**Solid Freeform Fabrication 2016: Proceedings of the 27th Annual International
Solid Freeform Fabrication Symposium – An Additive Manufacturing Conference
Reviewed Paper**

MASS CUSTOMIZATION: REUSE OF TOPOLOGY INFORMATION TO ACCELERATE SLICING PROCESS FOR ADDITIVE MANUFACTURING

Hang Ye[1], Chi Zhou[1]*, Wenyao Xu[2]

[1]Department of Industrial and Systems Engineering
[2]Department of Computer Science and Engineering
University at Buffalo, The State University of New York, Buffalo, NY 14260

*Corresponding author: chizhou@buffalo.edu, (716)645-4706

Abstract

Additive manufacturing (AM) can build objects with complex features with little extra effort, opening up potentials to realize mass customization. Continuous Liquid Interface Production (CLIP) prints object in a continuous fashion, leading to extremely high productivity and consequently enabling mass customization. CLIP adopts a large number of images as input, which poses a fundamental challenge in layer generation. The slicing procedure for a single customized model can take tens of minutes or even hours to complete, and the time consumption becomes more prominent in mass customization context. Motivated by the similarities among the customized products, we proposed a new slicing paradigm. It reuses topology information obtained from the template model for other customized products from the same category. The idea of topology information reuse is implemented at three levels, including self reuse, intra-model reuse, and inter-model reuse. Experimental results show that the proposed slicing paradigm can significantly reduce the time consumption on pre-fabrication computation, and ultimately fulfill mass customization enabled by AM.

Keywords: mass customization, additive manufacturing, topology, slicing, CLIP.

1. Introduction

The prevailing "mass production" is an important source of the nation's economic strength in the last century. Its characterized scale effect results in reduced cost and easiness to obtain a product, therefore it improves quality and sustainability of human life. However, due to the intense competition, any product could have countless substitutions on the market sharing similar functions. Customers are no longer satisfied with just realizing the basic function, rather they would prefer to purchase the products that can better meet their specific tastes. It is apparent that traditional mass production is not capable of handling this tremendous diversity in customer needs. Innovative practitioners are thinking about lifting their way to a new paradigm, mass customization, to meet the ever changing turbulent market environment. According to Tseng and Jiao, mass customization was defined as "the technologies and systems to deliver goods and services that meet individual customer's needs with near mass production efficiency" [1].

Over the last 30 years, additive manufacturing (AM) emerges as a new type of manufacturing process. Comparing to traditional manufacturing techniques, an important advantage of AM is that it provides “complexity for free” [2]. This property offers high flexibility and shortened product life cycle without extra penalty, thus AM has the potential to be the technical foundation for profitable customization. Several applications that utilize AM technology to enable mass production of highly customized products are shown in **Error! Reference source not found.**.

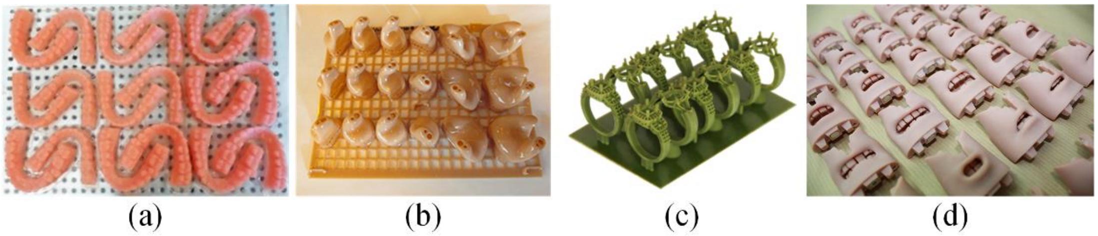

Figure 1. Mass Customization Applications based on Additive Manufacturing: (a) dental industry [3], (b) medical industry [3], (c) jewelry industry [3], (d) entertainment industry [3, 4].

AM technology holds the advantages of affordability and customizability, but the key challenge in applying it for mass customization is how to reduce the product lead time [2]. The time of building a 3D object is comprised of two components: the pre-fabrication computation and the manufacturing process. The later used to be the bottleneck of AM and comparing to this, the time spent on pre-fabrication computation seemed trivial as it usually takes only a few minutes or even seconds. However, recently Tumbleston et al. proposed a Continuous Liquid Interface Production (CLIP) approach to continuously grow an object from a vat of liquid material rather than printing them layer-by-layer [5]. This revolutionary breakthrough has proven to be 25-100 times faster than what is available on the market today. The continuous mode is an indication of using extremely small layer thickness or extremely great number of layers. Adopting this extremely thin layer increases the input size of pre-fabrication computation dramatically, and the computation cost also rockets up drastically.

In this study, we proposed a new slicing paradigm based on the observation that similarities exist among most customized products from the same category. The proposed slicing paradigm reuses topology information obtained from the template model, and this idea is implemented at three levels, including self reuse, intra-model reuse, and inter-model reuse. The remainder of the paper is organized as follows: In Sec.2, we overviewed the existing slicing method, the similarity among customized products, and the property of STL file. We also briefly described the proposed slicing paradigm in this section. In Sec.3, we proposed an improved slicing algorithm which took advantage of topological continuity. Intra-model topology information reuse was presented in Sec.4. And in Sec.5, we introduced inter-model topology information reuse which was capable of handling local feature add-on and/or removal. The conclusion and discussion were presented in Sec.6.

