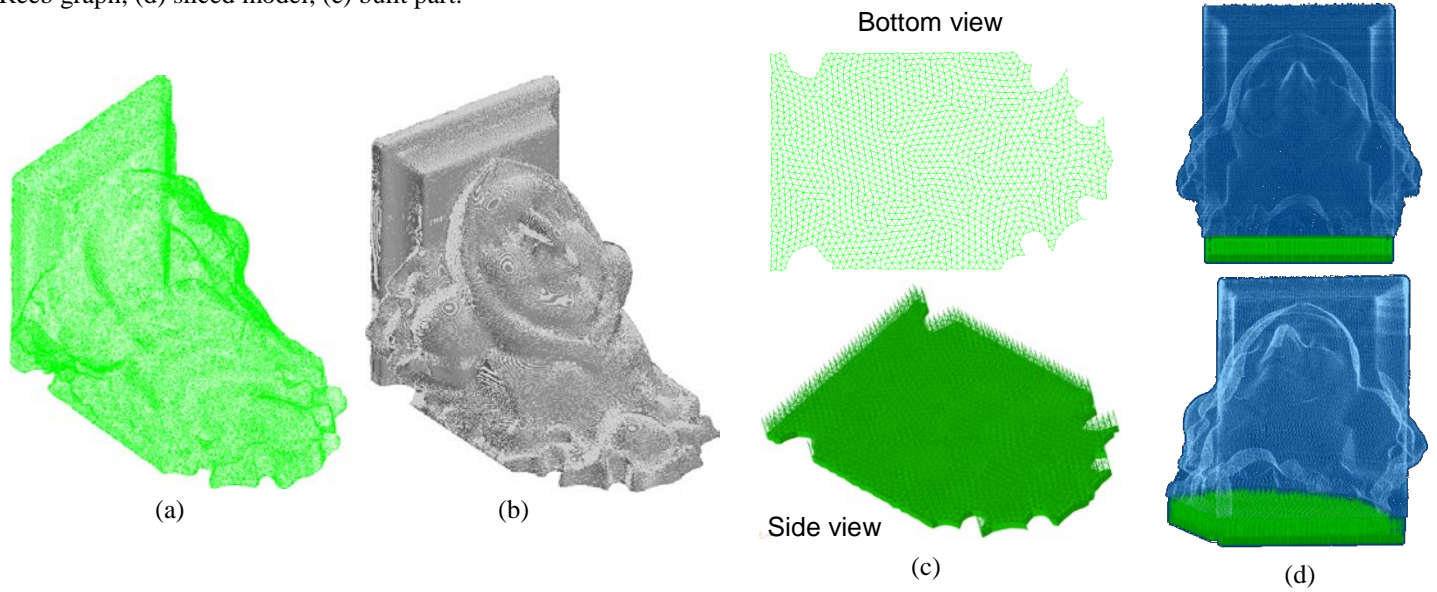


Figure 19. Tele-fabrication test 2: sculpture model at original orientation. (a) Scanned point-cloud data; (b) sliced model; (c) generated supports; (d) display of both point clouds and supports.

