Eng. &Tech.Journal, Vol. 33,Part (A), No.2, 2015

Tool Path Generation for CNC Milling based on STL file

Nadia Sami Hasan
Production Engineering and Metallurgy Dep.University of Technology/ Baghdad.
Dr. Laith Abdullah Mohammed
Production Engineering and Metallurgy Dep.University of Technology/ Baghdad.
Email:dr.laith@uotechnology.edu.iq

Received on:6/4/2014 & Accepted on:8/1/2015

ABSTRACT

This paper proposes and develops algorithms to read (STL) files and extract engineering entities required in CNC milling processes. The proposed algorithms are dependent on some mathematical modeling and manipulations of the engineering models by slicing an (STL) file to many slices and then building the required algorithms to adopt these slices to generate machining paths for CNC milling machines as (G-Codes).

The proposed system is divided into three parts, in the first part, an algorithm proposed to extract engineering object entities to some proposed models based on their (STL) files using Matlab program. The proposed models include cube, cylinder, dome, cone and cavity models. In the second part, a slicing algorithm is proposed to enable the slices along the proposed models z-axis to find and navigate the required manufacturing data. UGS program was used also to generate the tool paths and to simulate the machining process and then generate NC part program of the proposed objects (G-Code). Finally, the third part of this work includes comparing results produced based on both (STL) and (UGS), to achieve the required aim of this paper by experimentally comparing objects surface roughness, resulted values show that the maximum difference in average R_a is equal to (0.27µm).

Key Words: STL files, CNC Milling.

توليد مسار العدة لمكائن التفريز المبرمجة بأعتماد ملف STL

الخلاصة:

في هذا البحث تم افتراض وتطوير خوارزميات لقراءة ملفات (STL) واستنباط السمات الهندسية اللازمة لعمليات التفريز المبرمجة. اعتمدت الخوارزميات المطورة على النمذجة الرياضية والمعالجة لنماذج هندسية من خلال تقسيم ملف (STL) الى عدة شرائح وبعدها بناء خوارزميات لاعتماد هذه الشرائح في توليد مسارات القطع لمكائن التفريز المبرمجة كبرامج تشغيل (G-codes).

تم تقسيم النظام المقترح الى ثلاثة اجزاء. في الجزء الأول تم اقتراح خوارزمية لاستنباط السمات الهندسية لمجموعة من النماذج المقترحة اعتمادا على ملفاتها (STL) برنامج معد على نظام Matlab. تشمل النماذج المقترحة

https://doi.org/10.30684/etj.33.2A.8
2412-0758/University of Technology-Iraq, Baghdad, Iraq

المكعب ، الأسطوانة ، القبة ، المخروط والتجويف. في الجزء الثاني تم اقتراح خوارزمية للتقطيع للحصول على شرائح على طول محور z للنماذج المقترحة للحصول على البيانات التصنيعية. تم ايضا استخدام برنامج (UGS) لتوليد مسارات القطع ومحاكاة عملية التشغيل ومن ثم توليد برامج التشغيل لهذه النماذج. الجزء الثالث تضمن مقارنة النتائج المتولدة من كل من اعتماد ملفات (STL) واعتماد ملفات برنامج (UGS) من خلال المقارنة العملية لقيم الخشونة السطحية، حيث بينت النتائج أن أكبر فرق في معدلات قيم الخشونة السطحية يساوي (0.27µm).

INTRODUCTION

Today manufacturing relies heavily on CNC machines. The operations of CNC machines require modeling a part in the Computer Aided Design (CAD) system, many CAD systems are being used and each of them uses a different kernel to describe the geometry. It is not possible to read all of these different data formats using one software package. A standard format that can be output by most commercial CAD systems should be used as the input for Computer Aided Manufacturing (CAM) system [1, 2].

An STL file is a type of standardized computer exchange file which contains a 3D model. The representation of the surface(s) of the object(s) in the file is in the form of one or more polygon meshes. The polygon meshes in an STL file are entirely composed of triangular faces, edges and vertices. Further, the faces have assigned normal's which indicates their orientation (inside/outside).

The name STL is taken from its extension, .stl, originally because the files were intended for the rapid prototyping process called Stereolithography. The file format has become a world standard for exchanging 3D polygon mesh type objects between programs, and .stl's are now used as input for virtually all rapid prototyping processes, as well as some 3D machining [3].

