

# MULTI-DIRECTION SLICING OF STL MODELS FOR ROBOTIC WIRE-FEED ADDITIVE MANUFACTURING

D. Ding, Z. Pan\*, D. Cuiuri, H. Li, N. Larkin, S. van Duin

\* School of Mechanical, Materials & Mechatronic, Faculty of Engineering and Information Science, University of Wollongong, Northfield Ave, Wollongong, NSW 2500, Australia

[zengxi@uow.edu.au](mailto:zengxi@uow.edu.au)

REVIEWED

## **Abstract**

Robotic wire-feed additive manufacturing technology is possible to directly fabricate metallic overhangs without support structures through multi-direction deposition. To automatically produce complex components with overhangs, an efficient multi-direction slicing algorithm to slice CAD models into a set of proper layers is required. This paper reports the concept and implementation of a new strategy for multi-direction slicing of CAD models represented in STL format. An input STL model is firstly decomposed into sub-volumes using a simple curvature-based volume decomposition method. Accordingly, each sub-volume is able to be built in a single direction. Then a depth-tree structure is introduced to regroup the decomposed sub-volumes and provide the slicing sequences. Consequently, sub-volumes are separately sliced along their associated appropriate build directions in sequence. The proposed multi-direction slicing strategy is shown to be simple and efficient for STL models with sharp edges.

## **Introduction**

Robotic wire-feed additive manufacturing technology has gained research interest for manufacturing metallic components with medium to large size due to its benefits of the high deposition rate and the large build envelope [1-3]. Most of the commercial AM processes involve slicing 3D CAD models into a set of 2.5D layers with a constant or adaptive thickness perpendicular to the build-up direction (usually designated as the Z+ direction). However, to fabricate parts with complex shapes using wire-feed additive manufacturing technologies, processes based on a uni-directional slicing are generally limited by the need for support structures to deposit overhangs. For metal components, the supports are normally deposited using the same material. As a consequence, the deposition of supports results in the wastage of materials and the removal of these supports requires costly post-processing.

Robotic wire-feed additive manufacturing, which mounts a deposition nozzle on a multi-axis robot arm, is able to deposit materials along multiple directions. The application of robotic system together with a rotary worktable could eliminate or significantly decrease the usage of supports for complex components through multi-direction deposition. Such a multi-direction system, as shown in Fig.1, furthers the capability of layered manufacturing by reducing the need for supports.

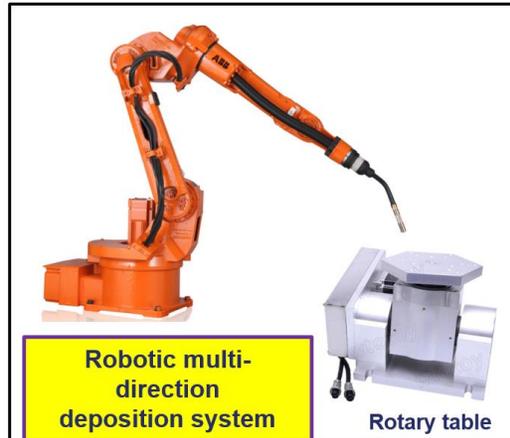


Fig.1 A schematic diagram of a robotic multi-direction deposition system

A key challenge in multi-direction additive manufacturing is to develop robust and efficient algorithms capable of automatically slicing any 3D model into a layers which satisfy support-less and collision-free layered deposition. Some existing multi-direction slicing strategies are reviewed here.

*Silhouette edges projection* [4]: This strategy firstly identifies the unbuildable surface features of a model by projecting silhouette edges along the user defined original build direction. Then the part is decomposed into buildable and unbuildable sub-volumes using the silhouette-edge based method. For the unbuildable sub-volume, a new suitable build direction is determined using the Gauss and Visibility maps. With the new build direction, the unbuildable sub-volume is further decomposed through repeating the same projection procedures. This projection method is recursively used to decompose the sub-volume until the generated sub-volume is buildable along one direction. The framework for the multi-direction slicing and some essential problems have been addressed and discussed by the authors. However, the implementation of the strategy could be complicated and computationally expensive for complex components with inner cavities.

*Transition wall* [5]: The key idea of this strategy is to identify the overhang layers by computing the difference between the current layer and the previous layer. Then, to build an overhang structure, the machine is turned 90 degree to start depositing a transition, namely thin wall. After the deposition of the first few layers, the wall is finished and the subsequent overhang structures can be deposited in the vertical direction again. Although this strategy is simple, it is only suitable for a subset of part geometries. In some cases, the deposition of the transition wall is difficult or impossible to implement due to deposition nozzle collision.