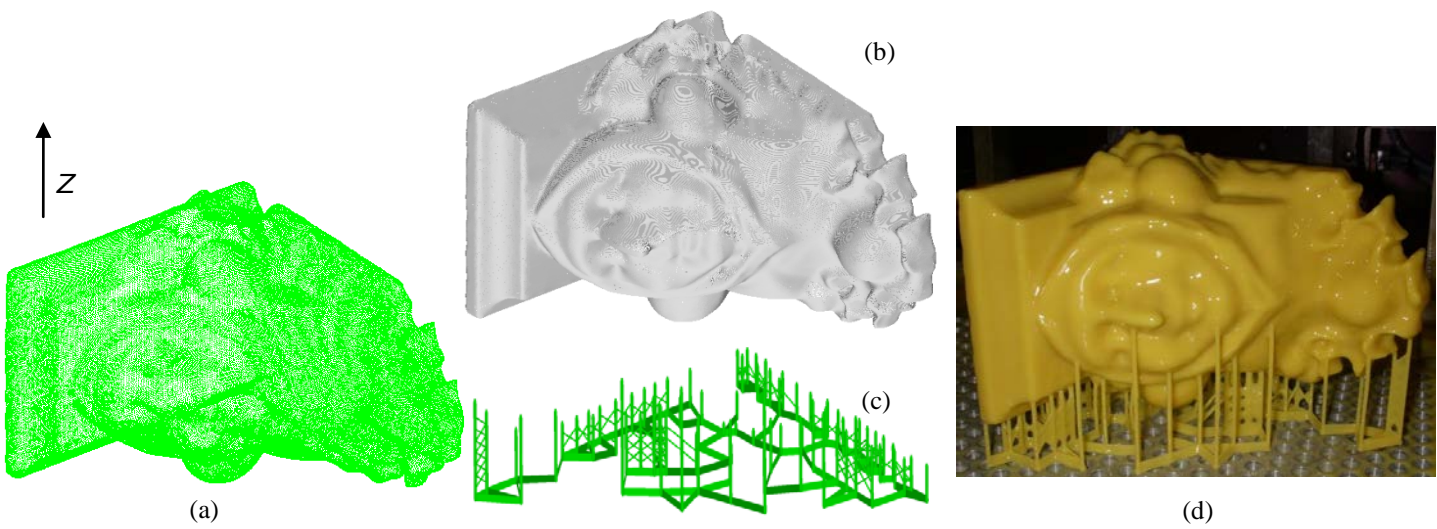


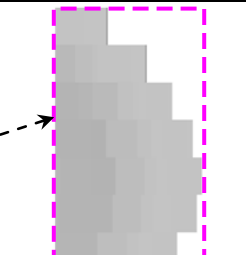


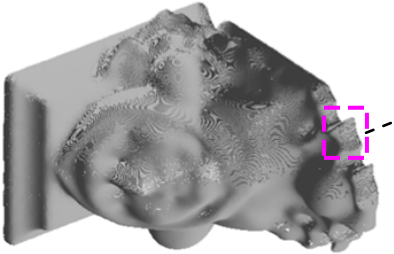
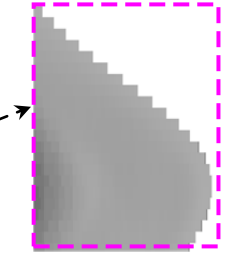


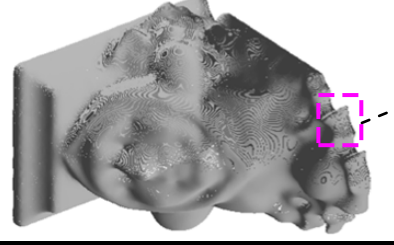
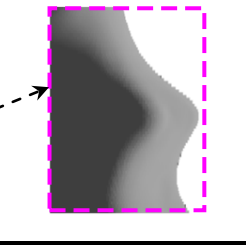




Figure 20. Test 2: sculpture model at a different orientation with fabrication machine A (layer thickness=0.01mm). (a) Scanned point-cloud data; (b) sliced model; (c) generated supports; (d) built part with attached supports structure.

Table 2. Sliced model, support structure and built part under three AM systems.

AM System Compatibility	Sliced Model		Support Structure	Built Part
	Global view	Local view 		
Machine A <u>Platform size</u> 260 X 160 mm <u>XY resolution</u> 0.14mm <u>Z layer</u> 0.15 mm				
Machine B <u>Platform size</u> 48 X 36 mm <u>XY resolution</u> 0.047mm <u>Z layer</u> 0.05 mm				
Machine C <u>Platform size</u> 14 X 10.5 mm <u>XY resolution</u> 0.014mm <u>Z layer</u> 0.01 mm				

6.1.2 Test 2: sculpture part

The second test is performed on a physical sculpture part. The physical object comes from a local craftsmanship decoration firm in Chicago. It is shown in Figure 2a next to a penny for scale reference. We first fabricate the object with the original orientation, then rotate the object and rebuild it at a different orientation for demonstrating the ability of supports generation and meanwhile choose fabrication machine A in Table 2 for slicing parameters selection. In addition, we use the second orientation but employ two different fabrication machines B and C and compare the results under three circumstances.

When the scanned point data is kept at its original orientation as shown in Figure 19a, it is directly sliced using the Marching Cube based slicing algorithm with a given layer thickness, and the sliced model is shown in Figure 19b. For such a building orientation, only the bottom surface needs supports. The identified support locations based on the discussed region covering approach are shown in Figure 19c for such a bottom surface. Figure 19d shows the display of both point data and the generated supports with two different viewing angles.