2. Background

2.1 Existing Slicing Algorithm

Almost all AM techniques build the 3D object in a layer by layer fashion, so they take the cross-sectional profile derived from the 3D CAD model as the input, and this information is used to direct the energy or material deposition. The common practice to obtain the cross-sectional information is intersecting the 3D model with a number of horizontal planes. Because of the adoption of STL (STereoLithography) format, which is the *de facto* file format for AM process, the intersection will be one or more simple closed polygons, and each polygon consists of a set of segments. The process to obtain these intersections is usually termed as "slicing".

The segment end point coordinate can be calculated by using Eqn.(1), where $V_1(X_1, Y_1, Z_1)$ and $V_2(X_2, Y_2, Z_2)$ are two end points of the intersected edge from the triangle and $Z = Z_k$ is the current slicing plane. The outcome of this computation is a set of unordered segments, and for most AM setups, it is necessary to know the connectivity among them for tool path planning, e.g., polygon offsetting, G-code generation etc. Closest point method [2] and marching algorithm [6] are two classic slicing algorithms. The former one calculates the coordinates of intersection points first, and retrieves the connectivity afterwards. The later one reconstructs the topology first, and then calculates the intersection coordinates according to the adjacencies.

$$
\begin{aligned}
X &= \frac{(Z_k - Z_1) \times (X_2 - X_1)}{Z_2 - Z_1} + X_1 \\
Y &= \frac{(Z_k - Z_1) \times (Y_2 - Y_1)}{Z_2 - Z_1} + Y_1 \\
Z &= Z_k
\end{aligned}
\tag{1}
$$

At a specific layer, assume n triangle facets intersects with current slicing plane. The closest point method calculates coordinates for $2n$ end points from n segments, and the time complexity is $O(n)$. However, in order to determine which point is connected to a given point, the distances between this point and all points from un-explored segments have to be computed, and this process increases the time complexity to $O(n^2)$. In contrast, marching algorithm propagates the contour from one triangle to its neighbor until getting back to the initial one, so it only calculates coordinates for n points, and the connectivity is exactly the same as the sequence of marching. Although marching algorithm seems more time-efficient than closest point method, it is necessary to know the adjacency information of the triangle facets in advance. This adjacency information does not come with the STL format model automatically, and constructing this information is also time-consuming.

2.2 Similarity among Customized Models and Mesh Property

Mass customization intends to fabricate individualized products with near mass production efficiency, thus numerous models with distinct features present a huge challenge to the pre-fabrication computation. Although some models are derived from the same template by simple transformations, such as translation, rotation, scaling etc., they still need to be processed separately. Fortunately, an important observation is that customized models from the same category usually share the characteristics of high similarity, and this high similarity exists in both geometry and topology. Figure 2 shows an example of two aligners from the same customer but at different treatment period. Those two aligner models are homeomorphic and share 99% similarity in geometry. This high similarity has already been utilized in the product design, e.g., two different models can be derived from the same template by local modification and/or deformation. However, the existing pre-fabrication paradigm treats each model independently, and each model needs to go through every single step separately, e.g., slicing, tool path planning etc.

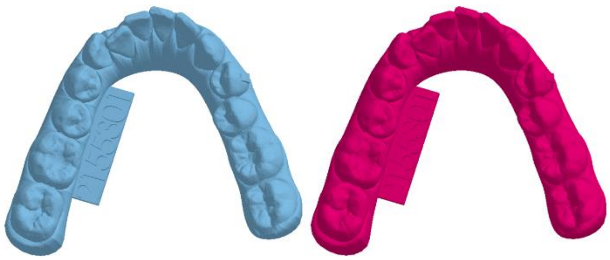

Figure 2. Two tooth aligner models share the same topology and 99% similarity in geometry. The left is for Phase 0-30 days, and the right is for Phase 31-90 days.

STL file, as the *de facto* file format for AM, is the simplest polygonal mesh. It consists of three types of mesh elements: vertices, edges and faces [7]. The information to describe the mesh property includes mesh topology and mesh geometry. The mesh topology describes the incidence relations among mesh elements [7], such as for each vertex the incident edges, and for each edge the incident triangular facets. The mesh geometry describes the position (coordinate) of each vertex. However, the connectivity/adjacency in STL model is implicit, which is not favorable for geometric algorithms. As mentioned in [8], explicit mesh connectivity will make many geometric algorithms much easier to be implemented. Inspired by this, if we can make mesh topology of a single product more predictable by taking advantages of piecewise continuity and the high similarity among customized products, the pre-fabrication computation of this product can be more efficient.

2.3 Proposed Method

In this study, we assume there exists a template mesh model, and customized models can be derived by modifying the template. We believe such assumption is valid in mass customization context,. For example, in teeth aligner industry, the first teeth model of the patient can serve as a template where the subsequent models can be incrementally adjusted based on the template. During this process, the original topological structure can be fully preserved and only the positions of the local features require minor adjustments. Because of the high similarity among customized products, this customized modification can at least preserve a large portion of the mesh connectivity from the template. In order to make mesh topology more predictable, we first explore the mesh connectivity of the template. During this exploration, piecewise continuity is utilized to reduce the redundant computation. The explored template mesh connectivity will be reused for customized products as they have the mesh topology in common. For local feature add-on/removal, we will deal with this local feature independently, thus the topology inherited from the template can keep intact and be reused in future.

