

## Development of A 3-D Solid Modeling System Based on The Parasolid Kernel for Gem Stones Faceting

Viboon Sangveraphunsiri, Kankriangkrai Sinthudechakul, and Prachya Pipattananont  
Robotics and Advanced Manufacturing Lab, Department of Mechanical Engineering,  
Chulalongkorn University

254 Phyathai, Patumwan Bangkok, 10330, Thailand

Tel. 66-2218-6448, 66-2218-6585 Fax 66-2218-6583

E-mail: [viboon.s@eng.chula.ac.th](mailto:viboon.s@eng.chula.ac.th), [reverserp@eng.chula.ac.th](mailto:reverserp@eng.chula.ac.th)

### Abstract

This research is to develop computer aided design (CAD) software, the CU-Solid. The software covers various type of 3-D solid modeling techniques based on the Parasolid kernel under Windows platform. This software can be used for creating and modifying solid models using Parametric Feature-Based technology. The supporting features included are extrusion, extrusion cutout, revolution, revolution cutout, swept, swept cutout, loft, loft cutout, fillet, chamfer and hollow. The Parasolid Transmission XT format is used as the main data format for storing the solid data. This data format is compatible with many commercial packages using the Parasolid kernel as modeler. The CU-Solid is extended capabilities to cover gem stones design. And from the design data, parameters needed for controlling our in-house development faceting machine can be obtained directly from the CU-Solid.

To verify the compatibility of the data format, with commercial packages, Unigraphics, SolidEdge, and SolidWorks, are used as reference system. These commercial packages based on the same Parasolid kernel. The results shown that the CU-Solid can import and export the commercial packages data files correctly.

**Keywords:** CAD, Solid Modeling, Feature-Based, Parasolid, CU-Solid, faceting machine, gem

### 1. Introduction

Solid modeling system is the most popular modeling system that used in many CAD packages. The models, which create with this modeling system, are look like the real-world work pieces and their data can be used in manufacturing process. However, commercial packages that sold in the present are very expensive.

In this work, we want to develop a graphic or a solid modeling software as a design tool for using with our in-house developed CNC machine, a faceting machine. The software can create and manipulate solid models with featured-based modeling concept and use the Parasolid as the modeler. This software can display solid models in 3-D visualization and it can exchange solids data with

another commercial packages. In additional, this software designed for flexible used such that users can program custom software modules to work with it.

### 2. Solid Model Representation

To represent solid model in CAD software, they are two main model representations often used, “*Constructive Solid Geometry (CSG)*” and “*Boundary representation (B-rep)*”

#### Constructive Solid Geometry (CSG)

A CSG modeler represents solids procedurally, by combining the operations with which they are made [Parasolid Technical Support Group 2001a]. Complex solids, which use CSG modeler, constructed by Boolean operation between simple solids called “*Primitive Solid*” including block, cone, cylinder, sphere, prism, and torus. This kind of modeler have no information about solid’s components such as faces, edges or vertices, then the calculation of solid information can be made very hardly and inaccurately.

#### Boundary Representation (B-rep)

B-rep modeler represents solids by their boundaries [Parasolid Technical Support Group 2001a]. It uses “*Topological entity*” and “*Geometrical Entity*” to describe structures of solids.

Topological entities comprise all the entities that constitute the structure or skeleton of a solid [Parasolid Technical Support Group 2001a]. They are interests only the relation between themselves and other entities. There are nine types of topological entity in a single solid part. These include body, region, shell, face, loop, edge, fin, and vertex.

Geometrical entities describe shape of solid using mathematical equation. There are three types of geometrical entity. These include surface, curve, and point.

### 3. Feature-Based Modeling

There are many ways to creates and modifies solid. Feature-based modeling is the one that chosen by many

recent applications. Conceptual of feature-based is to build or manipulate solid by features.

Features are elements used in generating, analyzing, or evaluating designs [Shah 1991]. They can be classified according to their definitions. One distinction can be made on the basis of the feature's purpose, as by Shah and Rogers [1988]:

- Form Features: elements related to nominal geometry.
- Precision Features: acceptable deviations from nominal form or size.
- Technological Features: related to performance and operation.
- Material Features: material composition, treatment, condition.
- Assembly Features: assembly sequence, orientation relative to the part.

Or by van Emmerik [1990]:

- Form Features.
- Pattern Features: regular pattern of similar entities.
- Connection Features: geometric constraints.
- Property Features: properties not related to explicit geometry.
- Application Features: related to process planning requirements.