The scanned point data can also be built in another orientation using a different layer thickness of 0.1mm as prescribed by fabrication machine with machine A in Table 2. For an orientation as shown in Figure 20a, the sliced model is obtained and shown in Figure 20b and generated supports are shown in Figure 20c. The prepared mask images for some of the layers are shown in Table 1. The built physical object using the MIP-SL system machine A is shown in Figure 20d. After removing the supports and cleaning the built model, a photo of the duplicate object is shown in Figure 2e.

Other than machine A, one can also choose machine B or C for fabrications in smaller scales. In Table 2, we have shown the sliced model, support structure and the fabricated objects with the capabilities of the three additive manufacturing systems. The three machines A, B and C have different platform size, XY resolution and Z layer resolution, which are described in the table. In the column of the sliced model, the parameter of slice thickness is guided by the capabilities of the related manufacturing system. The layer thickness of machine C is nearly one tenth of that of machine A. These adaptive function w.r.t machine parameters can be achieved by the slicing algorithm of scanning points. The actual sizes of the three built parts are compared in Figure 1 at the same scale.

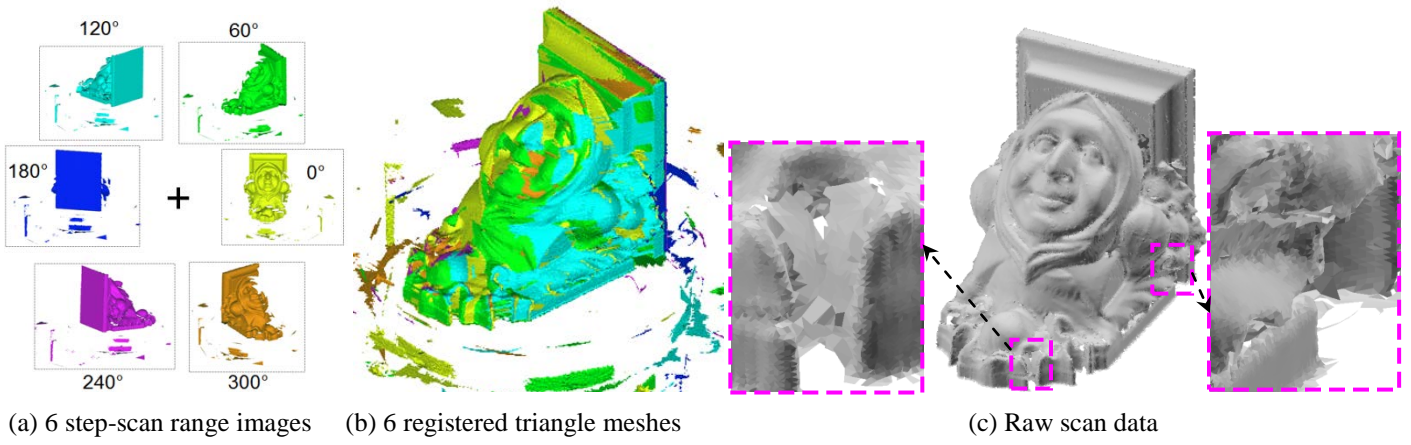


Figure 21. Raw scan data from digitizing and outlier removing.

