

# Machine Path Generation Using Direct Slicing from Design-by-Feature Solid Model for Rapid Prototyping

Mohammad T. Hayasi<sup>a</sup>, Bahram Asiabanpour<sup>b,†</sup>

<sup>a</sup>Department of Mechanical and Manufacturing Engineering, Faculty of Engineering, Universiti Putra Malaysia, 43000UPM Serdang, Malaysia

<sup>b</sup>Bruce and Gloria Ingram School of Engineering, Texas State University- San Marcos, San Marcos, TX 78666, USA

Reviewed, accepted September 10, 2008

## Abstract

Because of the differences in the rapid prototyping (RP) processes, there is no standard machine path code for them. For each RP process, based on its characteristics and requirements, the required data are extracted from the CAD model and converted into proper format for the RP machine. Slicing a CAD model through intersecting the model with the XY-plane at each Z increment is a well-known method of path generation. Slicing a CAD model is currently conducted through Stereolithography (STL) file slicing, direct slicing, and additive direct slicing. In this paper, a direct slicing approach inside a design-by-feature solid modeler is proposed. Autodesk Inventor solid modeler, as a design-by-feature solid modeler is used for 3D solid modeling. The proposed system is implemented by Visual Basic codes inside Inventor. In this approach, first protrusion and subtractive features that form a model are extracted. Then, the intersection of each feature and the XY plane is identified. Then, internal and external loops are found. Finally a continuous path in required format is generated. Depending on the specific RP process requirements, additional features such as internal or external hatch can be added to the machine path.

*Keywords:* Machine path generation, Direct slicing, Design-by-feature, Feature recognition.

## 1. Introduction

Since the introduction of the stereolithography (SLA) process as the first rapid prototyping (RP) process in 1988, several RP processes which use different mechanisms or materials have been introduced. Some of them such as SLA, Laminated Object Manufacturing (LOM), Selective Laser Sintering (SLS), Fused Deposition Modeling (FDM), and three dimensional printing (3DP) have become commercially available. RP systems have been reviewed frequently from different perspectives [1-5].

From the part prototyping sequence point of view, RP processes can be categorized into two groups. In layer-based processes, the entire layer is built at once by applying binder or heat to the base material (e.g., speedpart or to some extend 3DP). In point by point-based processes, the machine's build head is moving point by point in one layer to extrude, glue, fuse, or cut the material. In spite of the fact that the second method is slower, some of the most popular RP processes such as SLA, SLS, and FDM are in this group. Obviously, for the point-by-point RP processes a machine path is needed to control the movement of the machine's build head. Because of the differences in the RP processes in this group, there is no standard machine path code for them. For each RP process, based on its characteristics and requirements, the required data are extracted from the CAD model and converted into proper format for the RP machine. Part external wall formation, filling method, and part separation from the surrounding material determine the machine path pattern. This machine path pattern could be a robotic movement in the XY plane for FDM or Contour Crafting

---

<sup>†</sup> Corresponding author. Tel.: +1- 5122453059. Fax: +1-5122453052  
E-mail address: ba13@txstate.edu (Bahram Asiabanpour).

machines, a laser pattern for material solidification and sintering in the SLA and SLS machines, or a laser cutter pattern for the LOM machines [3].

All popular RP processes make parts in a uniform layer thickness fashion. Consequently most of the path generation efforts have focused on that area. In the uniform layer thickness method, all layers have equal thickness. Some theoretical and software works have also been proposed in adaptive layer thickness prototyping. In the adaptive slicing technique the slicing thickness varies according to the part geometry complexity. It is theoretically proven that this method can produce parts with higher accuracy, less deviation from the CAD model, and shorter fabrication time [6-8]. Although because of the technical difficulties, the implementation of this method on the current commercial RP processes has not been reported.

Slicing a CAD model through intersecting the model with the XY-plane at each Z increment is a well-known method of path generation. Slicing a CAD model is carried out through different methods such as stereolithography (STL) file slicing or direct model slicing. Among the available RP processes, the STL format is the most widely accepted file format in the RP industry. The reasons for this popularity are the simplicity of the format and ease of file generation without requiring very sophisticated CAD software [9 and 10]. Different methods for slicing an STL file have been proposed [3, 11, and 12].