In this development, form features are used to create and modify solid models.

**Extrusion**

Extrusion is form feature which creates a body by linear extrusion of profile body. The profile body may be a minimum body, a wire body or a sheet body. These profiles create extruded wire bodies, sheet bodies or solid bodies respectively. The start and end of the extrusion can be specified in a number of ways; in particular by intersection with other bodies. The direction of extrusion is defined by a unit vector. The start and end of the extrusion (the bounds) must be ordered along this vector so that the start comes first.

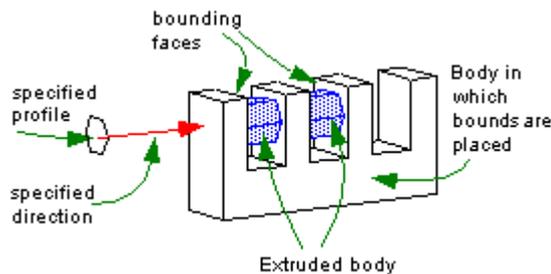


Fig. 1 Extrusion

**Revolution**

Revolution creates a body by revolves profile body about spin axis. Like extrusion, the profile body may be a minimum body, a wire body or a sheet body, and produce revolved wire bodies, sheet bodies or solid bodies

respectively. The bounds must be ordered along spin direction which defined by the right hand rule about spin axis.

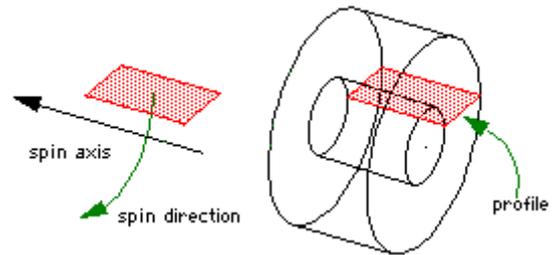


Fig. 2 Revolution

**Swept Along Guide**

Swept along guide is the creation of a body by sweeps profile body along guide. The guide may be a wire body, edge, or chain edge. The profile body may be a minimum body, a wire body or a sheet body. These profiles create swept wire bodies, sheet bodies or solid bodies respectively.

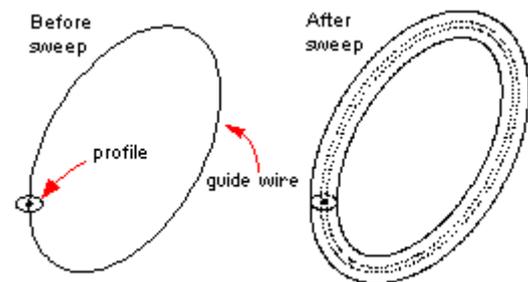


Fig. 3 Swept Along Guide

**Loft**

Loft creates a body using set of profiles which define cross sections of body. The profiles may be wire bodies, sheet bodies or combination of minimum bodies, wire bodies, and sheet bodies. These profiles create loft sheet bodies or solid bodies according to set of profiles. If each profile has a different number of vertices, then vertex mapping is required to map vertices of each profile to vertices of the next profile. If each profile has the same number of vertices then only start vertex of each profile required. In additional, loft feature can use guide to control deviation and twisted of cross section between profiles.

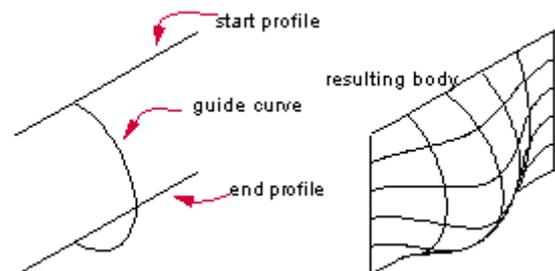


Fig. 4 Loft

## Fillet and Chamfer

These two features, sometime refer to as “*Edge Blend*” fillets, have a circular cross section. They are the result of rolling a sphere along the two surfaces adjoining the edge. The extent of fillets is determined by specifying the radius of the sphere. Chamfers are blends which have a linear cross section. In general, chamfers are not tangent continuous with the faces adjoining the original edge. The extent of chamfer blends are determined by the offset surfaces being intersected; the resulting curves being projected onto the underlying surface, between which, the chamfer blends are constructed.