3. <u>Self Topology Information Reuse</u>

In this section, the idea of topology information reuse will be incorporated in a single model by utilizing piecewise continuity in mesh connectivity. The classic closest point method is implemented in two steps: 1) calculating intersection coordinate; and 2) connecting each segment in the right order. The second step is based on the observation that the closest point to a given point is the point itself, and the flow chart of this step is shown in Figure 3 [2, 9]. It is the second step that dominates the time complexity, and at each layer the connectivity of each segment is explored separately. Because each segment is the intersection between the slicing plane and a triangular facet, two segments are connected if and only if their incident triangular facets are adjacent in space. Therefore, if both of the two adjacent triangular facets cross more than one layer (the edge shared by both facets crosses more than one layers), the connectivity of their corresponding segments will be preserved at all crossed layers.

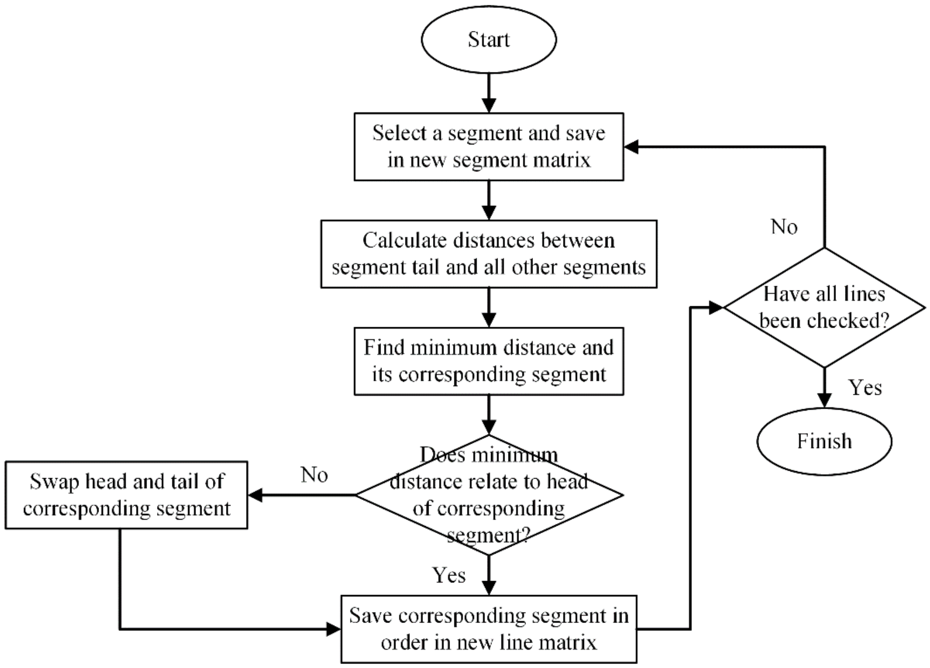

Figure 3. Flow Chart for Connecting Segments in Order

An example is shown in Figure 4. Triangular facet T_p and T_q have an edge in common, both facets cross i^{th}, $i+1^{th}$, and $i+2^{th}$ layer, and the corresponding segments $L_p^{\,i}$ and $L_q^{\,i}$, $L_p^{\,i+1}$ and $L_q^{\,i+1}$, $L_p^{\,i+2}$ and $L_q^{\,i+2}$ are connected. The closest point method does not utilize this piecewise continuity in mesh connectivity, and conducts redundant computations.

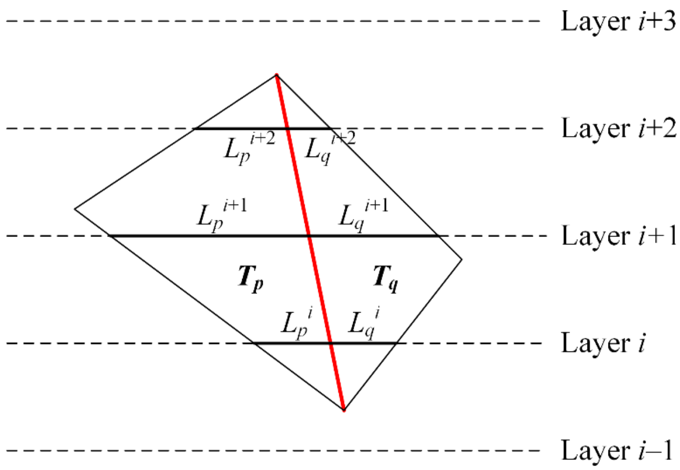

Figure 4. Piecewise Continuity in Mesh Connectivity

The idea of self topology information reuse takes advantage of this piecewise continuity in mesh connectivity, and explores the adjacency between two triangular facets only once. The explored necessary facet adjacency information is stored in a table, and in later layer if two segments are incident with the same facet pair, they will be connected automatically. As can be seen from Figure 4, if we slice the model from the bottom to the top, the adjacency between T_p and T_q is explored at layer i. At layer i+1, these two facets are still active, and we can march from L_p^{i+1} to L_q^{i+1} (or from L_q^{i+1} to L_p^{i+1}) directly without computing the distances between the tail of L_p^{i+1} (or L_q^{i+1}) and all other segments at this layer.

Since in a manifold mesh model each triangular facet has three edge-connected neighbors, a t×3 matrix is created to save the adjacency information, where t is the total number of facets from the model. Each row is corresponding to a facet, and each entry saves the neighboring facet of a given facet. For instance, $e[i, j]=k$ $(0 \leq j \leq 2)$ means the j^{th} neighbor of facet T_i is facet T_k. It is apparent that the adjacency between two facets is commutative, therefore, $e[k, j]=i$ $(0 \leq j \leq 2)$. For a given triangular facet, we save the adjacency according to the vertex sequence. Assuming three vertices from a specific facet are V_0, V_1, and V_2, the neighboring facet which is opposite to V_0 is saved at the first entry, the facet opposite to V_1 is saved at the second entry, and that is opposite to V_2 is saved at the third entry. Figure 5 shows an example.