Another slicing approach uses exact CAD models. Jamieson and Hacker [13] proposed a direct adaptive slicing inside a CAD system and then transferring the resultant contour to the RP machine. Computing exact contours of intersections is another slicing approach. Rajagopalan [14] developed a technique which intersects an object with a plane in the I-DEAS CAD software. The output is a contour containing a set of Non-Uniform Rational B-Splines (NURBS) curves. Similarly, Chen X. et al. [15] proposed the direct slicing macros inside PowerSHARE package for slicing an object model at certain Z level.

This paper proposes a direct slicing approach inside a design-by-feature solid modeler, Autodesk Inventor. Autodesk Inventor is one of the most popular solid modeling CAD software programs on the market. Because of its capabilities and ease of use it is widely used in the US industries, universities, and even high schools. One of the unique features of this new system is its ability to process very large and complex models. Additionally, it is capable of producing machine paths that are compatible with most popular commercial RP processes (e.g., LOM, SLS, and SLA).

The rest of this paper is organized as follows. In section 2, the structure of the proposed system is explained. In section 3, feature recognition of this system is discussed. Machine path generation algorithm, including facet and contour creation, hatch pattern generation, and system verification are explained in section 4. Finally, conclusion and discussion are given in section 5.

## **2. Direct slicing system structure**

In this approach, first the protrusion and subtractive features that form a model are recognized and extracted. Then, the faces of these features that are intersecting the XY plane at a certain Z level are listed for slicing. Then, for each face the intersections of facets, which are generated by Inventor API functions, with the XY plane are exploited. Facet generation API function is applied only to the faces of the selected feature that intersects with XY plane at each Z increment. Unlike most of the previous works that the system searches the entire

STL file to find an intersecting facet, this system is searching from very limited list of facets resulting from the intersecting features with XY plane. Therefore, the run time for machine path generation is significantly shorter. Finally, intersecting lines are sorted to form closed loop contours. While computing a contour, hatching data at X and Y directions are also computed to produce custom designed machine path for different RP processes.

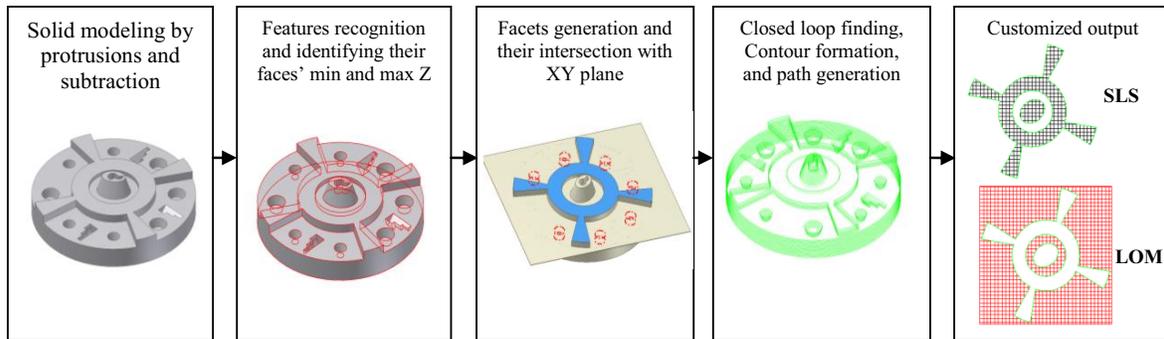


Figure 1. Direct slicing and path generation system structure

### 3. Feature recognition

Autodesk Inventor as a solid modeler uses various design features to form a solid model. The approach used for CAD models in this research is called design-by-feature solid modeling. All design features that form a main part are classified into two groups. The first group includes the features that decrease a volume of a part; these are subtractive features. The second group includes the features which increase a part volume; these are protrusion features. Each design feature in Inventor is defined as an object whose all properties would be accessed by Visual Basic codes. Some design features (e.g. Extrude, Revolve, Loft, Coil, and Sweep) in Inventor have a special property, called “Operation”, by which the type of features is easily recognized. In contrast, some features (e.g. Shell, Hole, Chamfer, and Fillet), based on their design, are recognized as subtractive. In addition to subtractive or protrusion features recognition, other important data for the model slicing such as the maximum and minimum Z points of a feature and the faces of a feature with their maximum and minimum Z points are extracted.