There are many articles in literature that deal with different CAD formats used in CNC milling operations. In this section the most relative articles related to the present work are reviewed. The research of R. Lin and C. Ye (2012) [4] proposed a method to produce accurate tool paths using STL file formats and machined using 5-axis machining. The result shows that the trajectory contour error is improved about 5 times by using conventional method and the proposed approach, respectively. The new method improves not only the max. contour error, but part surface smoothness. A new tool-path generation method is presented by M. Li, et al (2012) [5] by adopting the thought of generating cutter-location (CL) data directly from corresponding cutter-contact (CC) data. A computational model is proposed to calculate single-layer interference-free CL contour, instead of computing a precise CL point by checking potential interferences from candidate facet, vertices, and edges, respectively. So this method is more efficient and easier for program implementation. Implementation tests prove that the new method is effective and robust for STL models, and tool-path achieved is highly precise. **G.** Kiswanto and M. Azka (2012) [6] research focuses on detection of primitive features, based on faceted models, available in a part, this is done by applying part slicing and grouping adjacent facets. The results showed that this method can identify the primitive features automatically accurately in all planes of tested part, this covered pocket, cylindrical and profile features. To compute vertex normal of triangular meshes more accurately, the paper of Y. Chen, et al (2013) [7] presents an improved algorithm based on angle and centroid weights. Firstly, algorithms are analyzed by comparing their

weighting characteristics such as angles, areas and centroids. Following that, an improved algorithm is put forward based on angle and centroid weights. Finally, by taking the deviation angle between the nominal normal vector and the estimated one as the error evaluation standard factor. Experimental results demonstrate that the algorithm is effective.

The Proposed System

Figure (1) describes the main steps adopted in the algorithms which are achieved in this work. The basic modeling primitives that are used in this study are parameterized as follows:

1- A cube which is parameterized by its width, height, and length.
2- A cylinder which is parameterized by its radius and height.
3- A dome which is parameterized by its radius.
4- A cone which is parameterized by its radius and height.
5- Finally, A cavity which is parameterized by its radius.

AutoCAD program facilities are used to draw the object wireframe model as shown in Figures (2 a) which represents the wireframe models for the cube starting from the origin with 60 mm length, Figure (2 b) which represents the wireframe model for the cylinder with a base circle center starting from the origin with radius equal to 25 mm and cylinder height is equal to 60 mm, and Figure (2 c) which represents the wireframe model for the dome starting from the origin with radius equal to 50 mm, and Figure (2 d) which represents the wireframe model for the cone starting from the origin with radius equal to 25 mm and its height is equal to 60 mm, and finally Figure (2 e) which represents the wireframe model for the cavity starting from the origin with radius equal to 50 mm. These wireframe models were saved as an STL file extension format using "STLOUT" AutoCAD command. Consequently, this file format will be modeled for manipulation using MATLAB Package environment using an algorithm depending on the following procedures:

Object Mathematical Representation

In this stage of modeling, the program is built using MATLAB R2010a as a first step to generate the database of geometry as a set of points in three axes in Cartesian coordinate system. The set of points that represent the shape of the objects shapes models obtained from previous stage will be exported to the MATLAB environment to transfer the objects as a set of points to any other CAD/CAM software without any distortion. Figure (3) shows the models triangulation meshs for the proposed cube, cylinder, dome, cone and cavity shape respectively.

Based on this figure, one can recognize 12 triangular meshes with 36 vertices which represent the cube model, 144 triangular meshes with 432 vertices which represent the cylinder model, 900 triangular meshes with 2700 vertices which represent the dome model, 72 triangular meshes with 216 vertices which represent the cone model, and 900 triangular meshes with 2700 vertices which represent the cavity model.

The next step including the transforming process to the exported models to a set of surfaces as shown in Figure (4) which represent the cube, the cylinder, the dome, the

cone, and the cavity triangular surface models respectively, according to a procedure achieved using a developed algorithm in MATLAB environment.

Triangular-Mesh Intersection Detection

The triangular-mesh intersection detection approach is commonly used in current 3D graphics hardware and software, as it provides a precise intersection detection mechanism. Essentially, the triangular-mesh technique involves the detection of an intersection between a line and a triangle. However, this process needs to be applied for every triangle that comprises the primitive.

The foundation of the triangular-mesh approach is the algorithm used to detect an intersection between a line and a triangle in three dimensions. This process has been studied in depth to develop fast and efficient techniques. Existing techniques can be divided into two broad categories.

The first category of algorithms uses a two step process. These algorithms first locate where the straight line intersects with the plane containing the triangle. Following this, the algorithms detects if the intersection point lies within the triangle as shown in figure (5). The equation for the plane containing this triangle is given by Equation (1).