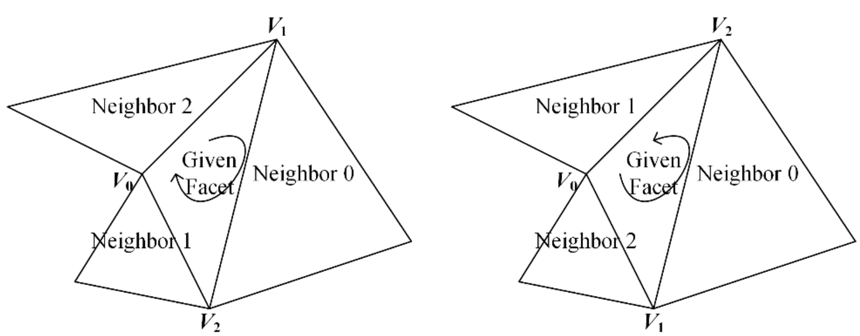

Figure 5. Neighbor Numbering of a Given Triangular Facet

Initially, all entries of the adjacency table are set as -1. At a specific layer, we start from a facet and march from one facet to another according to the known adjacency until arriving at a facet with -1 value in the entry corresponding to the neighbor in marching direction. This marching process results in a polyline instead of a segment, and the polyline consists of one or more connected segments. Figure 6 shows the flow chart of polyline computation.

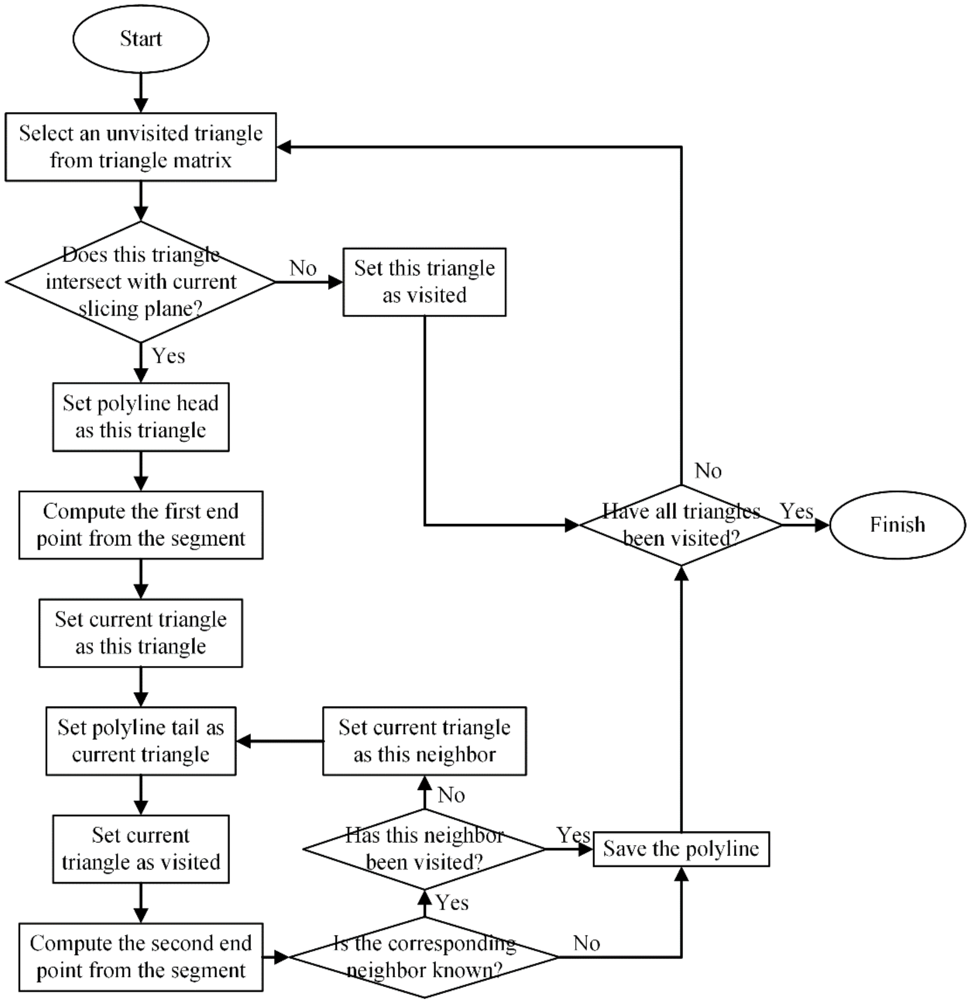

Figure 6. Flow Chart for Computing the Polylines

Once all polylines are generated, we only need to investigate the connectivity among polylines using the closest point method, and the outcome of this investigation will be used to update adjacency matrix afterwards. The flow chart of this process is shown in Figure 7. Comparing to Figure 3, there are only three differences:

1) In this process we investigate the connectivity among polylines rather than segments;
2) The distance between the head and tail from the same polyline needs to be compared as well, because it is possible that a single polyline is a closed polygonal contour;
3) The adjacency table is updated according to the minimum distance check.