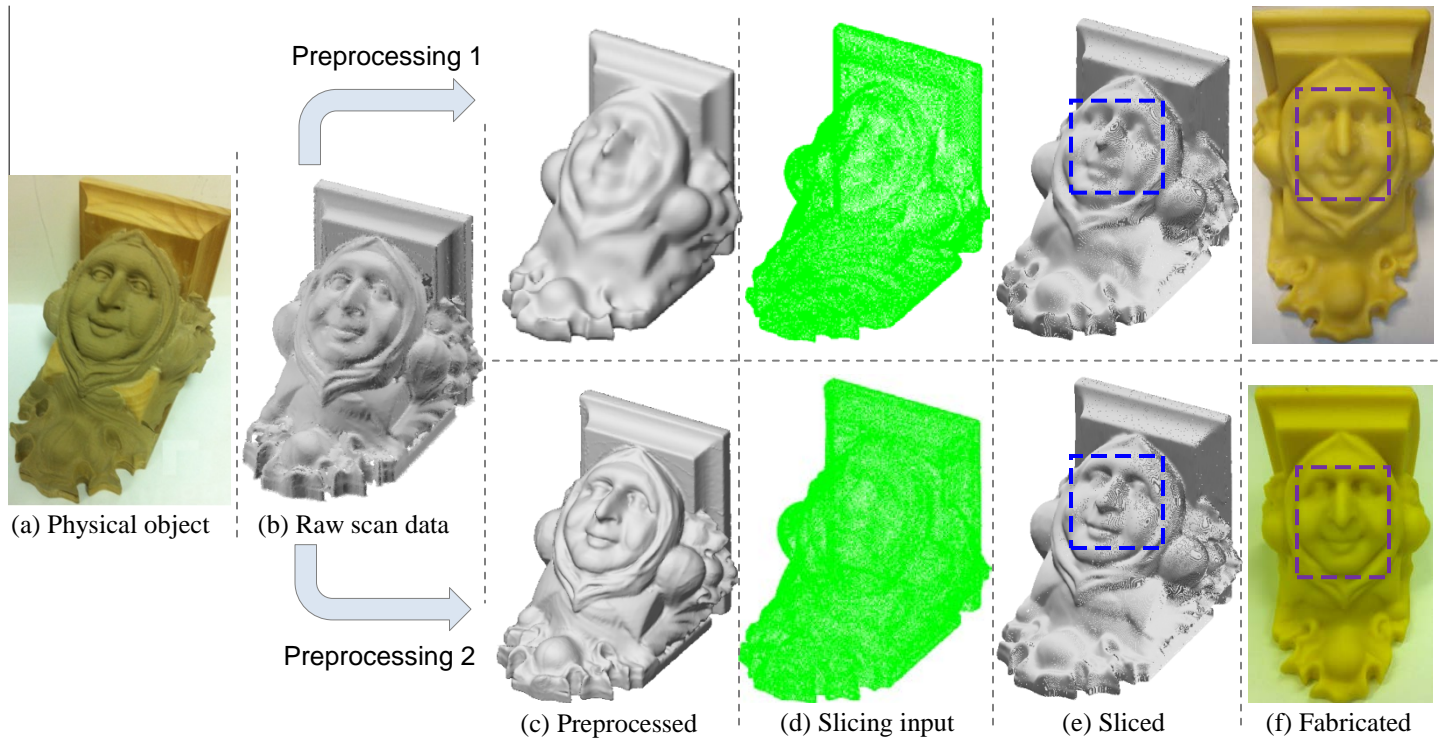


Figure 22. Geometric data flow in the intermediate steps of tele-fabrication.

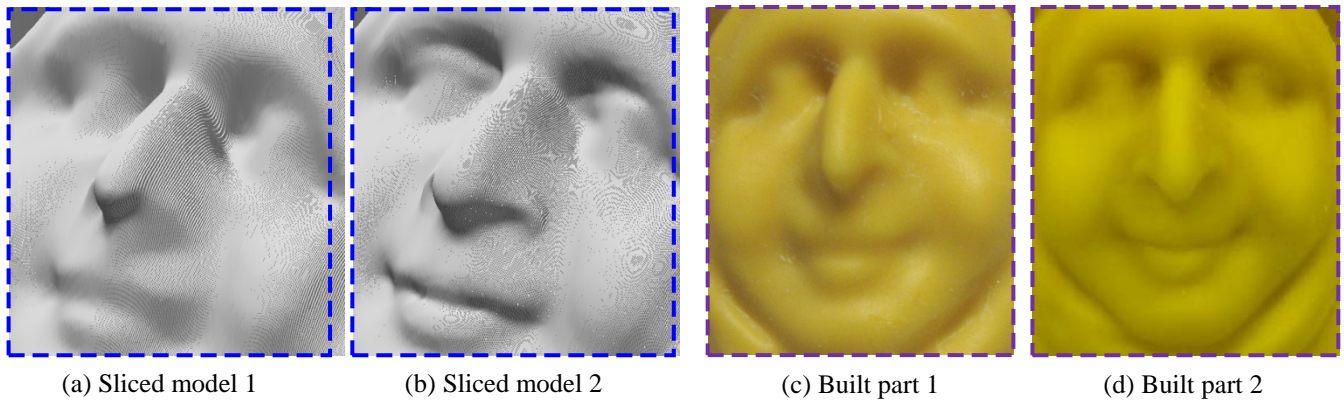


Figure 23. Face features comparison of sliced model and built part for two preprocessed slicing input.