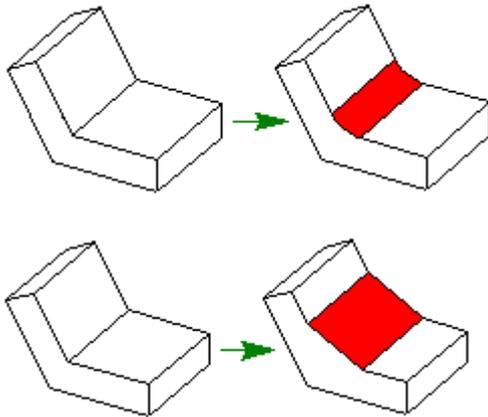


Fig 5. Fillet and Chamfer

## Hollow

Hollow is feature that modifies a solid body to hollow body by specifying pierce faces, offset faces, and offset distance.

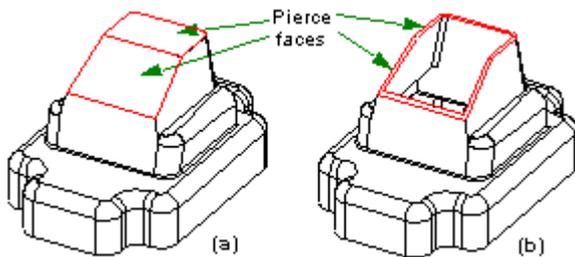


Fig. 6 Hollow

## 4. Boolean Operations Of Solids

Boolean operations are operations which are performed on volumes of solids to produce a new solid. The Boolean operations most commonly used in solid modeling are union, subtraction and intersection.

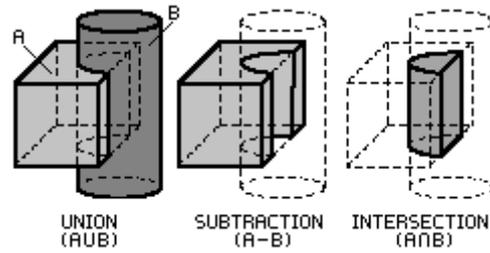


Fig. 7 Boolean Operation

In this development, combinations of form features with Boolean operations are used to produce solid creation commands. These include:

- Extrusion Command: uses extrusion feature with union.
- Extrusion Cutout Command: uses extrusion feature with subtraction.
- Revolution command: uses revolution feature with union.
- Revolution Cutout Command: uses revolution feature with subtraction.
- Swept Command: uses sweep along guide feature with union.
- Swept Cutout Command: uses sweep along guide feature with subtraction.
- Loft Command: uses loft feature with union.
- Loft Cutout Command: uses loft feature with subtraction.

## 5. Parasolid Kernel

Parasolid is an exact B-rep modeler. It can be used to build and manipulate solid objects, output objects in various ways, store objects in files on disk and retrieve them later, calculate mass and moments of inertia, and perform clash detection [Parasolid Technical Support Group 2001a].

### Interfaces to Parasolid

There are several interfaces between Parasolid and application. The PK Interface and the Kernel Interface sit on top of the modeler, and are the means by which application models and manipulates objects and controls the functioning of the modeler. The downward interfaces lie beneath the modeler, and are called by the modeler when it needs to perform data-intensive or system type operations. There are up to three downward interfaces. These include Frustrum, Graphical Output (GO) and Foreign Geometry [Parasolid Technical Support Group 2001a].

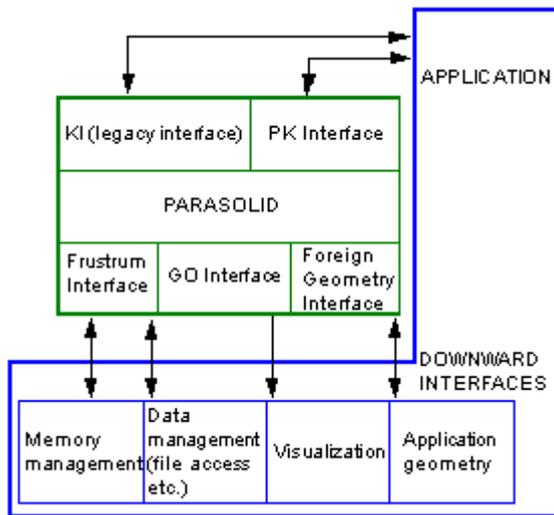


Fig. 8 Parasolid Structure

### Parasolid XT Format

Parasolid XT format contains formats in which Parasolid uses to represent model information in external files. Parasolid can encode the data it writes out in three different formats [Parasolid Technical Support Group 2001b]:

1. Text (usually ASCII)
2. Neutral binary
3. Bare binary (this is not recommended)

In text format all the data is written out as human readable text, they have the advantage that they are readable but they also have a number of disadvantages. They are relatively slow to read and write, converting to and from text forms of real numbers introduces rounding errors that can (in extreme cases) cause problems and finally there are limitations when dealing with multi-byte character sets. Carriage return or line feed characters can appear anywhere in a text transmit file but other unexpected non-printing characters will cause Parasolid to reject the file as corrupt.

Neutral binary is a machine independent binary format.

Bare binary is a machine dependent binary format. It is not a recommended format since the machine type which wrote it must be known before it can be interpreted.

## 6. The CU-Solid

The CU-Solid is CAD software, which is created in this development. It can build and manipulate solids with feature-based modeling concept using Parasolid kernel as the modeler. The CU-Solid has many small software components. Each of components has an individual duty. These components are:

### Sketch Tool

Because the CU-Solid uses conceptual of feature-based solid modeling, so, the profile bodies must be supplied for some features. Sketch tool designed for this opportunity. To create profile bodies, sketch tool is used to draw profile curves on sketch planes. Profile curves, which can be closed-curves or opened-curves, must convert to profile bodies before supply to solid features. Opened-curves

produce wire bodies and closed-curves produce either wire bodies or sheet bodies. Many types of profile curve can be created by CU-Solid. These include line, circle, arc, ellipse, rectangle and parallelogram.

### Data storage

CU-Solid can store and retrieve two external file formats. First one is the standard Parasolid XT format. This format has "xmt\_txt" or "xmt\_bin" extension and it can be use with other commercial packages which use Parasolid as the modeler. The other format is Parasolid XT format with additional attributes. This format has "part" extension and it was developed for store additional model data such as model construction, feature parameters, custom-feature definition, etc. This additional data made this kind of format can modify features and model constructions after restore data.

### 3-D visualization

To output solid models in 3-D visualization objects, Graphical Output (GO) functions must be provided by CU-Solid. These GO functions use "OpenGL" to render solid models, and Parasolid uses them as callback functions.

OpenGL is a software interface to graphics hardware [Woo, Neider, Davis, and Shreiner 2000]. It provides API functions to create and display 3-D visualization objects, and manipulate 3-D visualization environment. To render solid models, Parasolid extract solid's components into small pieces. If the components are faces or surfaces then facets are extracted. If the components are edges or curves then linear segments are extracted. Facets and linear segments contain vertices data. A facet (triangular facet) has three vertices data and a linear segment has two vertices data. OpenGL use these vertices data to create simple 3-D primitives and then combine these simple 3-D primitives into a complex 3-D visualization object.

### Plug-in

CU-Solid can be use with additional modules called "Plug-in". Plug-in made the CU-Solid more flexible used because the user can create custom command to build or manipulate solid models. User can create plug-in with the "CU-Solid Add-in AppWizard", which come with CU-Solid software package. CU-Solid Add-in AppWizard generates a Microsoft Visual C++ project. This project help user to access shared data in CU-Solid via internal interface of CU-Solid. Furthermore, this project can access directly to Parasolid to make more flexible programming plug-in. The Gem Stone Design and Manufacturing module is an example of plug-in module that help us for designing final form of Gem stones with output parameters to use for controlling the in-house developed faceting machine.

## 7. CU Faceting Machine

Fig. 9 is the in-house developed faceting machine. It is a automatic faceting machine controlled by a microcomputer. The diagram, shown in Fig.10, demonstrates the control strategy of the system. The

machine consists of 6 axes but only 4 axes needed high accuracy feedback control. The other 2 axes are for rotating grinding wheels. The 4 axes are (see Fig. 11): the first axis is for moving a robotic arm up and down (h), the second axis is for controlling a drop stick grinding angle (e), the third axis is for controlling a stone attachment angle or an index angle (i), and the fourth axis for exchanging grinding wheels during faceting process. The other 2 axes: the fifth axis is for generating oscillation of the grinding wheels and the sixth axis is spindling axis of the grinding wheels.