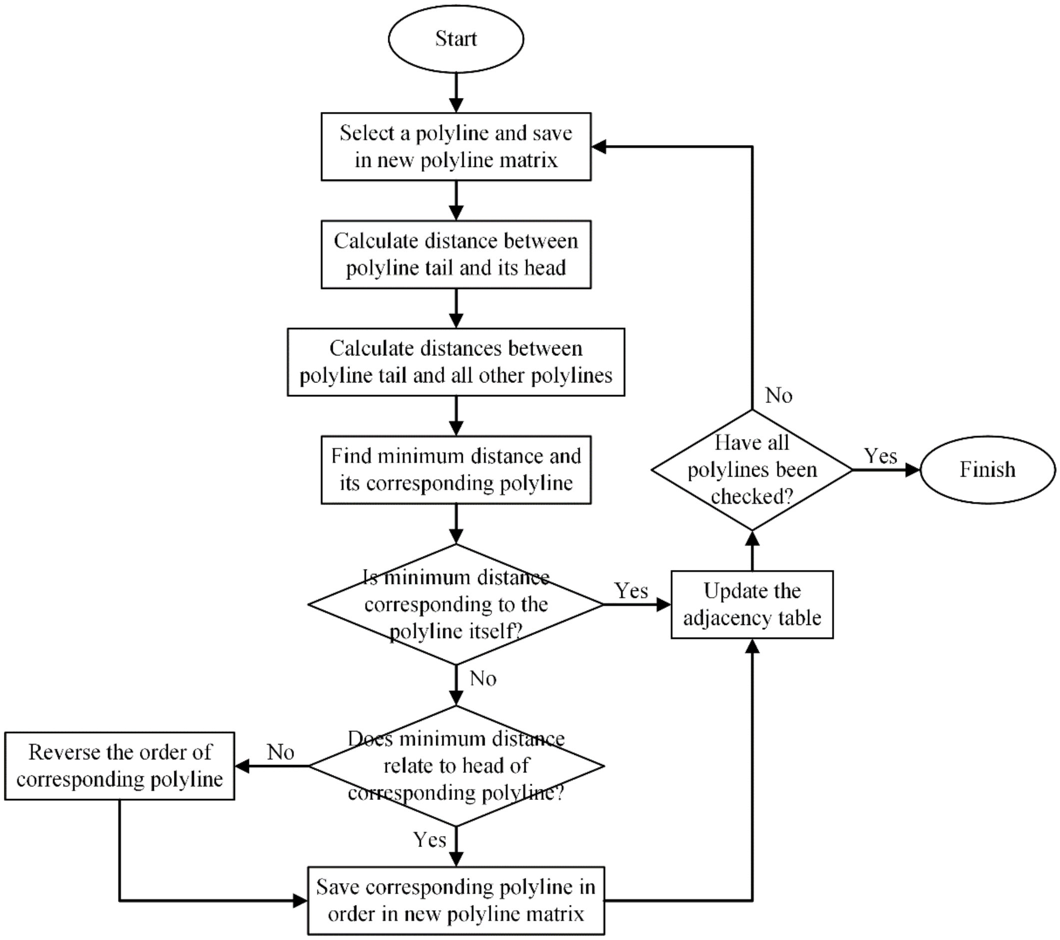

Figure 7. Flow Chart for Connecting Polylines in Order

Figure 8 shows the intersection between a model and the i^{th} slicing plane, and two contours, $L_1L_2L_6L_8L_{11}L_{10}L_3$ and $L_4L_7L_9L_5$, are included. The area enclosed by the interior contour is hollow, and the area nested between the interior contour and the exterior contour is solid. Assume that from the calculation for the previous layers, the connectivities of $L_1L_2L_6$, $L_4L_7L_9L_5$, $L_{11}L_{10}L_3$ are already known, and the incident facet of L_8 just starts to get intersected with current slicing plane. After traversing all facets, four polylines will be generated, and the details are shown in Table 1 (Assume the index of a facet is identical to the segment which is incident to this facet. i.e., Segment L_i is incident to triangular facet T_i). Once this layer is processed, the connectivity between P_1 and P_2, P_2 and P_4, P_3 and itself, P_4 and P_1 can be determined. The adjacency table needs to be upgraded accordingly and Table 2 shows the adjacency before and after this update.

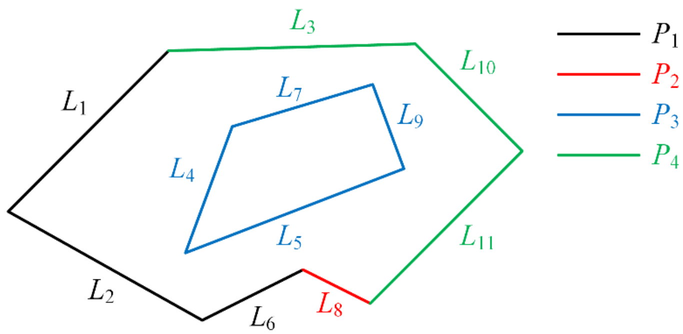

Figure 8. An Example of Self Topology Information Reuse

Table 1. Polyline Information

Polyline	Member Segment	Head	Tail
P_1	L_1, L_2, L_6	T_1	T_6
P_2	L_8	T_8	T_8
P_3	L_4, L_7, L_9, L_5	T_4	T_5
P_4	L_{11}, L_{10}, L_3	T_{11}	T_3

Table 2. Adjacency Table Update

Facet	Neighbor Before Update	Neighbor After Update
1	2	**3**, 2
2	1, 6	1, 6
3	10	10, **1**
4	7	**5**, 7
5	9	9, **4**
6	2	2, **8**
7	4, 9	4, 9
8		**6**, **11**
9	7, 5	7, 5
10	11, 3	11, 3
11	10	**8**, 10

