

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800

Open access books available

122,000

International authors and editors

135M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.

For more information visit www.intechopen.com



A Review on Shape Engineering and Design Parameterization in Reverse Engineering

Kuang-Hua Chang
The University of Oklahoma
Norman, OK
USA

1. Introduction

3D scanning technology has made enormous progress in the past 25 years (Blais, 2004); especially, the non-contact optical surface digitizers. These scanners or digitizers become more portable, affordable; and yet capturing points faster and more accurately. A hand-held laser scanner captures tens of thousands points per second with a level of accuracy around 40 μm , and can cost as low as fifty thousand dollars, such as *ZScanner 800* (ZCorp). Such technical advancement makes the scanners become largely accepted and widely used in industry and academia for a broad range of engineering assignments. As a result, demand on geometric modeling technology and software tools that support efficiently processing large amount of data points (scattered points acquired from a 3D scanning, also called point cloud) and converting them into useful forms, such as NURB (non-uniform rational B-spline) surfaces, become increasingly higher.

Auto surfacing technology that automatically converts point clouds into NURB surface models has been developed and implemented into commercial tools, such as *Geomagic* (Geomagic), *Rapidform* (INUS Technology, Inc.), *PolyWorks* (innovMetric), *SolidWorks/Scan to 3D* (SolidWorks, Inc.), among many others. These software tools have been routinely employed to create NURB surface models with excellent accuracy, saving significant time and effort. The NURB surface models are furnished with geometric information that is sufficient to support certain types of engineering assignments in maintenance, repair, and overhaul (MRO) industry, such as part inspection and fixture calibration. The surface models support 3D modeling for bioengineering and medical applications, such as (Chang et al., 2003; Sun et al., 2002; Liu et al., 2010; Lv et al., 2009). They also support automotive industry and aerospace design (Raja & Fernades 2008). NURB surface models converted from point clouds have made tremendous contributions to wide range of engineering applications. However, these models contain only surface patches without the additional semantics and topology inherent in feature-based parametric representation. Therefore, they are not suitable for design changes, feature-based NC toolpath generations, and technical data package preparation. Part re-engineering that involves design changes also requires parametric solid models.

On the other hand, shape engineering and design parameterization aims at creating fully parametric solid models from scanned data points and exporting them into mainstream

CAD packages that support part re-engineering, feature-based NC toolpath generations, and technical data package preparation. Although, converting data points into NURB surface models has been automated, creating parametric solid models from data points cannot and will not be fully automated. This is because that, despite technical challenges in implementation, the original design intent embedded in the data points must be recovered and realized in the parametric solid model. Modeling decisions have to be made by the designer in order to recover the original design intents. However, designers must be relieved from dealing with tedious point data manipulations and primitive geometric entity constructions. Therefore, the ideal scenario is having software tools that take care of labor intensive tasks, such as managing point cloud, triangulation, etc., in an automated fashion; and offer adequate capabilities to allow designers to interactively recover design intents. Such an ideal scenario has been investigated for many years. After these many years, what can be done with the technology and tools developed at this point? Many technical articles already address auto surfacing. In this chapter, in addition to auto surfacing, we will focus on solid modeling and design parameterization.

We will present a brief review and technical advancement in 3D shape engineering and design parameterization for reverse engineering, in which discrete point clouds are converted into feature-based parametric solid models. Numerous efforts have been devoted to developing technology that automatically creates NURB surface models from point clouds. Only very recently, the development was extended to support parametric solid modeling that allows significant expansion on the scope of engineering assignments. In this chapter, underlying technology that enables such advancement in 3D shape engineering and design parameterization is presented. Major commercial software that offers such capabilities is evaluated using practical examples. Observations are presented to conclude this study. Next, we will present a more precise discussion on design parameterization to set the tone for later discussion in this chapter.

2. Design parameterization

One of the common approaches for searching for design alternatives is to vary the part size or shape of the mechanical system. In order to vary part size or shape for exploring better design alternatives, the parts and assembly must be adequately parameterized to capture design intents.

At the parts level, design parameterization implies creating solid features and relating dimensions so that when a dimension value is changed the part can be rebuilt properly and the rebuilt part revealed design intents. At the assembly level, design parameterization involves defining assembly mates and relating dimensions across parts. When an assembly is fully parameterized, a change in dimension value can be automatically propagated to all parts affected. Parts affected must be rebuilt successfully; and at the same time, they will have to maintain proper position and orientation with respect to one another without violating any assembly mates or revealing part penetration or excessive gaps. For example, in a single-piston engine shown in Fig. 1 (Silva & Chang, 2002), a change in the bore diameter of the engine case will alter not only the geometry of the case itself, but also all other parts affected, such as piston, piston sleeve, and even crankshaft. Moreover, they all have to be rebuilt properly and the entire assembly must stay intact through assembly mates, and faithfully reveal design intents.