*Centroid axis extraction* [6]: The first step in this strategy is to extract the centroid axis of the model which provides a global perspective on the geometry, allowing the slicing procedure to be conducted on an optimal sequence. Through analysing the topological information from the centroid axis, the splitting surface is identified and the subsequent decomposition operation is conducted. For each sub-component obtained from decomposition, multi-axis slicing is performed and the collision free slicing sequence is finally generated. The centroid axis extraction method decomposes the component by detecting the change of pre-sliced layers, making the geometry

analysis process easier. However, in some cases it will be difficult to decompose components efficiently as required since the centroid axis does not always indicate the change of the geometry.

Other multi-direction slicing methods have been proposed, such as normal marching algorithm [7], offset slicing [8], skeleton method [9], modular boundary decomposition [10], and adaptive slicing [11], with each method is only suitable for a subset of part geometries. In addition, these methods are not sufficient for processing parts with holes and depression features.

This study presents a simple and efficient multi-direction algorithm for STL models. Algorithm overview is described firstly, and followed by the detailed algorithm. Consequently, a case study of the algorithm is provided. The paper ends with conclusions and future works.

### Algorithm overview

The input of the algorithm is a standard STL, which is most commonly used to represent 3D CAD models in AM process planning. A section of a STL file and its 3D model are shown in Fig.2. In an ASCII STL file, the model is represented using triangular facets, which is described by the X-, Y-, and Z-coordinates of the three vertices and a unit vector to indicate the normal direction that point outside of the facet.

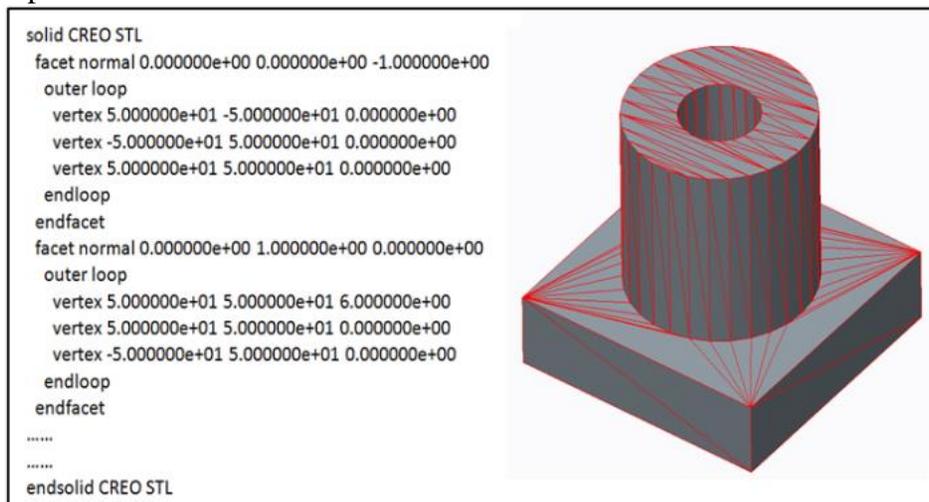


Fig.2 Example of a STL file format and its model

The algorithm proposed in this study for generating multi-direction slices mainly includes two modules, namely Volume decomposition-regrouping and Build direction identification.

Volume decomposition-regrouping module firstly decomposes a complex part into a variety of convex sub-volumes through progressively using a curvature-based decomposition method. A variety of sub-volumes are obtained from the volume decomposition, and the region from which the sub-volume has been split is stored as the topology information. Then a depth-tree structure based on the topology information is constructed to regroup the sub-volumes and provide the order of sequences for the latter slicing module.

Build direction identification module is developed to identify the build direction for each sub-volume to guide the subsequent slicing procedure. For convenience, Gauss Map is used to calculate the optimal build directions for sub-volumes [12]. Finally, each sub-volume is sliced along its build direction.

Fig.3 shows an example of multi-direction slicing. The component is decomposed into four sub-volumes (a, b, c, and d) through progressively decomposition process. Sub-volumes are regrouped together through depth-tree structure and sliced along their optimal build directions accordingly. With the multi-direction deposition method proposed in this study, the part can be fabricated without additional supports. The original orientation of the part with respect to its base is considered as a user defined input and not to be changed in the proposed algorithm.