Typical layer thickness used by existing AM processes is usually between 50 μm and 100 μm, but as a continuous AM process, CLIP can have a layer thickness as small as 1 μm [5]. After adopting such a tiny layer thickness, on one hand, the possibility that the same facet adjacency information can be reused for many layers is high; on the other hand, the effect of reusing this adjacency information is significant comparing to the closest point method. Three models, tooth aligner Figure 9(a), hearing aid Figure 9(c) and vertebral column Figure 9(e), are selected as test cases to go through both the classic closest point method and the proposed slicing method with topology reuse. These models are also typical human-centered products which have great demands for mass customization. The test environment is: 64 bit Windows 10 Pro system laptop with Intel(R) Core(TM) i7-4600U, CPU @ 2.10GHz 2.69 GHz and 8GB RAM, and the layer thickness is set as 1 μm. The results are shown in Table 3. It is apparent that by reusing the topology information, the time consumed for slicing can be dramatically reduced. It should be noted that such computational cost reduction is more prominent in mass customization context. On a typical large scale SLA machine, multiple tens of teeth aligners will be produced in each batch. The traditional slicing algorithm, according to Table 3, will take hours while the proposed self-reuse based algorithm will only take minutes to slice the models.

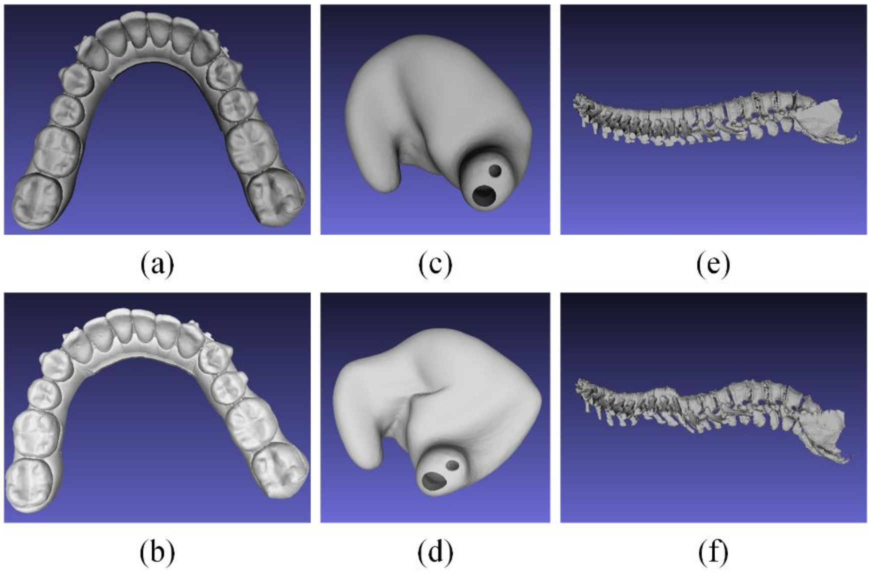

Figure 9. Test Cases: (a) original tooth aligner, (b) deformed tooth aligner, (c) original hearing aid, (d) deformed hearing aid, (e) original vertebral column, (f) deformed vertebral column.

Table 3. Time Statistic Comparison for Self Topology Reuse

Model	Size	# Layers	t_{CP}	t_{SR}
Aligner	545,250	14,484	67m8s	3m55s
Hearing Aid	327,428	8,789	5m25s	43s
Vertebral Column	548,784	10,354	55m56s	1m37s

The "Size" shows the number of facets, and "# Layers" is the number of layers. Time units are in minute (m) and second (s). "t_{CP}" and "t_{SR}" are the time consumed for closest point method and self topology reuse, respectively.

4. <u>Intra-model Topology Information Reuse</u>

In previous section, we introduced an improved slicing method, which takes advantage of the topology continuity in triangular mesh. It can be applied to a template model to obtain slices. Due to the similarity existing among customized products, we can derive other models by modifying the template. This modification can be any arbitrary change on the mesh geometry (vertex position), and the mesh topology is expected to be retained. Figure 10 shows three deformed hearing aid models, and they all share the same mesh topology with the original one.

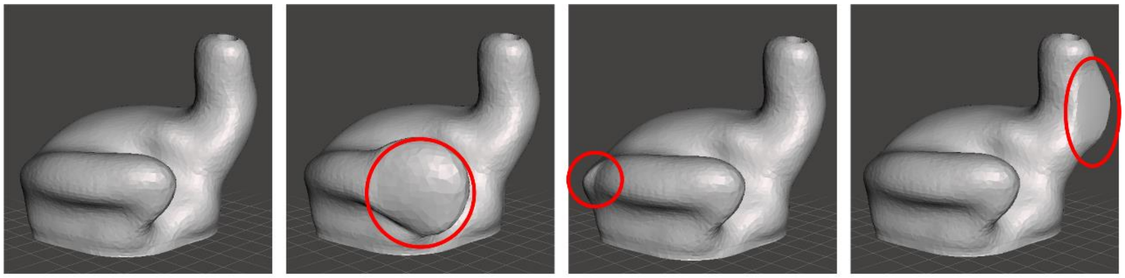

Figure 10. Deformed Models Derived from the Template

During the slicing for the template, each necessary adjacency is explored only once, and it is saved in the adjacency table. To enable the intra-model topology information reuse, this adjacency table is exported after the template is sliced. It is loaded first when we start to slice the customized model. Therefore, at least part of necessary adjacency is known for the customized models, and their slicing can be much faster compared to the model whose connectivity is totally unknown. It is possible that the exported adjacency table still has some entries whose values are -1 (which means corresponding edges of these entries do not intersect with any slicing planes during the template slicing, e.g., parallel to the slicing planes). However, in the modified model these edges may get intersected with slicing planes, and this is because of the vertex position change and/or slicing plane position change. The same strategy as the closest point method can be applied on these edges to further accomplish the adjacency table.