By having such data, instead of intersecting all faces of a part with XY plane at every Z level, the intersection lines are calculated only for the faces that are certainly intersect with XY plane in that Z level. Consequently, the system’s run time is significantly decreased. The type of recognized features will also contribute to the proposed system in generating an internal/external hatch for the custom designed machine path for different RP processes.

### 4. Machine path generation algorithm

After features recognition, the shortlist of faces intersecting XY plane at a  $Z=z$  are sent to the Inventor’s API function for the purpose of facet creation. At each increment of Z, a set of points resulted from intersection of XY plane with the created facets is generated. The next step deals with connecting the points to each other to form the loop(s) or contour(s). The final step uses the contour(s) to generate a machine path (boundary path and hatch pattern).

#### 4.1. Facet creation

As shown in Figure 2, a model may encompass numerous design features. With the assumption that a cross-section XY plane is passed through  $Z=z$ , the features F1, F2... F13 are selected due to crossing the XY plane at  $Z=z$ . Some faces of the selected features that

intersect with XY plane, are then selected and sent to the Inventor's facet creation function (GetExistingFacets). The intersection between XY plane and the created facets results in a group of vectors (lines). A data structure containing the following fields has been provided that would be exploited for saving the properties of each obtained vector.

- SP (indicating the start point of a vector)
- EP (indicating the end point of a vector)
- CreatedByFeature (Determining the type of a feature that would contribute in identifying external and internal hatch)
- Flag (Boolean value that would prevent from double computation)

Red colored texts indicate features that are selected for intersection purpose

Green colored texts indicate faces that are selected for intersection purpose

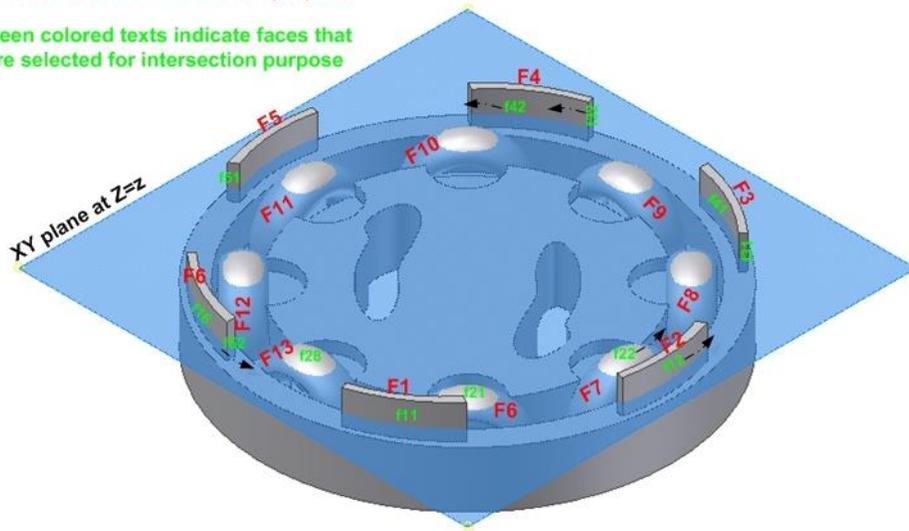


Figure 2. A sample part for facet creation test in Inventor

#### 4.2. Contour creation

After creating the intersecting vectors in each Z increment, they should be ordered and connected to form a continuous path. Flowcharts shown in Figures 3-5 illustrate the facet and contour creation main algorithm and its subroutines. In order to sort the vectors, algorithm starts from one vector and tries to find the next connected vector to this vector in a machine path generation process. Then, it continues the same approach until it reaches initial point of the first vector (a closed loop). While forming a vector loop using the required vectors from within the data structure, the field (Flag) of each selected vector is set to "True" by which the vector is prevented from double computation for the next loops. The connected vectors, which formed a loop, are saved in an array representing the loop's properties. This approach is followed until there is no vector whose Boolean field possesses the value of "False". Using the vectors data saved in the array, a boundary path for a RP machine based on the syntax of machine path code can be generated. The Selective Inhibition of Sintering (SIS) [16] machine, for example, has been taken into consideration for boundary path and hatch path generation in this research.