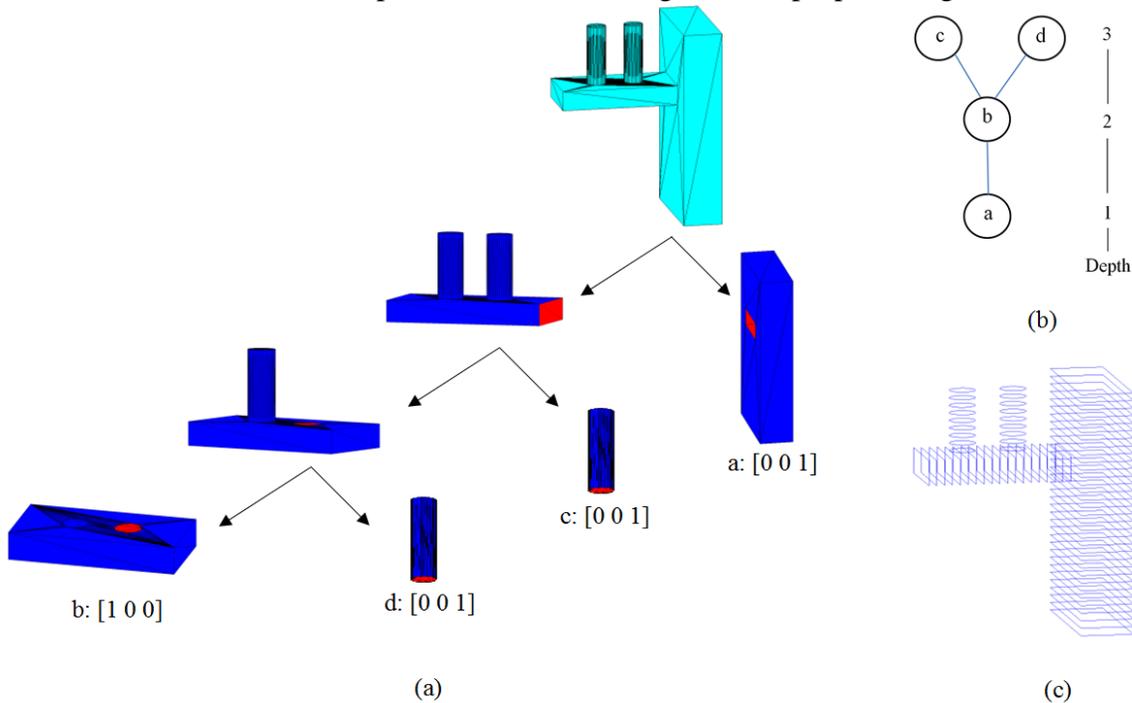


Fig.3 (a) a progressive decomposition of an example into sub-volumes with their own build directions; (b) sub-volume regrouping with depth-tree structure; and (c) model slicing along multiple build directions

### Multi-direction slicing algorithm

#### Volume decomposition

In this section, the algorithm for progressive decomposition of the volume is summarized with the aid of the flowchart shown in Fig.4. The input for this task is the simplified STL model (triangle mesh) in which all the holes been filled. The volume is decomposed into several sub-volumes by splitting the triangular facets along the concave loops. Consequently, a set of sub-volumes free of any concave loops is obtained.

Concave\_loop\_set extraction: Generate the boundary loops of the input mesh and check the convexity of the edges for each boundary loops. Then concave loop sets are extracted as all the

edges forming the loop are concave edges. The process of decomposing an input mesh can be considered to be the process of progressively eliminating concave loops. For a complex model with  $N_c$  concave loops, the algorithm outputs  $N_c+1$  sub-volumes. Algorithms for finding the boundary loops and calculating the convexity of edges in STL triangle mesh could be found elsewhere [13].