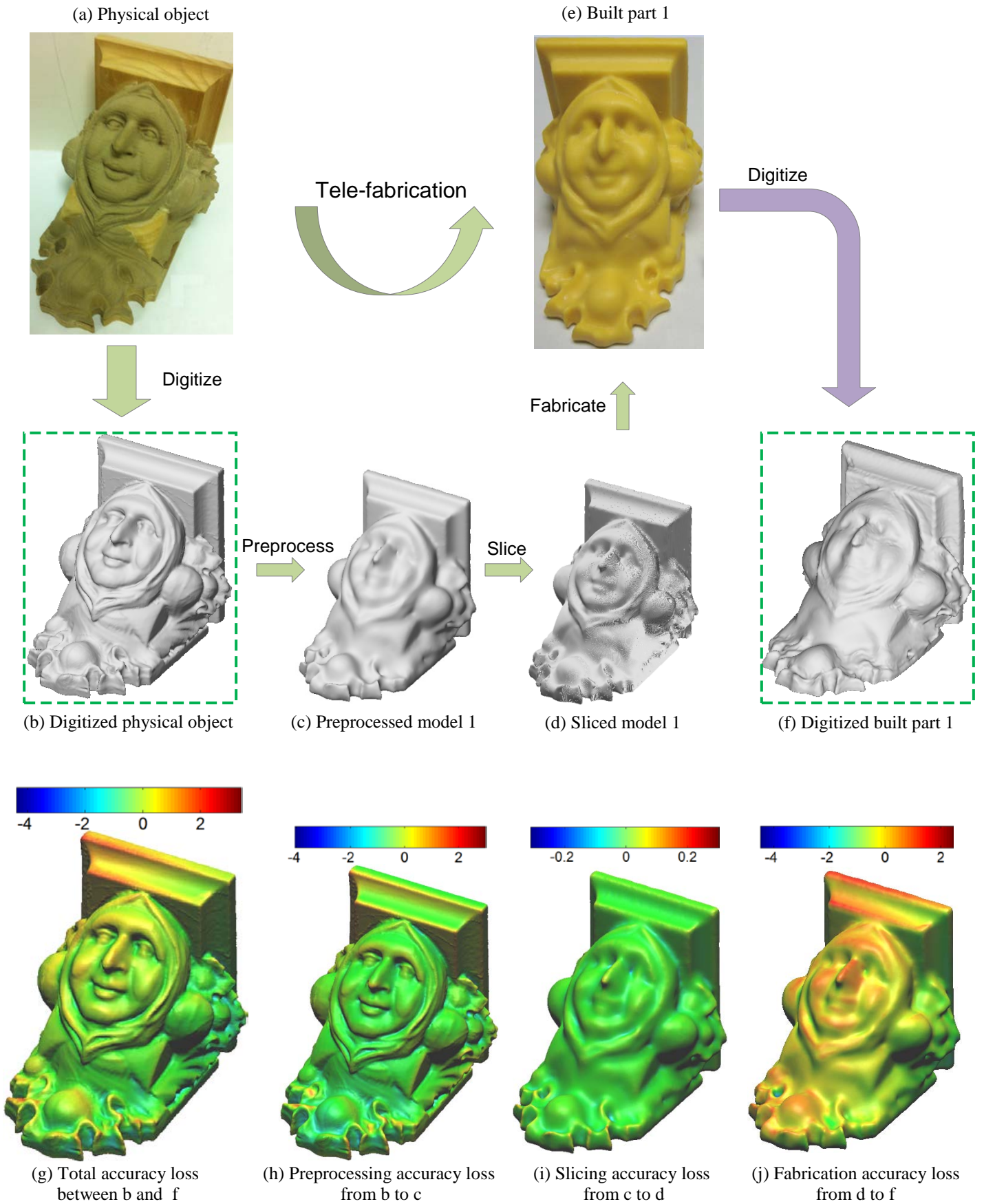


Figure 24. Comparison between physical object and built part 1

6.2 Process Analysis

We have seen that the proposed tele-fabrication process includes a series of steps: digitization (optional), point cloud slicing, support generation and mask image planning. The geometric data flow throughout all these intermediate steps is the key for the tele-fabrication; as the data flows downstream from input to output, there are inevitably dimensional accuracy loss or even feature loss incurred within each step. In this sub-section, we quantify the accuracy of the process.

As the first step in tele-fabrication, the digitizer scans the object and registers all step scans' range images to form the raw triangle mesh data. In Figure 21a, the six range images are obtained from the corresponding step scans in Figure 4. After the registration process is carried out by the built-in registration function of the scanner, the six aligned triangle meshes are shown in Figure 21b. Obviously, there exist a large amount of geometric outliers that could be easily identified and excluded. We refer to the triangle mesh data left as the "raw scan data" from the digitizer shown in Figure 21c.