$A_x + B_y + C_z + D = 0$
where: $A = (y_2 - y_1)(z_3 - z_1) - (z_2 - z_1)(y_3 - y_1)$
$B = (z_2 - z_1)(x_3 - x_1) - (x_2 - x_1)(z_3 - z_1)$
$C = (x_2 - x_1)(y_3 - y_1) - (y_2 - y_1)(x_3 - x_1)$
Hence;

$$\mathbf{D = -A \times x_1 - B \times y_1 - C \times z_1} \qquad \ldots..(1)$$

Using Equation (1) for the plane in 3D space, it is possible to determine the intersection point (x_i, y_i, z_i) using Equation (2).
$N = -(A \times x_s + B \times y_s + C \times z_s + D)$
$M = A(x_e - x_s) + B(y_e - y_s) + C(z_e - z_s)$

$$S = \frac{N}{M}$$

$$\begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} = S \times \begin{bmatrix} x_e - x_s \\ y_e - y_s \\ z_e - z_s \end{bmatrix} + \begin{bmatrix} x_s \\ y_s \\ z_s \end{bmatrix} \qquad \ldots.. (2)$$

Using Equation (2) one can determine the point of intersection coordinates between the line and the triangular plane, accordingly, Figure (6) represent the cube, the cylinder, the dome, the cone, and the cavity models respectively including the normal vectors for each triangular mesh.

The Slicing Process

The slicing planes, as shown in Figure (7), are planes parallel to the reference plane. In this case, the XY plane is simply chosen as reference plane. Then the database is

developed for facets which are intersected by each slicing plane. Facets indexes are stored in a vector data structure. The interval between slicing planes is determined based on the specified accuracy and also the distance between Z_{min} and Z_{max} value of the faceted model. Once the information regarding the index of facet, index of vertex, and coordinates of vertex are identified and stored in the data array, then facets that are intersected by the slicing plane are to be found. The search is done through the facet index. The slicing algorithm can be explained as follows in the flowchart shown in Figure (8).

In order to minimize searching and manipulating time and to avoid stair case, this algorithm is performed by checking each triangular facet, which has a normal vector parallel to the slicing plane, whether intersected by the slicing plane or not. If it is intersected, the intersection point coordinates is stored in an array data. Intersection checking with each facet was conducted for each slicing plane along the slicing axis. Figure (9) represent the resulting slicing process achieved on faceted cube, cylinder, dome, cone and cavity models respectively. Accordingly, these obtained figures represent the array manufacturing data for the suggested models, which is then exported to the UGS package as .dat file to achieve the machining process including tool path and g-code generation. Figure (10) represent the machined objects using the proposed algorithm.

Measurements Of Surface Roughness

The average roughness (R_a) has been examined for the machined objects by using surface roughness tester brand "Pocket Surf" USA made, this parameter (R_a) is chosen because it is simple and widely used as a parameter for the surface finishing. A full experimental procedure is carried out for each machined object, using 1 mm long as a measuring range that is based on the instrument probe movement, holding the object in arrangement permits to achieve five lines measurement recorded along the machining paths and the surface roughness has been measured from the object top towards its bottom perpendicular to the direction of the tool movement. Table (1) lists the resulted average surface roughness data. According this data, it is obvious that the maximum difference in average R_a is equal to (0.27μm).

Conclusions

On the basis of this study, and observations recorded experimentally, the following findings can be concluded:

1- STL file format consists of unordered list of triangular facets.

2- According to the conducted experiments and the plotted results, one can observe that the objects produced using STL models are clearly have rough surfaces compared with the surfaces produced by UGS with average percentage of 13μm.

3- The main cause of the high surface roughness values of the objects produced by STL is the cutting tool length and this motive increases the vibration in the tool axis leading to increase the magnitude of (R_a).

4- The proposed system give less number of machining blocks and minimum machining time as compared with those resulted from the UGS system, this is because the proposed algorithm is performed by checking each triangular facet, which has a normal vector parallel to the slicing plane, whether intersected by the slicing plane or not.

If it is intersected, the intersection point coordinates is stored in a manufacturing array data.

5- The points close to the object top are more prone to false feature extraction than the points around the object surface area, this can be seen clearly in both dome and cone shaped objects respectively.

Table (1) the resulted average surface roughness data

	Cube		Cylinder		Dome		Cone		Cavity	
	UGS	Proposed	UGS	Proposed	UGS	Proposed	UGS	Proposed	UGS	Proposed
Average Ra (μm)	1.7	1.73	1.6	1.74	1.41	1.56	1.67	1.94	1.46	1.52