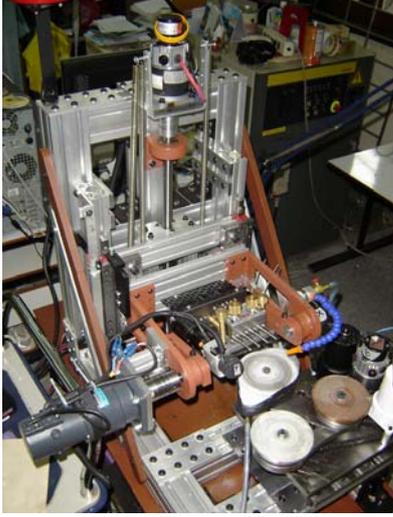


Fig. 9 CU Faceting Machine

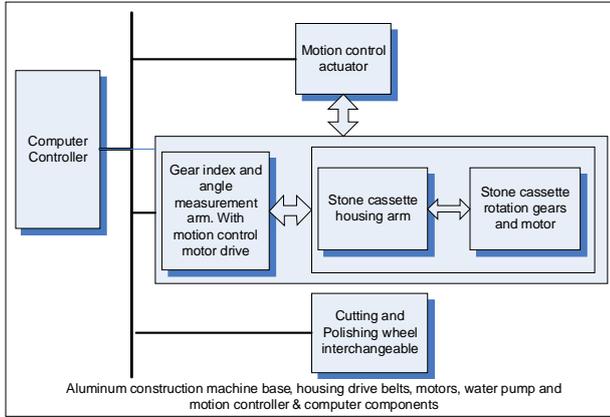


Fig. 10 Principal of the controller used in the faceting machine

## 8. GEMCUT NC File Module

GEMCUT is an application module developed based on CU-Solid. This module is to find parameters used in the CU Faceting Machine from 3D model data created in the CU-Solid. These parameters are a elevation angle (e), an index angle (i) and a depth of cut (h). With these parameters, we can calculate planes used for the faceting machine as following:

The normal vector:  $\bar{\mathbf{n}}$   
For the Crown side

$$\bar{\mathbf{n}} = \begin{bmatrix} n_x \bar{\mathbf{x}} \\ n_y \bar{\mathbf{y}} \\ n_z \bar{\mathbf{z}} \end{bmatrix} = \begin{bmatrix} \cos(i) \sin(e) \bar{\mathbf{x}} \\ \sin(i) \sin(e) \bar{\mathbf{y}} \\ \cos(e) \bar{\mathbf{z}} \end{bmatrix} \quad (1)$$

For the Pavilion side

$$\bar{\mathbf{n}} = \begin{bmatrix} n_x \bar{\mathbf{x}} \\ n_y \bar{\mathbf{y}} \\ n_z \bar{\mathbf{z}} \end{bmatrix} = \begin{bmatrix} -\cos(i) \sin(e) \bar{\mathbf{x}} \\ \sin(i) \sin(e) \bar{\mathbf{y}} \\ -\cos(e) \bar{\mathbf{z}} \end{bmatrix} \quad (2)$$

Reference Direction:  $\bar{\mathbf{r}}$

For the Crown side

$$\bar{\mathbf{r}} = \begin{bmatrix} r_x \bar{\mathbf{x}} \\ r_y \bar{\mathbf{y}} \\ r_z \bar{\mathbf{z}} \end{bmatrix} = \begin{bmatrix} -\sin(i) \bar{\mathbf{x}} \\ \cos(i) \bar{\mathbf{y}} \\ 0.0 \bar{\mathbf{z}} \end{bmatrix} \quad (3)$$

For the Pavilion side

$$\bar{\mathbf{r}} = \begin{bmatrix} r_x \bar{\mathbf{x}} \\ r_y \bar{\mathbf{y}} \\ r_z \bar{\mathbf{z}} \end{bmatrix} = \begin{bmatrix} \sin(i) \bar{\mathbf{x}} \\ \cos(i) \bar{\mathbf{y}} \\ 0.0 \bar{\mathbf{z}} \end{bmatrix} \quad (4)$$

Points on a plane:  $\mathbf{p}$

$$\mathbf{p} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} n_x h \\ n_y h \\ n_z h \end{bmatrix} \quad (5)$$

From the equations (1) – (5) and a 3-D model data, we can calculate parameters used in the CU Faceting Machine as following:

Crown side:

For z-axis (up-down)

$$n_z = \cos(e) \quad (6a)$$

so, we will have

$$e = \cos^{-1}(n_z), \quad 0 \leq e \leq \pi \quad (7a)$$

For X and Y axes:

$$n_x = \cos(i) \sin(e) \quad (8a)$$

$$n_y = \sin(i) \sin(e) \quad (9a)$$

so, we will have

$$\frac{n_y}{n_x} = \tan(i)$$

$$\text{or, } i = \tan^{-1}\left(\frac{n_y}{n_x}\right), \quad 0 \leq i \leq 2\pi \quad (10a)$$

Pavilion side:

$$n_z = -\cos(e) \quad (6b)$$

$$\text{so, } e = \cos^{-1}(-n_z), \quad 0 \leq e \leq \pi \quad (7b)$$

For X and Y axes:

$$n_x = -\cos(i) \sin(e) \quad (8b)$$

$$n_y = \sin(i) \sin(e) \quad (9b)$$

so, we will have

$$\frac{n_y}{n_x} = -\tan(i)$$

$$\text{or } i = \tan^{-1}\left(-\frac{n_y}{n_x}\right), 0 \leq i \leq 2\pi \quad (10b)$$

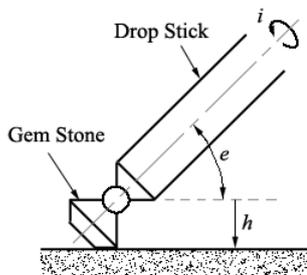
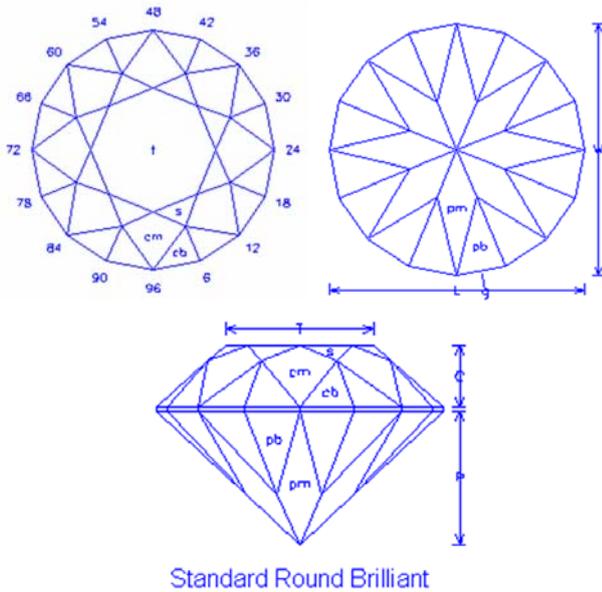


Fig. 11 Parameters needed for the faceting machine

Pavilion		
Facet	Angle	Index
Pb	45	03-09-15-21-27-33-39-45-51-57-63-69-75-81-87-93
Pm	43	96-12-24-36-48-60-72-84
Crown		
Facet	Angle	Index
Cb	47	03-09-15-21-27-33-39-45-51-57-63-69-75-81-87-93
Cm	42	96-12-24-36-48-60-72-84
S	27	06-18-30-42-54-66-78-90
T	0	Table

Table 1 Parameters needed for the gem stone shown in Fig. 11.

Example: A standard gem stone form shown in Fig. 11 (instead of creating by using CU-Solid) can be faceted from the parameters shown in the Table 1. And Fig. 12 shows the gem stones faceted from the parameters in Table 1 and the CU Faceting Machine.



Fig. 12. The samples of the gems from the parameters shown in Table 1.

## 9. Conclusion

This work is to develop a 3-D solid modeling software, the CU-Solid with GEMCUT application module. This software was created with feature-based modeling concept. It can use for building and manipulating solids with various commands. These include extrusion, extrusion cutout, revolution, revolution cutout, swept, swept cutout, loft, loft cutout, fillet, chamfer and hollow. To store solid data into external files and retrieve them later, the CU-Solid uses standard Parasolid XT format and Parasolid XT format with additional attributes. The standard format can be used with commercial packages, Unigraphics, SolidEdge, and SolidWorks, and the CU-Solid can import and export these commercial packages data without any difficulty. The format with additional attributes can be used with the CU-Solid only, and users can modify the model construction later while the standard format can not.