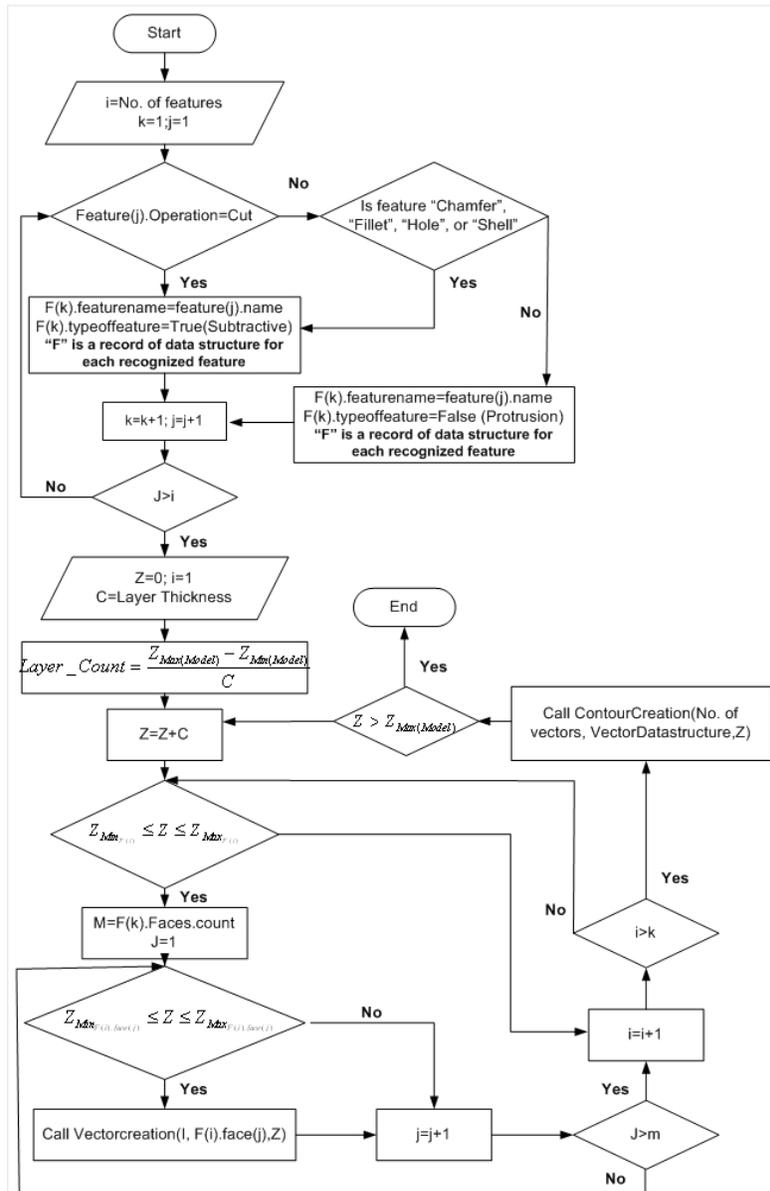


Figure 3. The boundary path generation algorithm

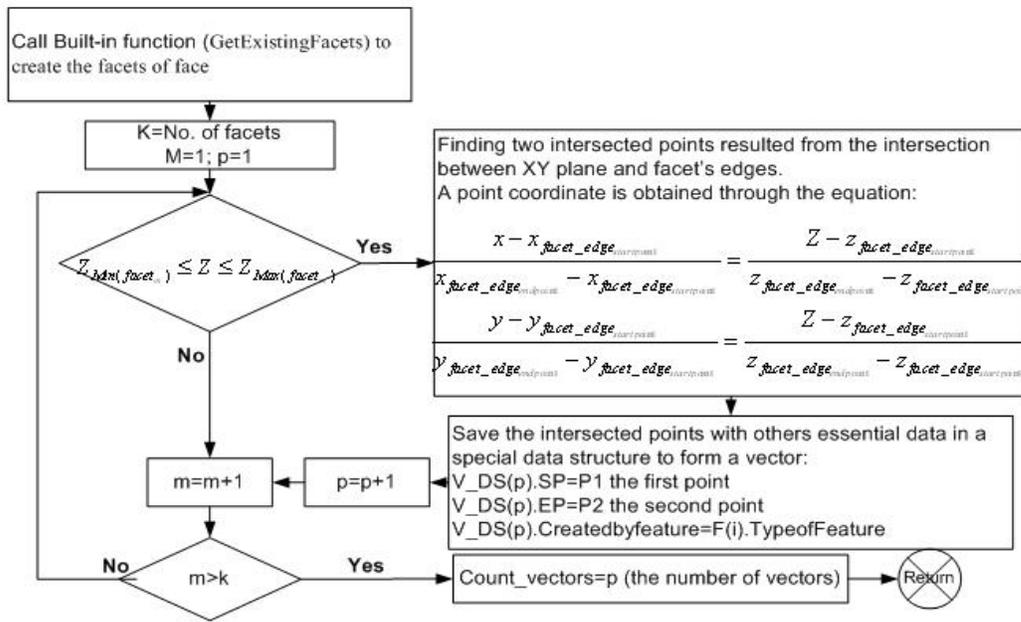


Figure 4. Subroutine for vector creation

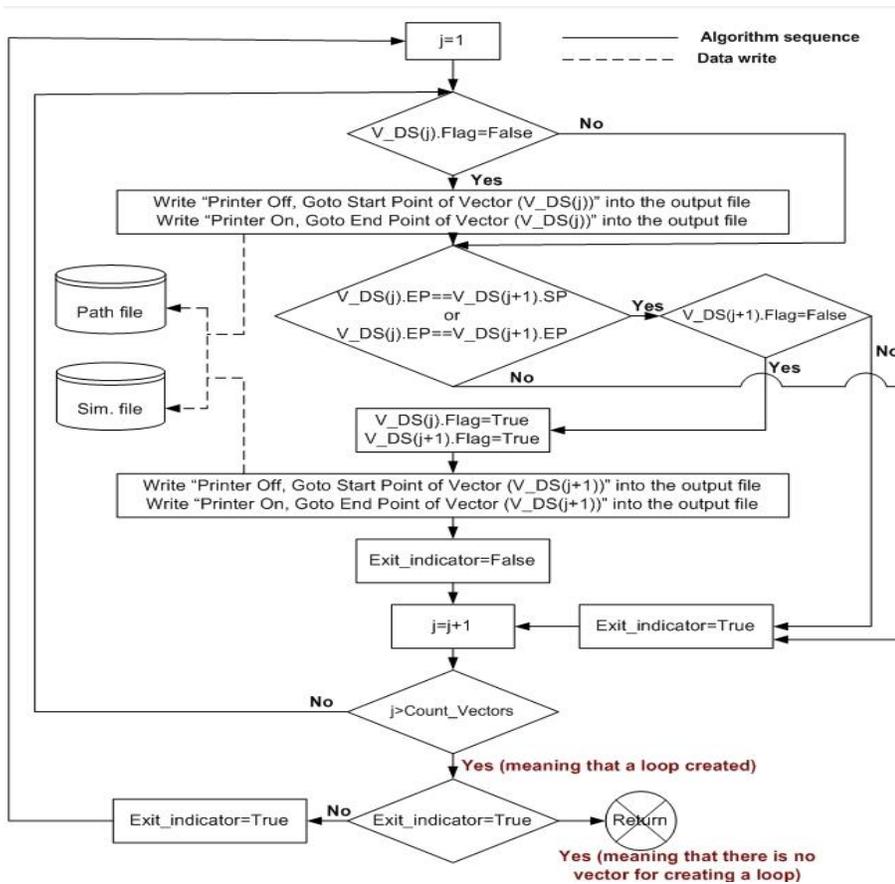


Figure 5. Subroutine for contour creation and machine path generation

### 4.3. Hatch pattern generation and customization

Hatch patterns are needed for different RP processes. For processes such as SIS and LOM, external hatch patterns are needed to extract the final part from the surrounding materials. Furthermore, in processes such as SLS and SLA, an internal hatch pattern is needed to make fully solid parts. In this system, hatch spacing in X and Y direction are adjustable. The flowchart shown in Figure 6 depicts the external hatch pattern generation algorithm.