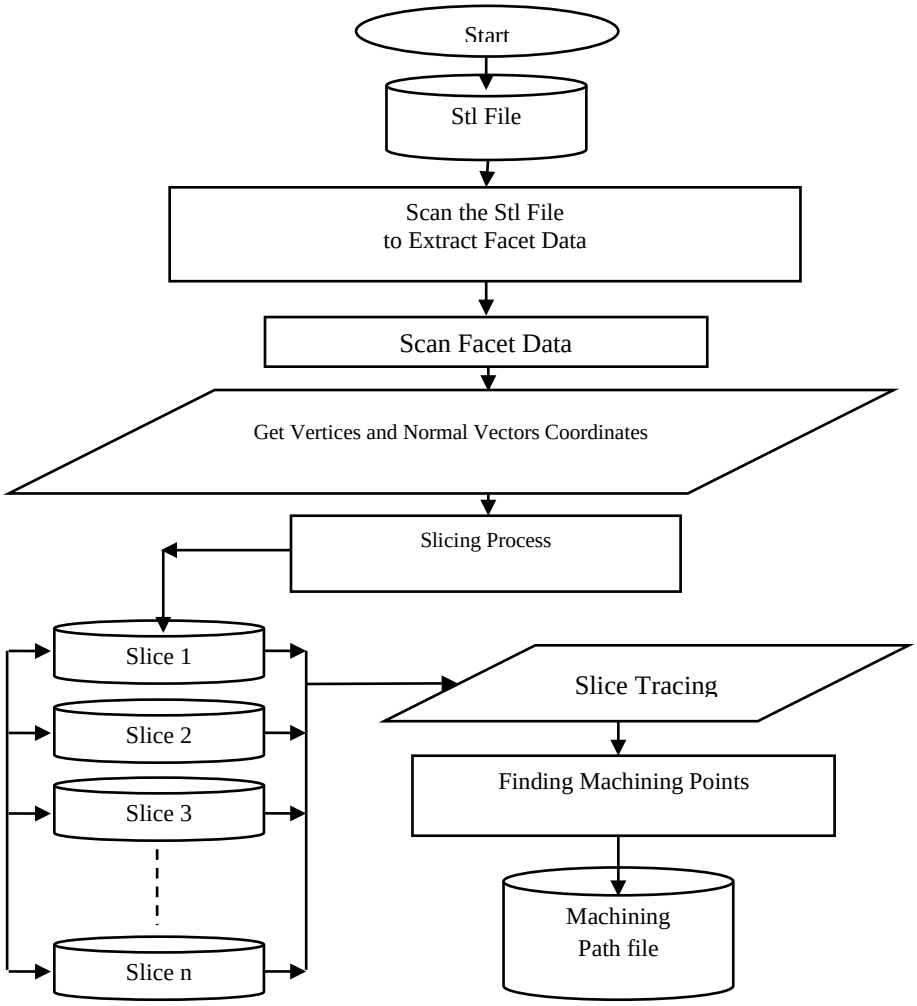
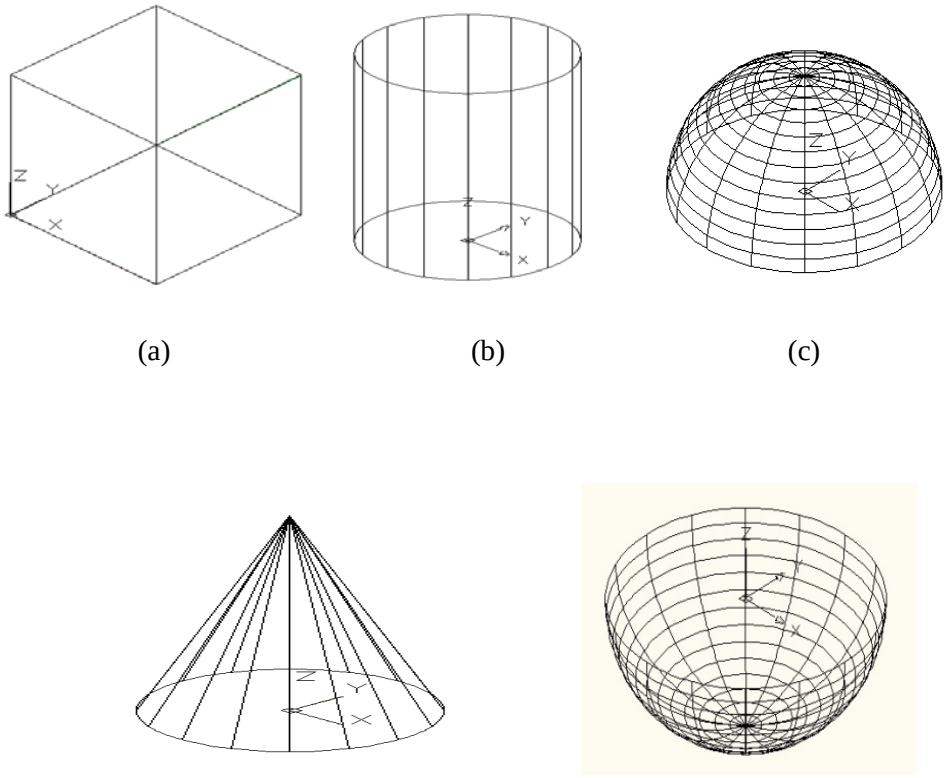