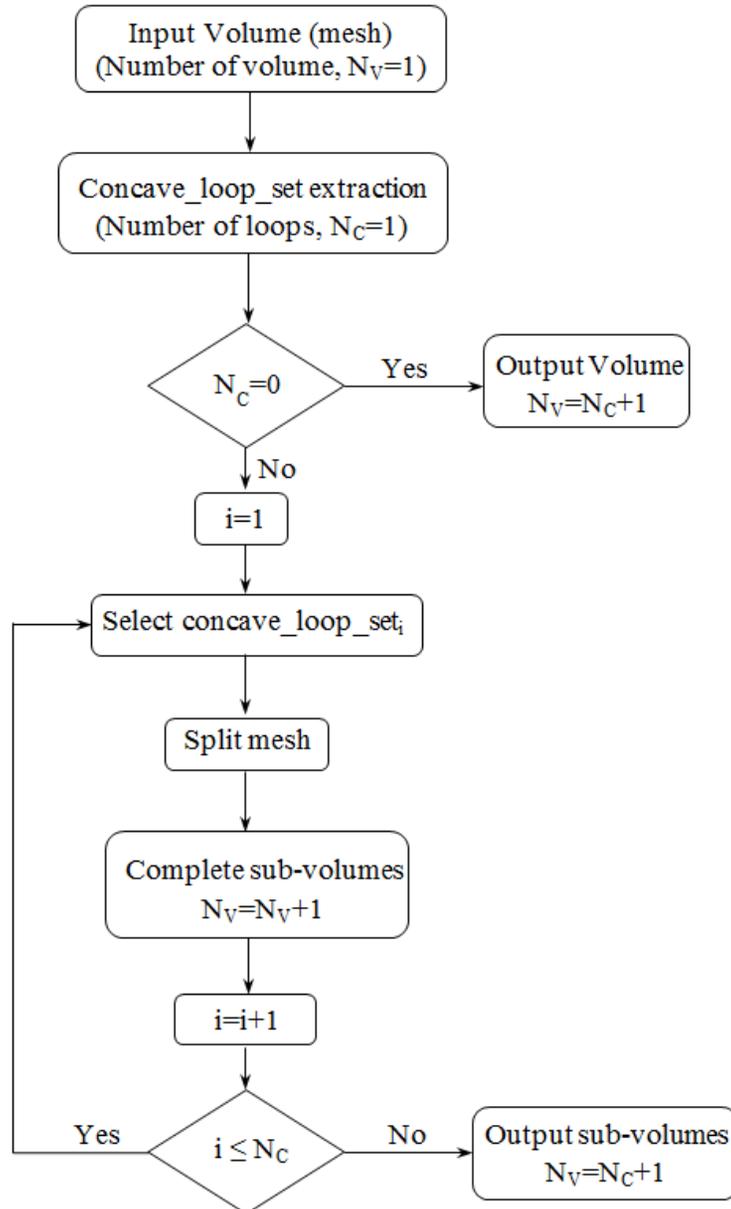


Fig.4 The algorithm for progressive decomposition

**Split mesh:** When the `concave_loop_seti` is selected as the split loop, an effective algorithm is required to divide the facet set into two facet sets. An advancing front search technique is developed to split the mesh into two new facet sets. This method consists of the following steps:

*Step 1.* Initialize the front using the selected `concave_loop_seti`. As shown in Fig.5,  $e_i$  and  $e_{i+1}$  are the front edges.

*Step 2.* Search the related triangle facet for each front edge. In Fig.5a and 5b,  $j$  and  $k$  are the related triangle facets of  $e_i$  and  $e_{i+1}$  respectively; while in Fig.5c,  $e_i$  and  $e_{i+1}$  share the same facet  $j$ .

*Step 3.* Add triangles to a new facet set and update the front with the following three rules, as shown by the three cases in Fig.5. Rule (a): If the related triangle facets are independent (such as facet  $j$  and  $k$  in Fig.5a), add the related triangle facets into the new facet set, and update the front edge using the other edges of the related triangle facets (dashed red lines with arrow). Rule (b): If the related triangle facets share one edge (such as facet  $j$  and  $k$  in Fig.5b), add the related triangle facets into the new facet set, but update the front using the other edges except the shared edge of the related triangle facets. Rule (c): If the related triangle facets are overlapped (such as facet  $j$  in Fig.5c), add the triangle facet into the new facet set, and update the front using the rest edge of the related triangle facet.

*Step 4.* Repeat Steps 2 through 3 until the front is empty.

*Step 5.* The triangle facets, which haven't been detected from Step 2 through 4, are collected to another new facet set.

This method can always split the input mesh into two facet sets, whatever its shape, and guarantee the robustness of the proposed algorithm. Note that the generated two facet sets are two open half volumes.

Complete sub-volume step is used to complete the two sub-volumes from the two open facet sets generated by applying the above procedure. The selected concave\_loop\_set<sub>i</sub> is considered as the hole loop, and the hole-filling algorithm [14] is used again to close the two open sub-volumes. The filled hole regions are recorded as the feature regions of the sub-volume, which is useful for the subsequent grouping process. With the progressive decomposition, the component is decomposed into a variety of sub-volumes.