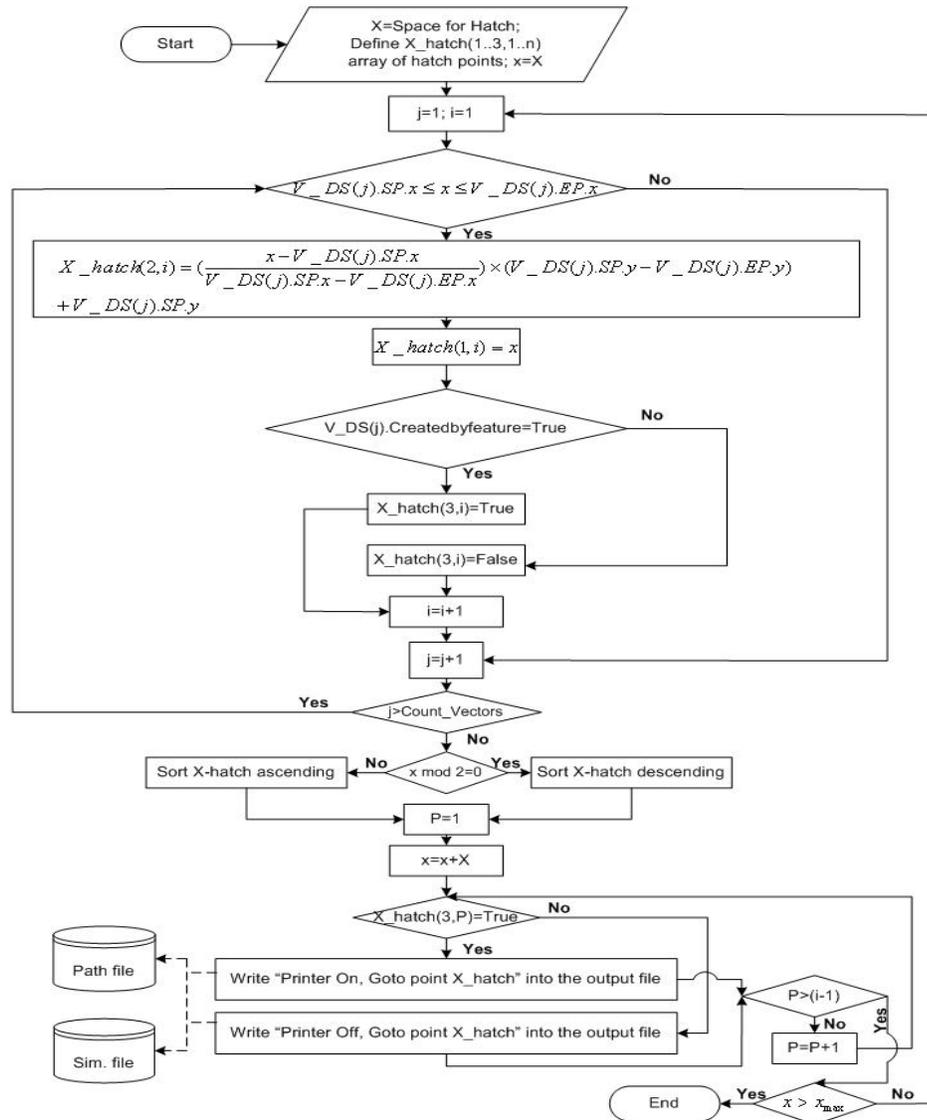


Figure 6. Subroutine for external X hatch pattern generation

As shown in this figure, the intersecting vectors of each layer that were saved in an special data structure, are in turn considered for the possibility of intersection with the lines passing through  $X=x$  and  $Y=y$ . The hatch intersection points are sorted in ascending and descending orders for every other line. After sorting the intersection points, the Printer (for the SIS process or laser for LOM) On/Off command for the hatch lines are set. As described in Section 4.1, a vector has a typical field (CreatedByFeature) by which the type of feature is

assigned in order to contribute to the hatch generation process. When the intersection, for example, between a line vertically passing through  $X=x$  and a vector extracted from a facet of a subtractive feature occurs, the field (CreatedByFeature) of data structure belonging to the achieved point is set to “True” by which a machine code (Printer On herein) is defined. This approach continues for all the intersected points in X and Y hatch direction.