However, the vertices in the raw data cannot be directly taken as the input point cloud data for slicing due to its extremely bad condition such as holes, severe noise and overlap.

As a result, we make use of common preprocessing techniques to clean and optimize raw scan data. Figure 22 demonstrates the geometric data flow during all the tele-fabrication steps: digitizing (from Figure 22 a to b), preprocessing (from Figure 22 b to c), taking vertices as point cloud (from Figure 22 c to d), point cloud slicing (from Figure 22 d to e) and fabrication (from Figure 22 e to f).

During preprocessing 1 shown in Figure 22, many features have been excessively smoothed out along with the noise. Attempting to relieve this sizable feature loss due solely to preprocessing, we have done preprocessing 2 with more carefulness on delicate features; the newly preprocessed model and the subsequent intermediate results are shown on the bottom row of Figure 22.

If we zoom in on local areas, it becomes easy for one to notice the differences between the corresponding intermediate geometric forms after preprocessing 1 and 2. In Figure 22, the local regions around the sculpture's face are zoomed in on. The feature difference between the sliced models and built parts for the two preprocessing situations are shown in Figure 23ab and Figure 23cd respectively. Preprocessing 2 has obviously more accurate features in terms of the delicate characteristics in the face region on both sliced and built parts.

In order to analyze the accuracy loss during the course of tele-fabrication through Figure 24a, b, c, d and e, we need to compare the two digitized models of the original physical part (Figure 24a) and fabricated part (Figure 24e). Take the fabrication that uses slicing input point cloud from preprocessing 1 as an example, the digitized models of them are shown in Figure 24b and Figure 24f respectively. The digitized physical object in Figure 24b is the same as the preprocess model 2 shown in the bottom row of Figure 24c.

The accuracy loss in a process is just the error between the input model and output model of the process. Quantifying

this accuracy loss requires comparing two models consisting of a collection of points. The first step in the comparison is to align the output model with the input model by use of any rigid registration algorithms, for example, the Iterative Closest Point algorithm [40]. Then the error of a point in the input model is just the Euclidean distance from the closest point in the output model to this point. Plotting all the computed errors in the input model gives the accuracy loss distribution on the input model.

The total accuracy loss result is plotted on the digitized physical object as shown in Figure 24g, and the error range is from -4.38 to +3.40. Recalling that the height of the sculpture is 100, the error range could be written as [-4.38, +3.40] % to denote relative accuracy loss. This total accuracy loss is comprised of the cumulative accuracy losses caused by the three intermediate steps: preprocessing, slicing and fabrication. And the accuracy loss of them are shown in Figure 24h, Figure 24i and Figure 24j plotted respectively on the digitized physical object, preprocessed model and the sliced model. Specifically the accuracy loss in preprocessing 1 is [-4.06, +2.99] %, in slicing [-0.31, +0.30] % and in fabrication [-4.47, +2.43] %. Table 3 lists the total and intermediate process accuracy losses in the tele-fabrication of the sculpture object, and it is observed that the point cloud slicing process barely causes any accuracy loss; the preprocessing step is responsible for a significant portion of accuracy loss; and the fabrication step also loses much accuracy, some of it may be due to the digitizing error of the fabricated object.

Table 3. Accuracy loss in the tele-fabrication of sculpture part

Accuracy loss (%)	Total	Preprocessing	Slicing	Fabrication
From	-4.38	-4.06	-0.31	-4.47
To	3.40	2.99	0.30	2.43

7 CONCLUSION AND FUTURE WORK

Reproducing physical objects in a tele-fabrication fashion where a three-dimensional physical object is scanned in one location and reproduced in another location is critical for future manufacturing. A tele-fabrication approach by integrating 3D scanning and direct digital manufacturing systems has been developed. The geometric data flow in such an integration system is a key technical challenge. In this paper, we presented a set of techniques including direct point cloud slicing, contour-based support generation, and digital mask image planning to enable the direct fabrication of scanned data. We demonstrated that it is feasible to avoid polygonal meshes in the system. Hence many laborious procedures such as manual data segmentation, surface fitting and re-discretization into STL files are bypassed. A prototyping tele-fabrication system has been developed. Based on it, a set of experimental studies have been performed. The experimental results demonstrate the effectiveness of the developed system in faithfully duplicating given physical objects, and our future work includes extending

the approach to other additive manufacturing processes. Future work would also study the data transmission of 3D data.