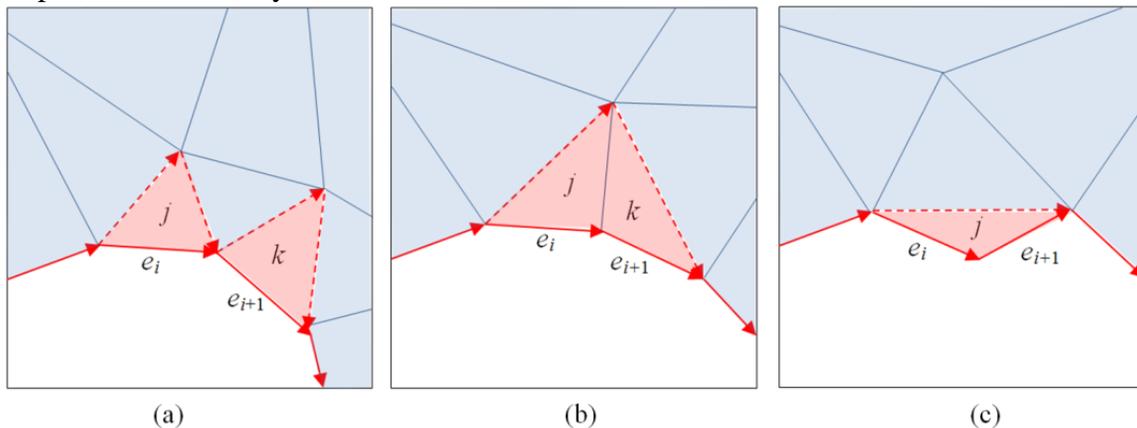


Fig.5 Rules for adding triangles and updating the front

### Sub-volume regrouping

Since the obtained sub-volumes are disordered, topology information between each sub-volume and its feature regions is constructed for the grouping process. The feature region represents the connection area between two sub-volumes and is important information for ordering sub-volumes. As the orientation is the user defined input, the base region is the bottom of the input component. Generally, only one of the sub-volumes has the base region. Starting from the base

region on basis of the characteristics of AM, all sub-volumes are connected by searching the shared feature regions.

The sub-volumes then are ordered and grouped with the depth-tree structure starting from the sub-volume that has the base region. As schematically shown in Fig.6, sub-volumes are stored as  $\{i, j\}$ , where  $i$  represents the depth of the sub-volume. As the orientation is the user defined input, the base region is the bottom of the input component, therefore sub-volume  $\{1, 1\}$  is easily identified from the sub-volumes. Usually, there is only one sub-volume that has the depth of 1. Then, sub-volumes whose base regions are connected with the sub-volume  $\{1, 1\}$ , have the depth of 2. Finally, all the sub-volumes are stored within the depth-tree structure.

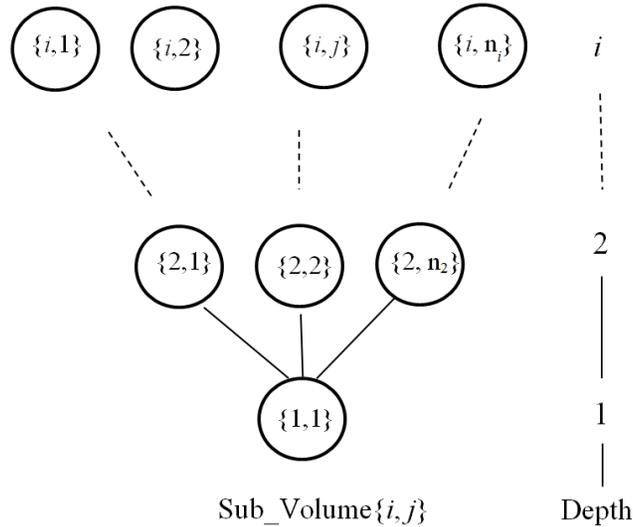


Fig.6 Sub-volumes grouping with the depth-tree structure

### Build directions determination

The optimal build direction of each sub-volume must be determined to guide the subsequent slicing procedure. Fig.7 shows two sub-volumes  $V_1$  and  $V_2$ . Assume the build direction of  $V_1$  has already been determined, but the optimal build direction of  $V_2$  is not yet determined. All the regions except the base region of  $V_2$  are used to find the optimal build direction  $B_{opt}$ . The normal vectors of the facets in these regions are recorded in a set  $n$ . The problem is stated as follows:

*Given a set of normal vectors  $n = \{n_1, \dots, n_i, \dots, n_n\}$ . Find the optimal vector  $B_{opt}$  that produces the smallest maximum angle between this vector and any vector  $n_i$  within the set.*

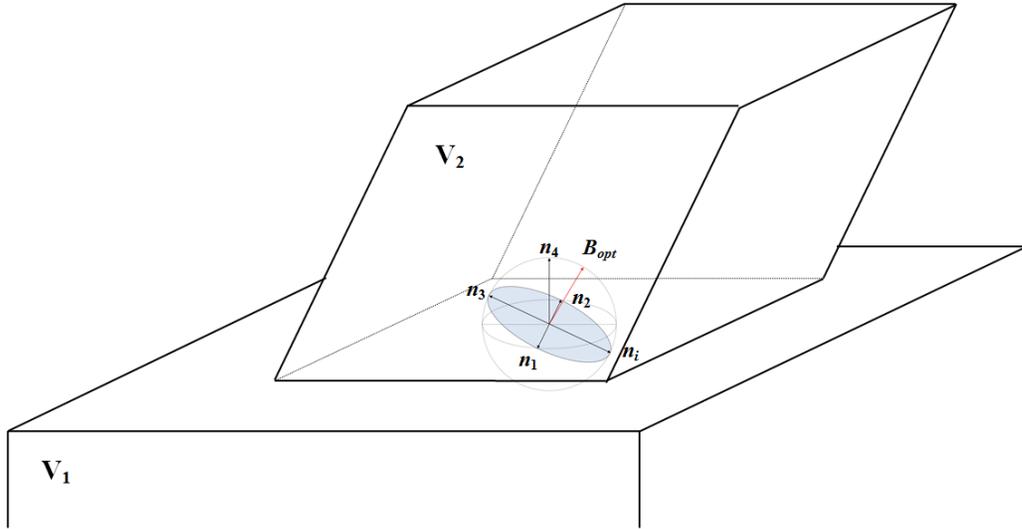


Fig.8 The optimal build direction for sub-volume

Gauss Map [15] is used, which collects all the normal on a unit sphere. A set of normal vectors  $n$  can be represented in a set of points on the surface of a unit sphere. For this set of points, a spherical crown  $C$  with a minimum-radius bottom surface that contains all the points can be found on the sphere surface [7]. The unit vector from the centre of the sphere to the centre of the bottom circle of the spherical crown  $C$  is the desired optimal build direction. The optimisation problem is thereby simplified to searching for the minimum spherical crown which contains all the points of the normal vectors  $n$ . Algorithms to find the minimum spherical crown basing on the algorithm for the smallest enclosing discs problem were developed [7], [16].

### Slicing

According to the depth of each sub-volume, the slices of each sub-volume along its optimal build direction can be generated one by one. A tolerant slicing algorithms for STL models along the  $Z+$   $[0\ 0\ 1]$  direction has been widely reported [17]. In this study, matrix transformations are used to apply these basic tolerant slicing algorithms to each sub-volume in the model along the optimal build direction.

Assume the build direction of a sub-volume is  $B_i = [B_{ix}\ B_{iy}\ B_{iz}]$ . Through the rotation  $\theta$  and  $\varphi$  about  $x$  and  $y$  respectively, the  $z$  axis is made parallel to the build direction. One can obtain the following equation:

$$\begin{pmatrix} B_{ix} \\ B_{iy} \\ B_{iz} \end{pmatrix} = M^{-1} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} \cos \varphi & \sin \varphi & 0 \\ -\cos \theta \sin \varphi & \cos \theta \cos \varphi & \sin \theta \\ \sin \theta \sin \varphi & -\sin \theta \cos \varphi & \cos \theta \end{pmatrix}^{-1} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}.$$

One can obtain the value of rotation  $\theta$  and  $\varphi$ , and consequently the transfer matrix  $\mathbf{M}$  and the inverse matrix  $\mathbf{M}^{-1}$ . Transfer all the facets vertices and facet normal of the sub-volume  $i$ , then slice along  $Z+$  direction using the existing algorithms [17]. Finally, transfer the slice layers back to the original orientation of the sub-volume. With this process, each sub-volume is sliced along its

build direction individually. Note that this proposed process can also be extended to utilise the adaptive direction and thickness slicing algorithm [7] which could generate non-uniform thickness layers matching the surface curvature of the sub-volume appropriately.

At this point, filled holes need to be reconsidered. Each hole is uniquely associated to its related sub-volume. If the hole is parallel to the build direction of its sub-volume, it is not require filling. Therefore, the slice of the hole is conducted, otherwise, deposition of the hole is needed for support purposes and the volume of the hole is extracted from the final part by post-process milling.