#### 4.4. System implementation

The mentioned algorithms were implemented using the Visual Basic for Application (VBA) and Autodesk Inventor mechanical design software. The Autodesk Inventor Application Programming Interface (API) functions were used to gain access to specific objects of a part model. The Inventor’s VBA provided an easy way of accessing the design features and generating the facets for all faces of the features. A user interface form facilitates data entries (i.e. layer thickness and hatch spacing in X and Y directions) Figure 7 illustrates user interface form.

#### 4.5. System verification

This software was successfully tested for many complex models with a variety of features. For better evaluation of the system, an AutoCAD script file generation was added to the system’s output. Such a file is easily run in AutoCAD to illustrate the boundary and hatch patterns for different layers (Figures 8 and 9).

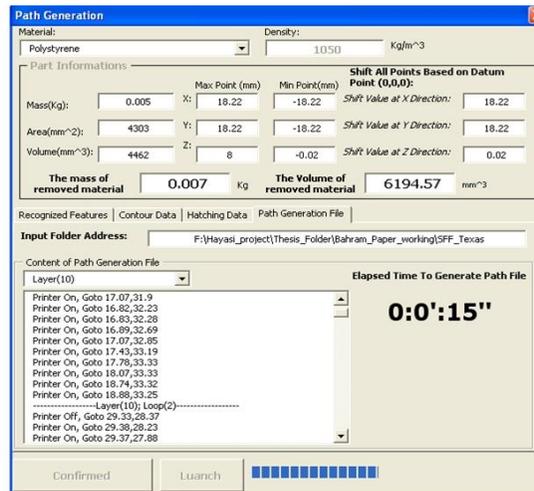


Figure 7. The system user interface

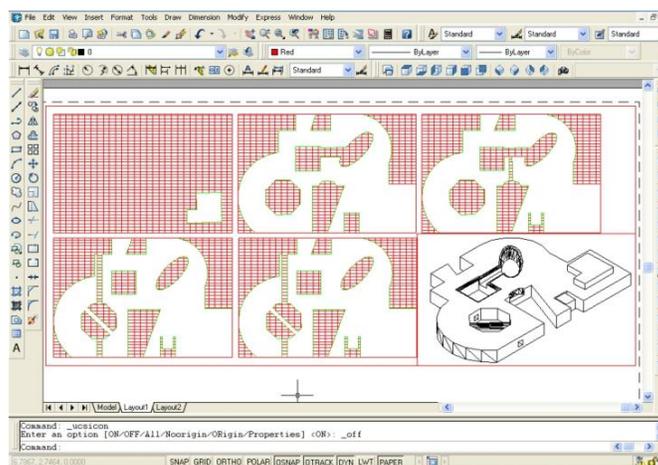


Figure 8. Visualization of CAD model and a few slices of the machine path for the RP processes with external hatch (e.g., LOM and SIS)

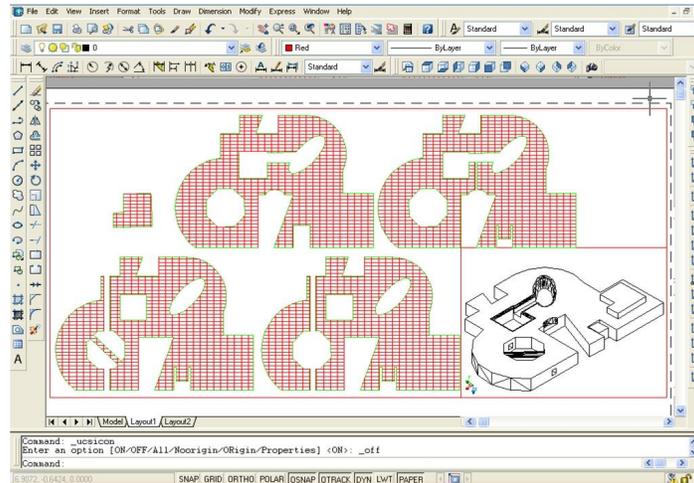


Figure 9. Visualization of CAD model and a few slices of the machine path for the RP processes with internal hatch (e.g., SLA and SLS)