ACKNOWLEDGMENTS

The first author acknowledges the financial support from the National Science Foundation under grant CMMI-0927397 and CMMI-1151191. The last two authors acknowledge the financial support from the National Science Foundation under grant CMMI-0900597 and CMMI-1030347.

REFERENCES

- [1] Bailey, M.J., 1995, "Tele-manufacturing: rapid prototyping on the internet". *Computer Graphics and Applications*, **15**(6), pp.20-26.
- [2] Jiang, P.Y., and Fukuda, S., 1999, "Internet service and maintenance for rRP-oriented tele-manufacturing". *Concurrent Engineering*, **7**(3), pp. 179-189.
- [3] Lan, H., Chin, K.S., and Hong, J.,2005, "Development of a tele-service system for rp service bureaus". *Rapid Prototyping Journal*, **11**(2), pp. 98-105.
- [4] Luo, R.C., Lee, W.Z., Chou, J.H., and Leong, H.T., 1999, "Telecontrol of rapid prototyping machine via internet for automated tele-manufacturing. In robotics and Automation", *Proceedings. 1999 IEEE International Conference*, vol. 3, pp. 2203-2208.
- [5] Luo, R.C., and Tzou, J.H., 2004, "The development of an intelligent web-based rapid prototyping manufacturing system". *Automation Science and Engineering, IEEE Transactions*, **1**(1), pp. 4-13.
- [6] Marsan, A., Kumar, V., Dutta, D., and Pratt, M.J., 1998, "An assessment of data requirements and data transfer formats for layered manufacturing". *NIST, US department of Commerce*.
- [7] Liu, G.H., Wong, Y.S., Zhang, Y.F., and Loh, H.T., 2003, "Modelling cloud data for prototype manufacturing". *Journal of materials processing technology*, **38**(1), pp. 53-57.
- [8] Wu, Yifeng. Adaptive slicing of cloud data for reverse engineering and direct rapid prototyping model construction. *Journal/conference name?*, 2004.
- [9] Park, H.T., Chang, M.H., and Park, S.C., 2007, "A slicing algorithm of point cloud for rapid prototyping". *Proceedings of the 2007 summer computer simulation conference, Society for Computer Simulation International*, pp. 24.
- [10] Shin, H., Park, S., and Park, E., 2004, "Direct slicing of a point set model for rapid prototyping". *Proceedings of CAD*, Vol. 4, pp. 24-28.
- [11] Javidrad, F. and Pourmoayed, A.R., 2011, "Contour curve reconstruction from cloud data for rapid prototyping". *Robotics and Computer-Integrated Manufacturing*, **27**(2), pp. 397-404.
- [12] Yang, P., Schmidt, T., and Qian, X., 2010, "Direct Digital Design and Manufacturing from Massive Point-Cloud Data". *Computer-Aided Design & Applications*, Vol. 6, No. 5, pp. 685 - 699, 2009..
- [13] Yang, P., Li, K. and Qian, X., 2011, "Topologically enhanced slicing of MLS surfaces," *ASME Journal of Computing and Information Science in Engineering*, Vol. 11, No. 3, pp. 031003.
- [14] Yang, P., Qian, X., 2008, "Adaptive slicing of Moving Least Squares surfaces: toward direct manufacturing from point cloud data," *ASME Transactions Journal of Computing and Information Science in Engineering*, **8**(3), pp. 433-442.
- [15] Amenta, N., Kil, Y. J., 2004, "Defining point-set surfaces," *ACM Transactions on Graphics*, **23**(3), pp. 264-270.
- [16] Hart, J.C., 1998, "Morse theory for implicit surface modeling," *Mathematical Visualization, Hege H-C, Polthier K. ed., Springer-Verlag*, pp. 257-268.
- [17] Pauly, M., Keiser, R., Kobbelt, L. P., and Gross, M., 2003, "Shape modeling with point-sampled geometry," *ACM Trans. Graph.*, **22**(3), 641-650.
- [18] Yang, P., Qian, X., 2009, "Direct Boolean intersection between acquired and designed geometry," *Computer-Aided Design*, **41**(2), pp. 81-94.
- [19] Zhang, D., Yang, P., and Qian, X., 2009, "Adaptive NC Path Generation from Massive Point Data with Bounded Error," *ASME Transactions Journal of Manufacturing Science and Engineering*, **131**(1), pp. 011001-113.
- [20] Amenta, N., Kil, Y. J., 2004, "The domain of a point set surface," *Proceedings of 2004 IEEE/Eurographics Symposium on Point-based Graphics*, pp. 139-147.
- [21] Hopkinson, N., R. Hague, and P. Dickens, 2006, *Rapid Manufacturing: An Industrial Revolution for the Digital Age*, John Wiley & Sons.
- [22] Beaman, J. J., J. Barlow, D. L. Bourell, R. H. Crawford, H. L. Marcus and K. P. McAlea, 1997. *Solid Freeform Fabrication: A New Direction in Manufacturing*. Springer.
- [23] Allen, S., and D. Dutta, 1995, Determination and Evaluation of Support Structures in Layered Manufacturing. *Journal of Design and Manufacturing*, Vol. 5, pp. 153-162.
- [24] Hur, J. and K. Lee, 1996, Efficient Algorithm for Automatic Support Structure Generation in Layered Manufacturing, *ASME Computers in Engineering Conference*, DETC96/1324, Irvine, CA, August 18-22, 1996.
- [25] Kulkarni, P., A. Marsan, and D. Dutta, 2000, A Review of Process Planning Techniques in Layered Manufacturing, *Rapid Prototyping Journal*, Vol. 6, No. 1, pp. 18-35.
- [26] Chalasani, K., L. Jones, L. Roscoe, 1995, Support generation for fused deposition modeling, *Proceedings of Solid Freeform Fabrication Symposium*, Austin, Texas.
- [27] Chen, Y., C. C. L. Wang, 2011, "Uniform Offsetting of Polygonal Model based on Layered Depth-Normal Images." *Computer-aided Design*, Vol. 43, No. 1, pp. 31-46.