### Case study

The strategies presented in this study were implemented by programs written in Matlab. Besides the slicing results shown in Fig.3, one more example and slicing results is presented here. As shown in Fig.9a, the component has many holes which are difficult to handle when using previously-developed multi-direction slicing algorithms from the literature.

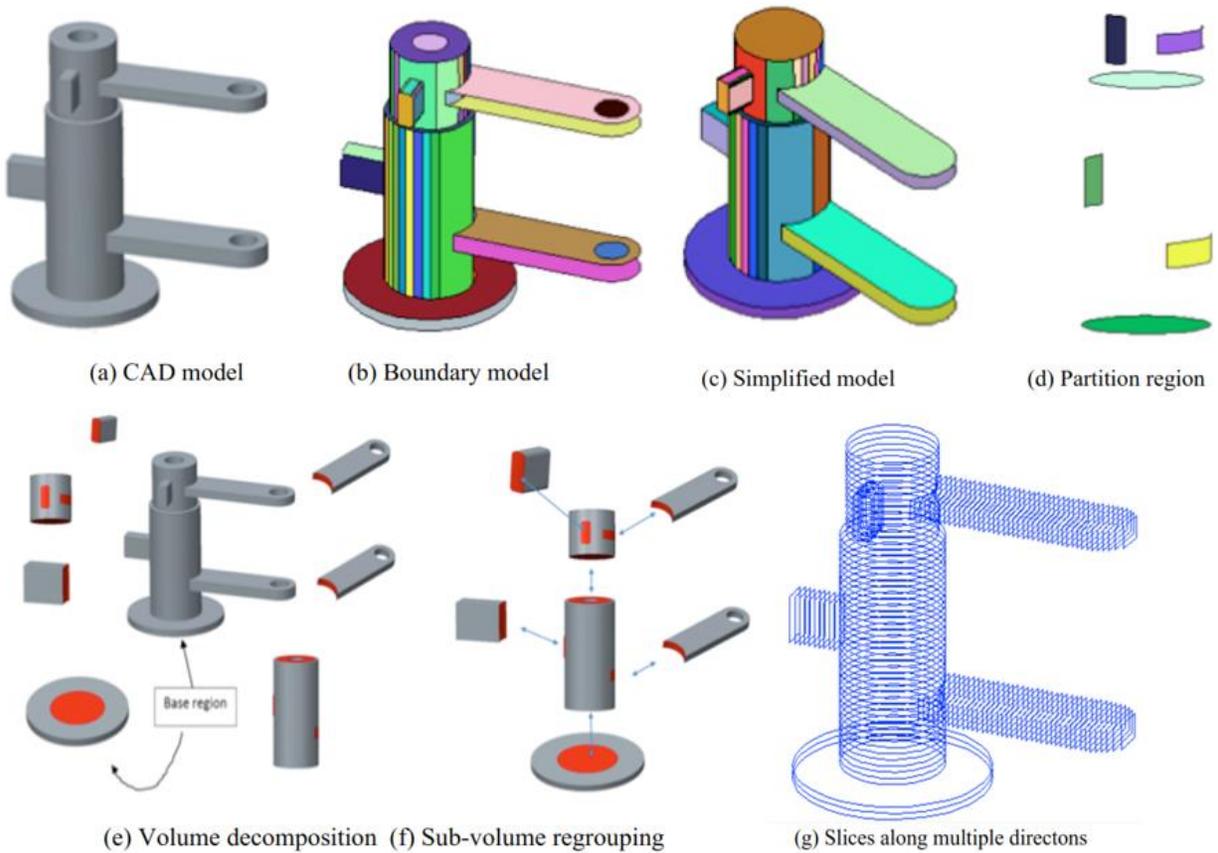


Fig.9 Illustrations of the proposed multi-direction slicing algorithms

The boundaries of the CAD mode are generated and the model is segmented into several patch region as shown in Fig.9b. Two holes are detected and filled as the simplified model shown in Fig.9c. Concave loops are extracted and their enveloped regions called partition region as shown in Fig.9d. The model then decomposed into a set of disordered sub-volumes with each sub-volume can be built along single direction, as shown in Fig.9e. Consequently, these sub-volumes are

regrouped together (see Fig.9f) according to the layer-by-layer basis of additive manufacturing. Finally, the CAD model is sliced in multiple directions, as shown in Fig.9g.