Fig. 13 – Fig. 15 show the examples of part creation with various viewing display features using the CU-Solid. Fig. 16 shows the gem stone 3D modeling step. With 3D model data, the parameters used with the CU Faceting Machine can be calculated as shown in Fig 17. So, instead of using standard gem stone forms (as shown in Table 1, we can use CU-Solid with GEMCUT application module to design various form of gem stones. These will help a gem stone designer uses output stones for studying the reflection of light in gem stones more convenience.

Fig 18 shows the manufacturing features of the CU-Solid which is not cover in this paper. The NURB surface is the key feature for the manufacturing module.

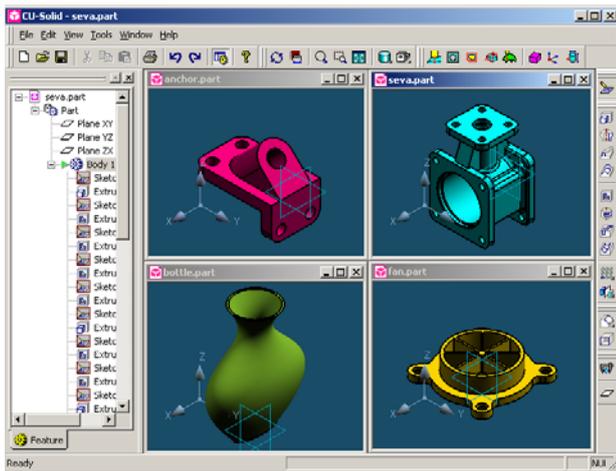


Fig. 13 CU-Solid Screen displays multiple parts

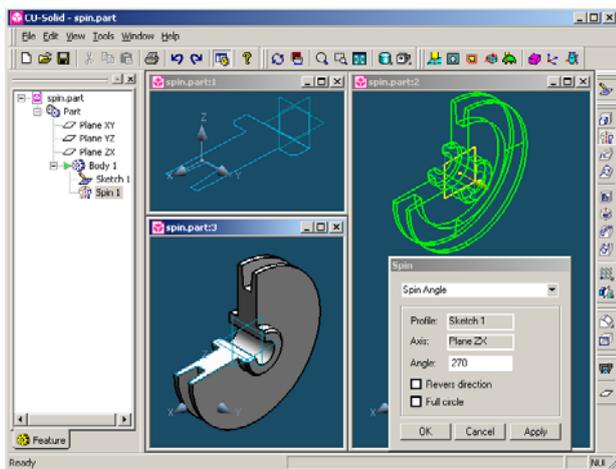


Fig. 14 An example of CU-Solid View

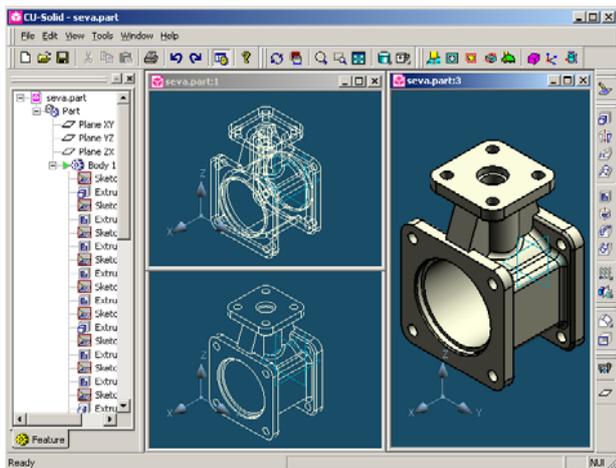


Fig. 15 An example of CU-Solid Part Modeling

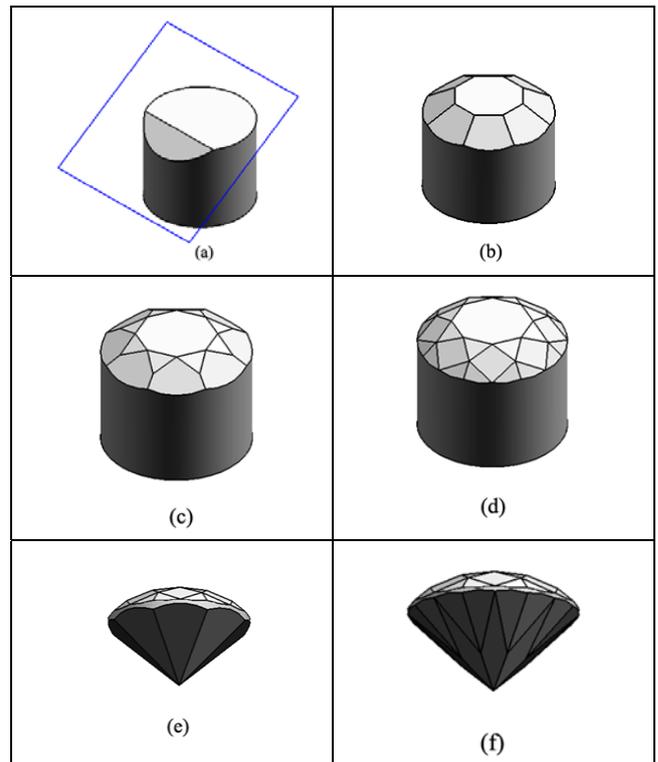


Fig. 16 Gem stone modeling step