- [28] Wang, C. C. L., Y. Leung, Y. Chen, 2010, "Solid Modeling by Layered Depth-Normal Images on the GPU." *Computer-aided Design*, Vol. 42, No. 6, pp. 535-544.
- [29] Zhao, H., C. C. L. Wang, Y. Chen, X. Jin, 2011, "Parallel and Efficient Boolean on Polygonal Solids." *Visual Computer*, Vol. 27, No. 6-8, pp. 507-517.
- [30] Chen, Y., C. C. L. Wang, 2010, "Contouring of Structured Points with Small Features." *ASME Computers and Information in Engineering Conference*, DETC2010-29094, Montreal, Quebec, Canada, Aug. 15 ~ 18.
- [31] Schneider, P. J., D. H. Eberly, 2003, "Geometric Tools for Computer Graphics", Morgan Kaufmann.
- [32] Du, Qiang, V. Faber, and M. Gunzburger, 1999, *Centroidal Voronoi Tessellations: Applications and Algorithms*, SIAM Review, 41(4), pp. 637-676.
- [33] Du, Qiang, and D. Wang, 2006, *Recent Progress in Robust and Quality Delaunay Mesh Generation*, Journal of Computational and Applied Mathematics, 195(1).
- [34] Zhou, C., Y. Chen, R. A. Waltz, 2009, "Optimized Mask Image Projection for Solid Freeform Fabrication." *ASME Journal of Manufacturing Science and Engineering*, Vol. 131, No. 6, pp. 061004-1~12.
- [35] Zhou, C., Y. Chen, "Calibrating Large-area Mask Projection Stereolithography for Its Accuracy and Resolution Improvements." *Proceedings of Solid Freeform Fabrication Symposium*, Austin, Texas, 2009.
- [36] Xu, K., and Y. Chen, 2012, "Mask Image Planning for Deformation Control in Projection-based Stereolithography Process." *ASME Computers and Information in Engineering Conference*, DETC2012-71523, Chicago, IL, USA, Aug. 12 ~ 15, 2012.
- [37] Zhou, C., Y. Chen, Z. Yang, B. Khoshnevis, 2013, "Digital Material Fabrication Using Mask-Image-Projection-based Stereolithography." *Rapid Prototyping Journal*, Vol. 19, No. 3.
- [38] Pan, Y., C. Zhou, Y. Chen, 2012, "A Fast Mask Projection Stereolithography Process for Fabricating Digital Models in Minutes." *ASME Journal of Manufacturing Science and Engineering*, Vol. 134, No. 5, pp. 051011.
- [39] Network of Excellence Aim@Shape. (<http://www.aimatshape.net>).
- [40] Besl, Paul J., and Neil D. McKay, 1992, "A method for registration of 3-D shapes." *IEEE Transactions on pattern analysis and machine intelligence*, 14.2, 239-256.