## 5. Conclusion

This paper proposed a direct slicing approach in order to generate a machine path inside a design-by-feature solid modeler (Autodesk Inventor). The proposed system was implemented by Visual Basic code inside Inventor. The proposed algorithm used the facets created for each face of a solid model inside Inventor. Subtractive and protrusion features that formed a solid model were extracted, by which the system was enabled to find internal and external loops as well as to generate internal and external hatch. The system reported in this paper has been successfully tested on several complex 3D models created in Inventor. Because this system receives the CAD data directly from the solid modeler system, unlike other systems that use STL files, it never faces problems such as low quality STL file. The system output can also be custom generated for RP processes that need external hatch (e.g., SIS and LOM) or internal hatch (e.g., SLS and SLA).

## References

- [1] Yan, X., and Gu, P., "A review of rapid prototyping technologies and systems", *Computer Aided Design*, 1996, Volume 28, Number 4, PP. 307-318.
- [2] Onuh SO and Yusuf Y., "Rapid prototyping technology: applications and benefits for rapid product development", *Journal of Intelligent Manufacturing* 1999; 10:301-311.
- [3] Asiabanpour, B. and Khoshnevis, B., "Machine path generation for the SIS process", *Robotics and Computer-Integrated Manufacturing* 2004; 20:167-175.
- [4] Chua, C.K., Chou, S. M. and Wong, T.S., "A study of the state-of-the-art rapid prototyping technologies", *The International Journal of Advanced Manufacturing Technology*, 1998, Volume 14, PP. 146-152.
- [5] Kamrani, A. and Nasr E., "Rapid Prototyping: Theory and Practice", Springer, 2006.
- [6] Dolenc A. and Mäkelä I. "Slicing procedure for layered manufacturing techniques", *Computer-Aided Design* 1994; 26(2):119:126.
- [7] Vouzelaud, T. and Bagchi, A., "An adaptive lamina generation for shape dependent process control and/or object decomposition", 1995, Patent Number: 5,432,704.

- [8] Kulkarni, P., Marsan, A., Dutta, D., “*A review of process Planning techniques in layered manufacturing*”, 2000, Rapid Prototyping Journal, Volume 6, Number 1, PP. 18-35.
- [9] Jurrens, K.K., “*Standards for the rapid prototyping industry*”, Rapid Prototyping Journal, 1999, Volume 5, Number 4, PP. 169-178.
- [10] Koc, B., Ma, Y., Lee, Y. “*Smoothing STL files by Max-Fit biarc curves for rapid prototyping*” Rapid Prototyping Journal, 2000, Volume 6, Number 3, PP. 186-203.
- [11] Vuyyuru, P., Kirschman, C. F., Fadel, G., Bagchi, A., Jara, C. C., “*A NURBS-based approach for rapid product realization*”, Proceedings of the 5<sup>th</sup> International Conference on Rapid Prototyping, The university of Dayton, 1994, PP. 229-239.
- [12] Choi, S. H. and Kwok, K. T., “*A memory Efficient slicing Algorithm for large STL files*” Solid Freeform Fabrication Symposium, Austin, Texas, 1999, PP. 155-162.
- [13] Jamicson R and Hacker H., “*Direct slicing of CAD models for rapid prototyping*”, Rapid Prototyping Journal, 1995; 1(2):4-12.
- [14] Rajagopalan, M., “*An approach to CAD interface for stereolithography using Non-Uniform Rational B-Splines*”, 1992, MS thesis, Clemson University, Clemson, SC.
- [15] Chen X, Wang CY, Xiao Y, Huang S., “*Direct slicing from PowerSHARE models for rapid prototyping*”, International Journal of Advanced Manufacturing Technology, 2001; 17:543-547.
- [16] Khoshnevis, B., Asiabanpour, B., Mojdeh, M., and Palmer, K., “*SIS - A New SFF Method Based on Powder Sintering*”, Rapid Prototyping Journal, 2003, Volume 9, Number 1, Pages 30-36.