Three test cases, deformed tooth aligner (Figure 9(b)), deformed hearing aid (Figure 9(d)) and deformed vertebral column (Figure 9(f)), are selected to demonstrate the efficiency from intra-model topology information reuse. The test environment is the same as Sec.3, and the layer thickness is also set as 1 μm. The computational time statistics are report in Table 4.

Table 4. Time Statistic Comparison for Intra-Model Topology Reuse

Model	t_{CP}	t_{SR}	t_{IAR}
Deformed Aligner	68m6s	3m54s	1m37s
Deformed Hearing Aid	5m27s	43s	34s
Deformed Vertebral Column	54m17s	1m39s	1m14s

"t_{IAR}" is the time consumed for intra-model topology reuse slicing, and the time for loading the adjacency table is included.

5. <u>Inter-model Topology Information Reuse</u>

Sometimes modifying the template model only by changing the vertex position cannot generate desired customized feature, and local feature add-on and/or removal might be necessary. For example, a customer or a manufacturer may want to add a unique monogram on a specific customized product to differentiate this product from others. The monogram can be numbers, characters, or even QR code, and it is not easy to be achieved by changing the mesh geometry only.

Adding a feature to a mesh model can be accomplished by Boolean operations, and commercial software, such as Magics, has such functionality. The union operation merges two manifold mesh models into one, and the resultant model has to be manifold as well. During this process, the facets at or near the conjunction have to be merged, cut, or deleted. For mass customization, the impacted facets may only be a small portion of the whole model, but the original mesh topology has been compromised. If the impacted facets can be identified, we can edit the adjacency table accordingly to make it still reusable. Unfortunately, this topology change occurred during the modification is not traceable for most mesh processing software.

In order to reuse the template topology, the add-on feature is saved as a separate manifold mesh model, and it is placed at the designated position relative to the template model. In the following slicing process, the template and the add-on feature are sliced independently. Therefore, the topology information for the template can still be reused. After both parts have been sliced, the contour profiles from both parts are combined together to convert into binary images. Figure 11 is a hearing aid model with a 3D personalized monogram, and we use this model as an example to illustrate the details.

(a) (b) (c)

Figure 11. Hearing Aid with Monogram: (a) hearing aid model, (b) monogram, (c) resultant model.

For the original hearing aid model (Figure 10(a)), the adjacency table derived from template slicing still can be used. Comparing to the hearing aid model, any add-on feature usually has a smaller size in terms of number of facets. In this example, the hearing aid model has 327,428 triangles, whereas the monogram only contains 6,932 facets. Along this line, whether the connectivity of add-on feature is known is insignificant. Because the monogram is an add-on feature, the polygonal area enclosed by the contour of template and that enclosed by the contour of monogram must have some overlap, i.e., they have to at least share some segments in order to be connected to each other. This overlap results in intersection of the contours, and leads to invalid loops. Because CLIP is a mask projection based SLA process, we can skip the process of invalid loop identification and removal, which is usually required in planar polygon offsetting problem, to generate a valid image directly from contour with invalid loops.

In this paper, the concept of winding number [10] is used to generate aforementioned valid images. If we adopt the convention that the exterior contour is oriented counterclockwise (CCW) and the interior contour oriented clockwise (CW), the winding number is defined as following [10]:

Definition 1: Let $P \subset \mathbb{R}^2$ be a set of oriented polygonal contours, $q \subset \mathbb{R}^2$ be a point, and $r \subset \mathbb{R}^2$ be any ray from q to infinity that intersects no vertex of P. The winding number $\omega(r, P)$ of r with respect to P is:

$$\omega(r,P)=\sum_{e_i\in P}\Psi(r,e_i)$$

where for each edge e_i, $\Psi(r, e_i)$ is defined as follows:

$$\Psi(r,e_i)=\begin{cases}0 & \text{if } r \text{ does not intersect } e_i;\\ 1 & \text{if } e_i \text{ crosses } r \text{ in CCW as viewed from } q;\\ -1 & \text{if } e_i \text{ crosses } r \text{ in CW as viewed from } q.\end{cases}$$

In order to determine whether a single connected region is solid or hollow, positive winding rule is selected. If the winding number for this region is positive, it is classified as solid, and the pixels on the image it covers are set as foreground. Otherwise, the region is identified as hollow, and its corresponding pixels on the image are set as background. Figure 12(a) represents the contour at a specific height for the hearing aid model with monogram, and the calculated winding number is shown in Figure 12(b). Figure 12(c) is the binary image after conversion which can be used for CLIP directly. It is noteworthy that this method is also capable of handling feature removal, and the principal idea is the same as Boolean "difference". In this application, the orientation of the contour from removing feature needs to be reversed. Figure 12(e)-(h) shows an example for feature removal.