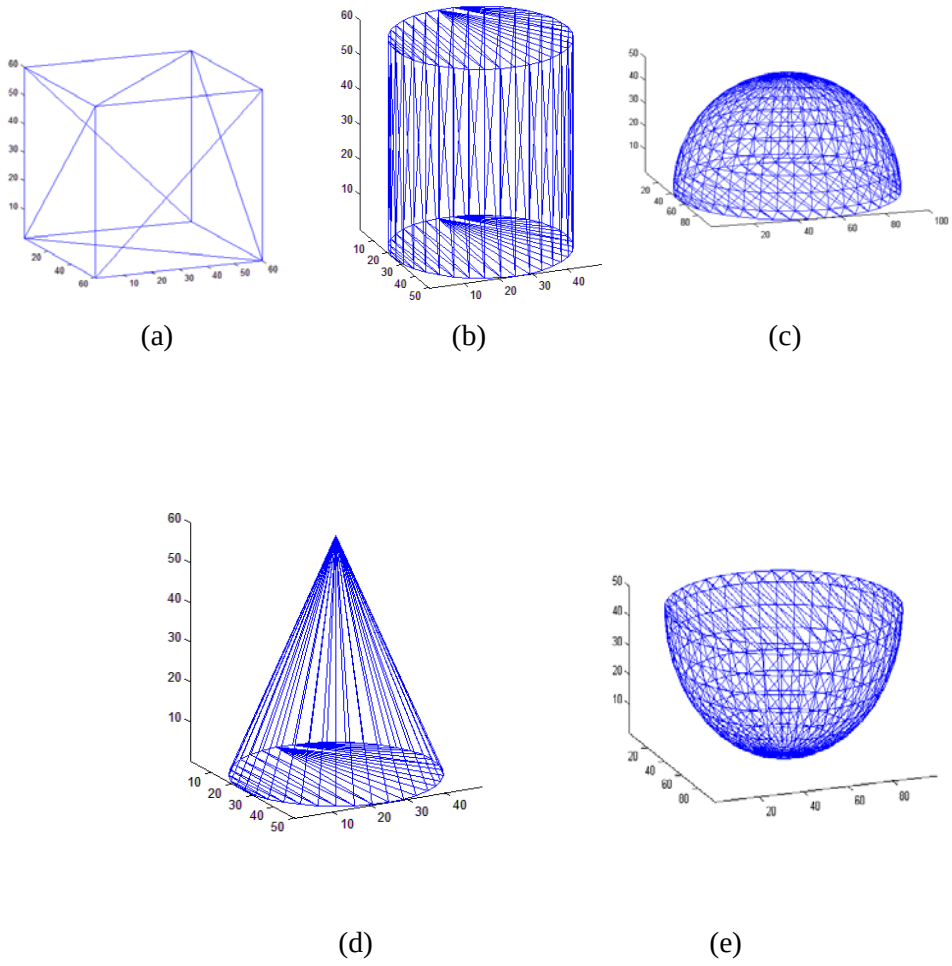
Figure (1) Flowchart describing the main steps of the algorithms achieved in this work



(a) (b) (c)

(d) (e)

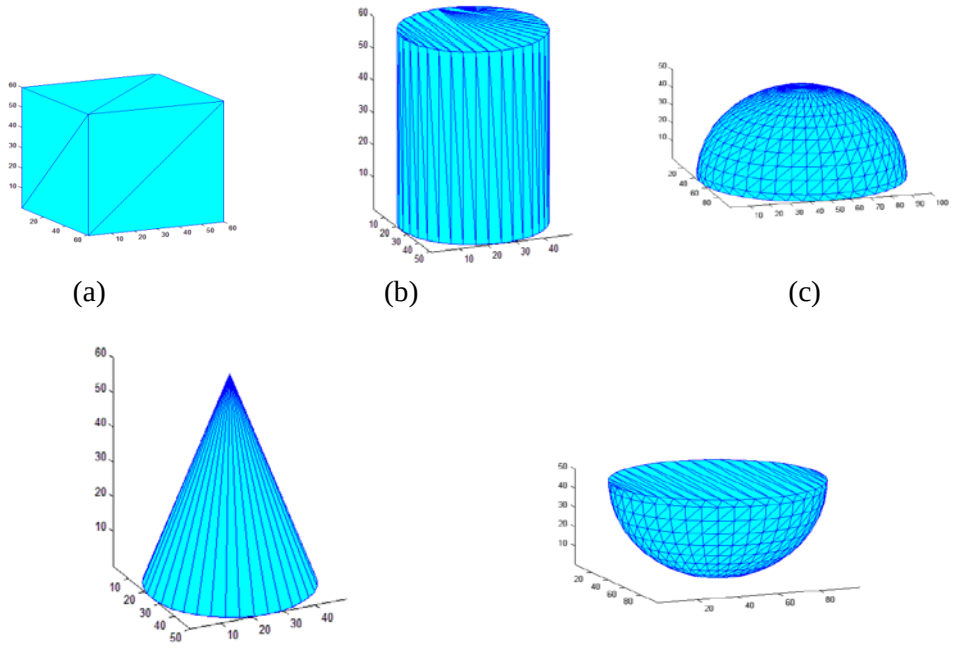
Figure (2) The Main Wireframe Models for the (a) Cube, (b) Cylinder, (c) Dome, (d) Cone, and (e) Cavity