The implementation indicates that the proposed multi-direction slicing strategy is simple and efficient for CAD models with sharp edges. Simplification on hole-features could significantly enhance the capability of the proposed slicing algorithms. As the reader will note, both parts in Fig.3 and Fig.9 can indeed be fabricated by conventional (unidirectional) deposition. However, support structures will be required.

### **Conclusions and future works**

This paper presents a new strategy for multi-direction slicing of CAD models represented in STL format for wire-feed additive manufacturing (AM). Differing from the previously devised methodologies in the literature, model simplification and the concept of decomposition-regrouping are introduced in the workflow. The algorithm is verified to be an efficient and flexible method for various types of CAD models in AM application. Since each step of the process maintains the integrity of STL format (triangle mesh), then intermediate results from each of the modules (such as model simplification, volume decomposition, and holes-extraction) can produce standard STL files to be used at other downstream CAD/CAM applications that are not limited to AM.

It should be noted that in this paper, only the closed concave loops are adopted to decompose the components. A complete decomposition module, which could process components with any complex geometry, collision problems, as well as issues related to the orientation of the components for multi-direction AM, will be our future research interest.

### **Acknowledgements**

The authors would like to thank Professor J. Norrish for his expertise in the welding field. This work is supported in part by the State Scholarship Fund of the China Scholarship Council (No. 2011684067).

### **References**

- [1] J. Ding, P. Colegrove, J. Mehnen, S. Ganguly, P.M. Sequeira Almeida, F. Wang, and S. Williams: *Theomo-mechanical analysis of wire and arc additive layer manufacturing*, Computational Materials Science, vol. 50, pp. 3315-3322, July. 2011.
- [2] D. Ding, Z. Pan, D. Cuiuri, and H. Li: *Wire-feed additive manufacturing of metal components: technologies, developments and future interests*, The International Journal of Advanced Manufacturing Technology, vol., pp., May. 2015.
- [3] N. Kapustka and I.D. Harris: *Exploring arc welding for additive manufacturing of titanium parts*, Welding Journal, pp. 32-35, March. 2014.
- [4] P. Singh and D. Dutta: *Multi-direction slicing for layered manufacturing*, Journal of Computing and Information Science in Engineering, vol. 1, p. 129, 2001.
- [5] Y. Yang, et al.: *Multi-orientational deposition to minimize support in the layered manufacturing process*, Journal of Manufacturing Systems, vol. 22, pp. 116-129, 2003.
- [6] J. Ruan, et al.: *Automated Slicing for a Multi-Axis Metal Deposition System (Preprint)*, DTIC Document2006.

- [7] J. Zhang and F. Liou: *Adaptive slicing for a multi-axis laser aided manufacturing process*, Journal of Mechanical Design, vol. 126, p. 254, 2004.
- [8] P. Singh and D. Dutta: *Offset slices for multidirection layered deposition*, Journal of Manufacturing Science and Engineering, vol. 130, p. 11011, 2008.
- [9] R. Dwivedi and R. Kovacevic: *Process planning for multi-directional laser-based direct metal deposition*, Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, vol. 219, pp. 695-707, 2005.
- [10] L. Ren, et al.: *Process planning strategies for solid freeform fabrication of metal parts*, Journal of Manufacturing Systems, vol. 27, pp. 158-165, 2008.
- [11] X. Wang, H. Zhang, G. Wang, L. Wu: *Adaptive slicing for multi-axis hybrid plasma deposition and milling*, The 25<sup>th</sup> Annual SFF Symposium 2014, 1277-1287.
- [12] J. Gan, et al.: *Spherical maps: their construction, properties, and approximation*, Journal of Mechanical Design, vol. 116, pp. 357-363, 1994.
- [13] J. Hao, F. Liang, E.W. Robert: *An efficient curvature-based partitioning of large-scale STL models*, Rapid Prototyping Journal, 17/2, pp. 116-127, 2011.
- [14] W. Zhao, et al.: *A robust hole-filling algorithm for triangular mesh*, The Visual Computer, vol. 23, pp. 987-997, 2007.
- [15] T. C. Woo: *Visibility maps and spherical algorithms*, Computer-Aided Design, vol. 26, pp. 6-16, 1994.
- [16] M. De Berg, et al., Computational geometry: Springer, 2000.
- [17] S. Choi and K. Kwok: *A tolerant slicing algorithm for layered manufacturing*, Rapid Prototyping Journal, vol. 8, pp. 161-179, 2002.