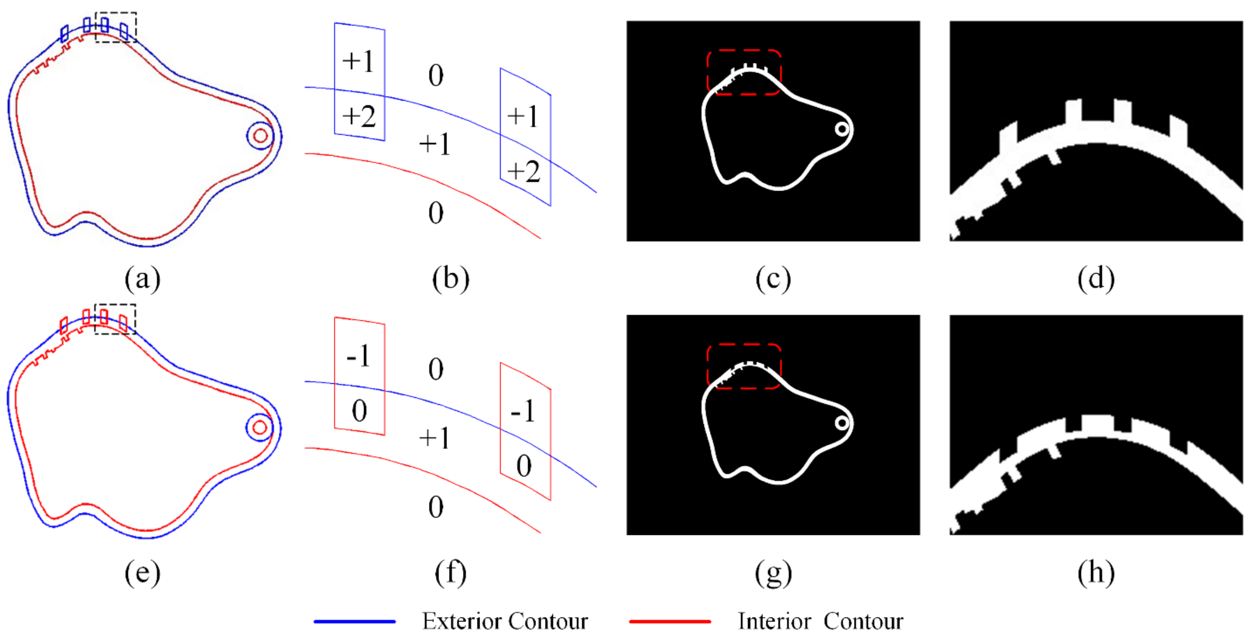

Figure 12. Hearing Aid with Monogram: (a) contour with feature add-on, (b) winding number for feature add-on, (c) image for feature add-on, (d) zoom-in for feature add-on; (e) contour with feature removal, (f) winding number for feature removal, (g) image for feature removal, (h) close-up view for feature removal.

6. <u>Conclusion and Discussion</u>

In this paper, we embraced the idea of topology reuse in the computational field for additive manufacturing. The emerging CLIP technology can dramatically reduce the fabrication time, and now the pre-fabrication computation becomes the bottleneck. By taking advantage of the high similarity existing among customized products, we proposed reusing the topology information to accelerate slicing process. This reuse was implemented in three levels, including self reuse, intra-model reuse, and inter-model reuse. Self reuse is based on the piecewise continuity in mesh connectivity, and it reuses the topology information explored from the calculation for previous layers. This improved slicing algorithm does not aim at decreasing the order of asymptotic time complexity, instead it attempts to reduce the size of input by eliminating redundant calculations. Comparing to the classic closest point method, time saving is significant, though this method consumes more storage. For a given model and a specific layer thickness, the total amount of connectivity needs to be explored is a constant. If all necessary connectivity is known, the asymptotic time complexity is linear. Intra-model reuse utilizes the topology information obtained from template slicing, and expects to attain a near linear time complexity. Customized model which is suitable for intra-model reuse can be derived from the template by applying vertex position change on all three dimensions as long as the mesh topology persists. And inter-model reuse was also proposed to solve local feature add-on and removal, which is very common in practice. It treats the add-on/removal feature independently to keep the mesh topology from template intact, thus this information can be reused. In the contour-image conversion, Boolean operations are adopted to render a valid mask image. Experimental results show that prominent time saving can be achieved by adopting the proposed method.

References

1. Tseng, M.M. and J. Jiao, *Mass Customization*, in *Handbook of Industrial Engineering Technology and Operations Management*, G. Salvendy, Editor. 2001, John Wiley & Sons: New York, NY. p. 684-687.
2. Gibson, I., D.W. Rosen, and B. Stucker, *Additive Manufacturing Technologies Rapid Prototyping to Direct Digital Manufacturing*. 2010, New York: Springer.
3. EnvisionTEC INC.; Available from: http://envisiontec.com.
4. Animations, A.; Available from: http://www.aardman.com/.
5. Tumbleston, J.R., et al., *Continuous Liquid Interface Production of 3D Objects.* Science, 2015. **347**(6228): p. 1349-1352.
6. Rock, S.J. and M.J. Wozny. *Utilizing Topological Information to Increase Scan Vector Generation Efficiency*. in *International Solid Freeform Fabrication Symposium*. 1991. Austin, TX.
7. Gotsman, C., S. Gumhold, and L. Kobbelt, *Simplification and Compression of 3D Meshes*, in *Tutorials on Multiresolution in Geometric Modelling*, A. Iske, E. Quak, and M.S. Floater, Editors. 2002, Springer: Berlin, DE. p. 319-361.
8. Shirley, P. and S. Marschner, *Fundamentals of Computer Graphics*. 3 ed. 2009, Natick, MA: A K Peters.
9. Vatani, M., et al. *An Enhanced Slicing Algorithm Using Nearest Distance Analysis for Layer Manufacturing*. in *World Academy of Science, Engineering and Technology*. 2009.
10. Chen, X. and S. Mcmains. *Polygon Offsetting by Computing Winding Numbers*. in *ASME International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*. 2005. Long Beach, CA.