(a) (b) (c)

(d) (e)

Figure (3) The Models Triangular Mesh for the (a) Cube, (b) Cylinder, (c) Dome, (d) Cone, and (e) Cavity



(a) (b) (c)

(d) (e)

Figure (4) The Triangular Surface Models for the (a) Cube, (b) Cylinder, (c) Dome, (d) Cone, and (e) Cavity

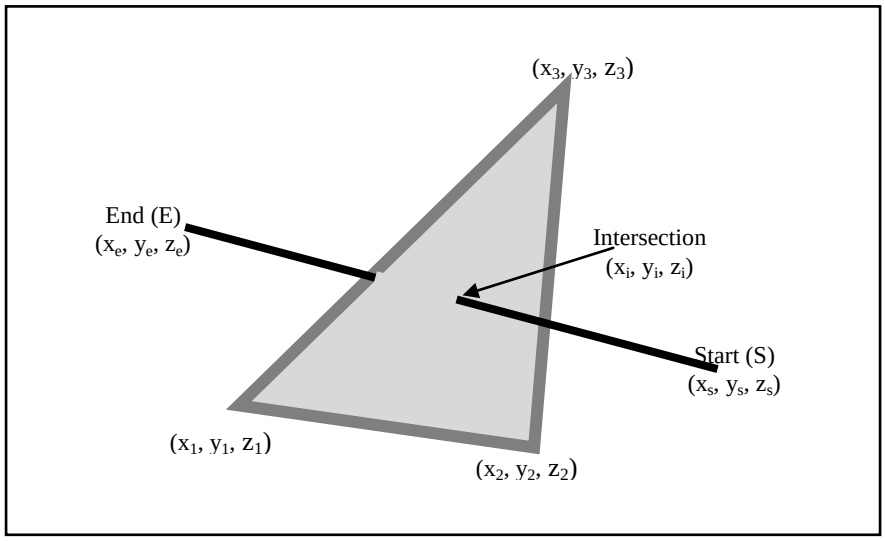
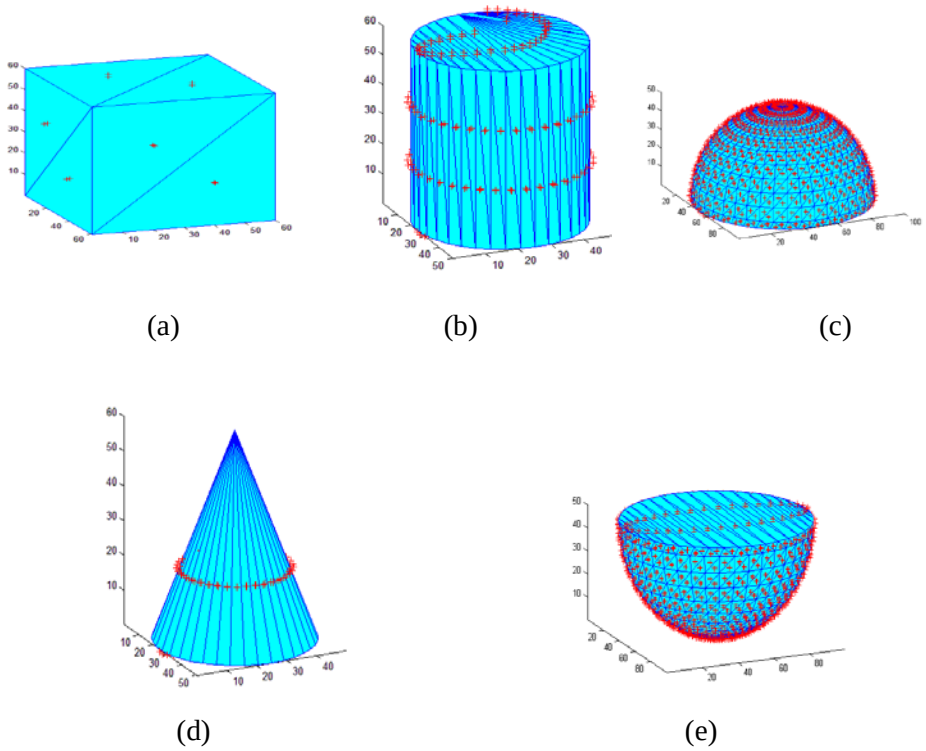


Figure (5) Triangular Mesh Intersection in 3D Cartesian Space



(a) (b) (c)

(d) (e)

Figure (6) The Normal Vectors for Each Triangular Mesh for the (a) Cube, (b) Cylinder, (c) Dome, (d) Cone, and (e) Cavity