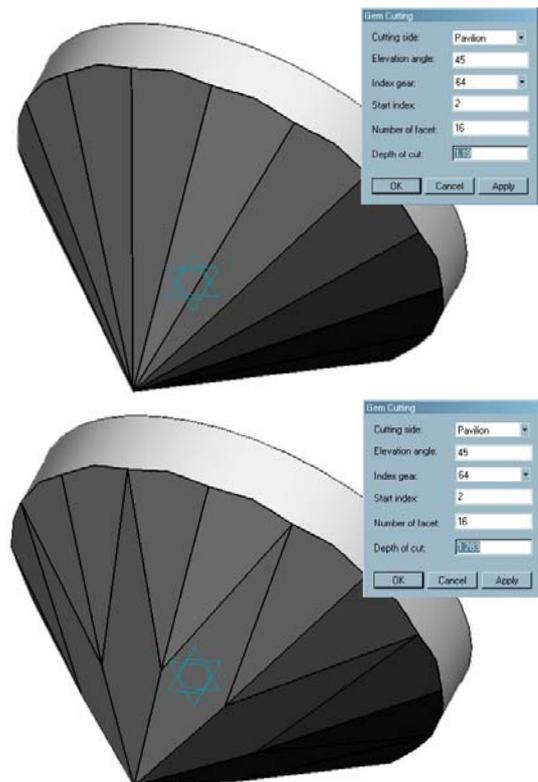


Fig. 17 The parameters obtained for the 3D-model directly

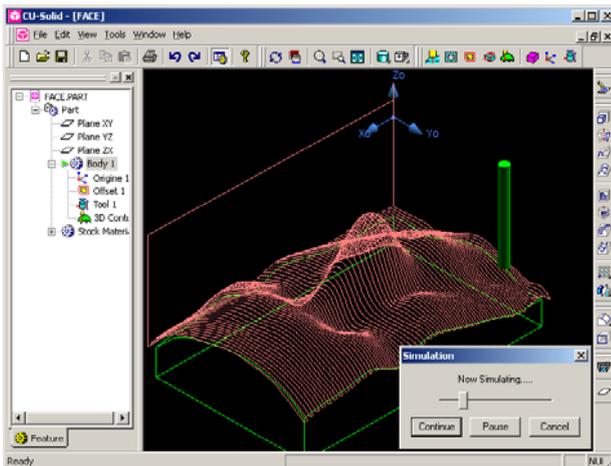


Fig. 18 An example of CU-Solid Manufacturing Features

## 10. References

- [1] van Emmerik, M.J.G.M. 1990. Interactive design of parameterized 3D models by direct manipulation. Ph.D. thesis. Delft: Delft University Press.
- [2] Parasolid Technical Support Group. 2001a. Parasolid V13.2 Functional Description. [Electronic book]. pp. 49-117. Cambridge: Unigraphics Solutions Inc.
- [3] Parasolid Technical Support Group. 2001b. Parasolid XT Format. [Electronic book]. p.2. Cambridge: Unigraphics Solutions Inc.
- [4] Shah, J.J. & Rogers, M.T. 1988. "Functional requirements and conceptual design of the Feature-Based Modelling System". Computer-Aided Engineering Journal. vol. 5 no. 1. pp. 9-15.
- [5] Shah, J.J. 1991. "Assessment of features technology". Computer-Aided Design. vol. 23 no. 5. pp. 331-343.
- [6] Woo, M., Neider, J., Davis, T., and Shreiner, D. 2000. OpenGL programming guide. 3rd ed. Delhi: Addison Wesley.