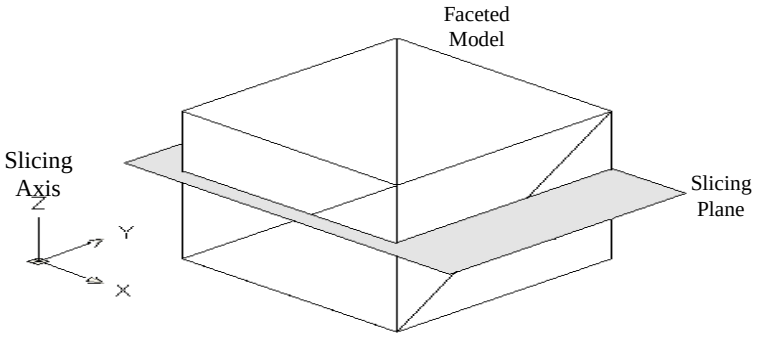


Figure (7) The Slicing Process on the Faceted Models

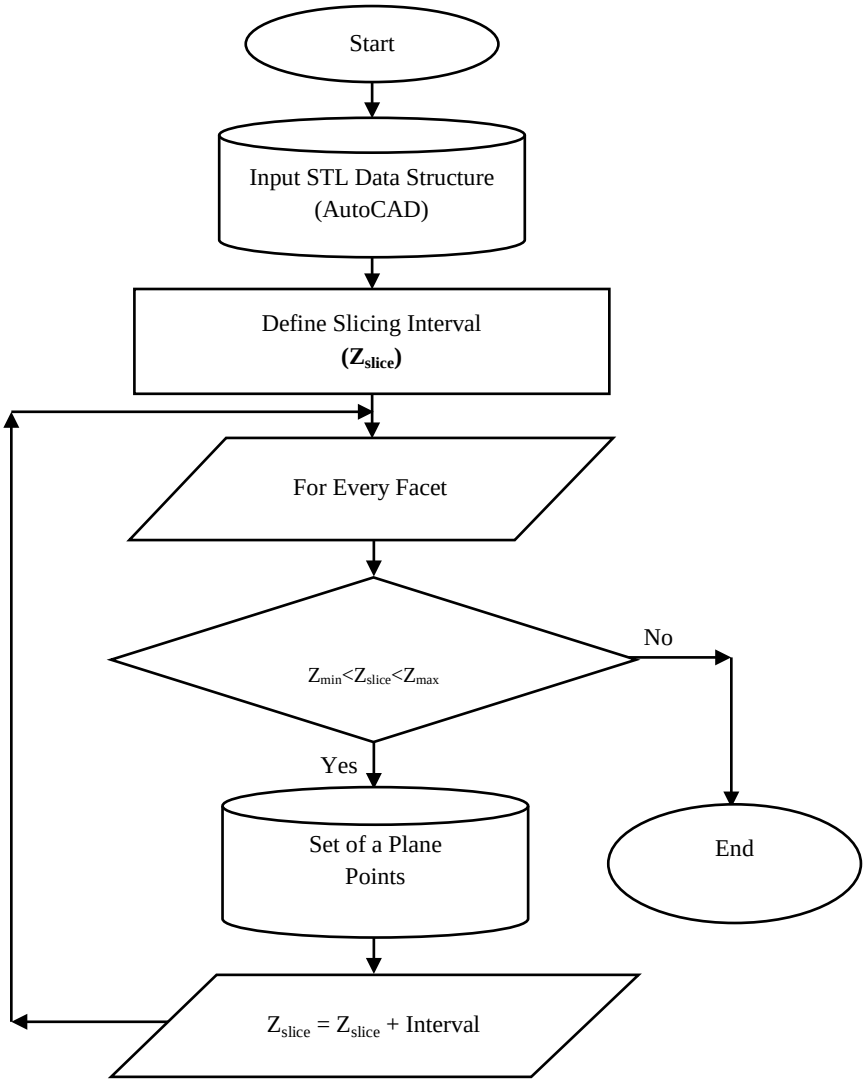


Figure (8) The Developed Slicing Algorithm Flowchart

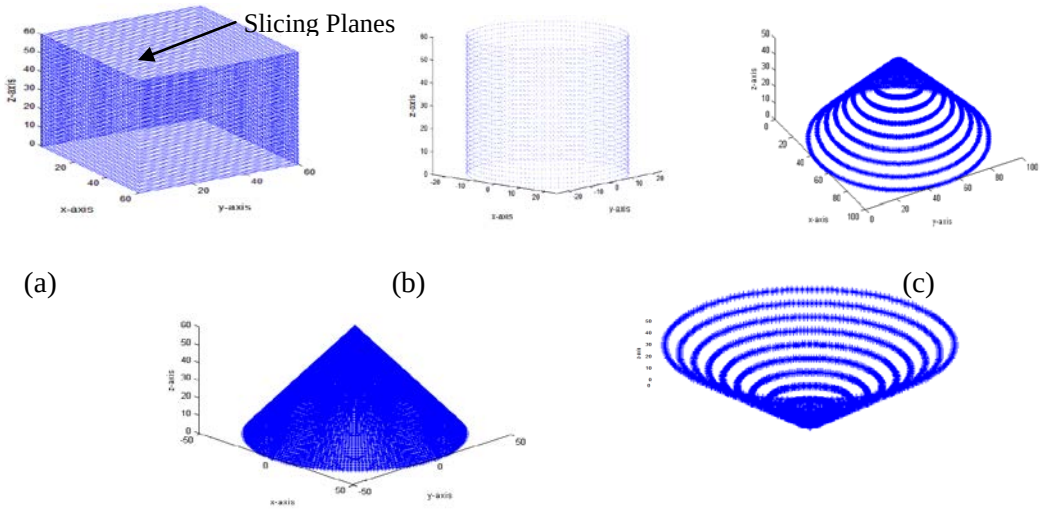


Figure (9) The Slicing Process Achieved on the 3D Models for the (a) Cube, (b) Cylinder, (c) Dome, (d) Cone, and (e) Cavity

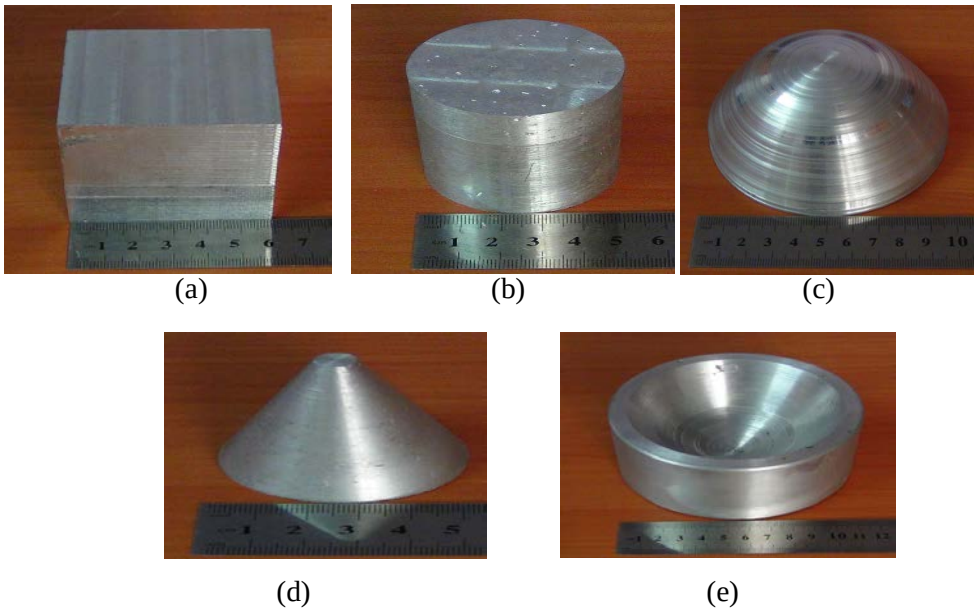


Figure (10) The Machined Objects Using The Proposed Algorithm for the (a) Cube, (b) Cylinder, (c) Dome, (d) Cone, and (e) Cavity

REFERENCES:

[1] J. W. Gunnick, "Multi-Axis High Speed Milling: How to Speed up Prototyping & Tooling Process by Using STL Technology", Proceedings TCT Conference 1998, Nottingham, pp.43-65,Oct. 1998.

[2] L. Lennings, "Selecting Either Layered Manufacturing or CNC Machining to Build Your Prototype", Rapid Prototyping and Manufacturing Conference, Rosemont(Chicago), April 2000.

[3] M. Löwdin, "Time Reducing Actions With Freeform Fabrication and Rapid Prototyping", M.Sc. Thesis, Örebro University,Sweden, 2004.

[4] R. Lin and C. Ye, "Accurate Trajectory Control for Five-Axis Tool-Path Planning", Proceeding of the international Multi conference of engineers and computer scientists, Vol. II, Hong Kong, March. 14-16, 2012. [www.ivsl.org].

[5] M. Li, L. Zhang, J. Mo, and Y. Lu, "Tool-path generation for sheet metal incremental forming based on STL model with defects", International Journal of Advanced Manufacturing Technology, pp. 535–547, (2012). [www.ivsl.org].

[6] G. Kiswanto and M. Azka, "Automatic Part Primitive Feature Identification Based on Faceted Models", International Journal of Computer Science Issues, Vol. 9, Issue 5, No 2, September 2012. [www.ivsl.org].

[7] Y. Chen, J. Gao, H. Wen and X. Chen, "Estimation Normal Vector of Triangular Mesh Vertex by Angle and Centroid Weights and its Application", TELKOMNIKA, Vol. 11, No. 4, pp. 1841-1848, April 2013. [www.ivsl.org].