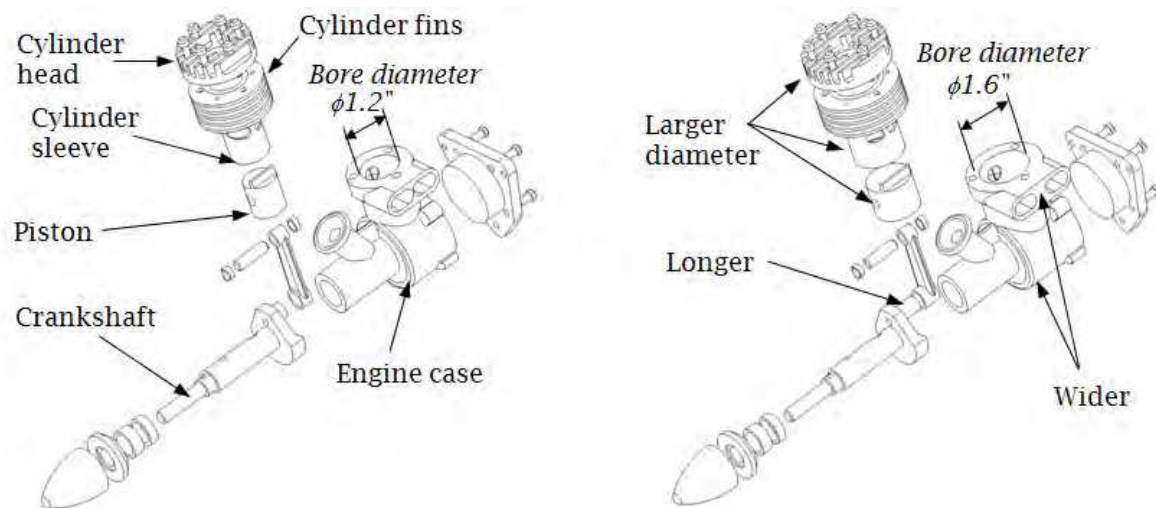


Fig. 1. A single-piston engine—exploded view, (a) bore diameter 1.2", and (b) bore diameter 1.6"

3. Shape engineering

The overall process of shape engineering and parametric solid modeling is shown in Fig. 2, in which four main phases are involved. They are (1) triangulation that converts data points to a polygon mesh, (2) mesh segmentation that separates a polygon mesh into regions based on the characteristics of the surface geometry they respectively represent, (3) solid modeling that converts segmented regions into parametric solid models, and (4) model translation that exports solid models constructed to mainstream CAD systems. Note that it is desired to have the entire process fully automated; except for Phase 3. This is because that, as stated earlier, Phase 3 requires designer's interaction mainly to recover original design intents. These four phases are briefly discussed in the following subsections.

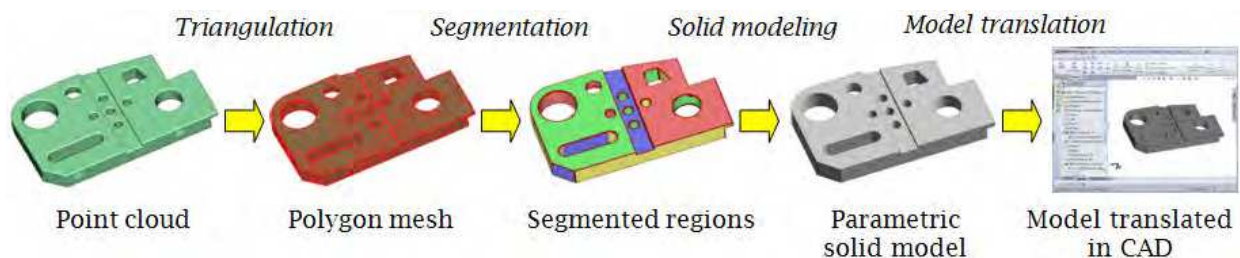


Fig. 2. General process of shape engineering and parametric solid model construction

3.1 Triangulation

The mathematic theory and computational algorithms for triangulation have been well developed in the past few decades. A polygon mesh can be automatically and efficiently created for a given set of data points. The fundamental concept in triangulation is Delaunay triangulation. In addition to Delaunay triangulation, there are several well-known mathematic algorithms for triangulation, including marching cubes (Lorensen et al., 1987), alpha shapes (Edelsbrunner et al., 1983), ball pivoting algorithm (BPA) (Bernardini et al., 1999), Poisson surface reconstruction (Kazhdan et al., 2006), moving least squares (Cuccuru et al., 2009), etc. A few high profile projects yield very good results, such as sections of Michelangelo's Florentine

Pietà composed of 14M triangle mesh generated from more than 700 scans (Bernardini et al., 1999), reconstruction of “Pisa Cathedral” (Pisa, Italy) from laser scans with over 154M samples (Cuccuru et al., 2009), and head and cerebral structures (hidden) extracted from 150 MRI slices using the marching cubes algorithm (about 150,000 triangles), as shown in Fig. 3.



Fig. 3. Sample projects of scanning and triangulation, (a) Michelangelo’s Florentine Pietà, (b) Pisa Cathedral, and (c) head and cerebral structures

Although many triangulation algorithms exist, they are not all fool-proof. They tend to generate meshes with a high triangle count. In addition, these algorithms implicitly assume topology of the shape to be reconstructed from triangulation, and the parameter settings often influences results and stability. A few mesh postprocessing algorithms, such as decimation (for examples, Schroeder, 1997; Hoppe et al., 1993), and mesh smoothness (e.g., Hansen et al., 2005; Li et al., 2009), are worthwhile mentioning for interested readers.

3.2 Segmentation

One of the most important steps in shape engineering is mesh segmentation. Segmentation groups the original data points or mesh into subsets each of which logically belongs to a single primitive surface.

In general, segmentation is a complex process. Often iterative region growing techniques are applied (Besl & Jain, 1988; Alrashdan et al., 2000; Huang & Meng, 2001). Some use non-iterative methods, called direct segmentation (Várady et al., 1998), that are more efficient. In general, the segmentation process, such as (Vanco & Brunnett, 2004) involves a fast algorithm for k -nearest neighbors search and an estimate of first- and second-order surface properties. The first-order segmentation, which is based on normal vectors, provides an initial subdivision of the surface and detects sharp edges as well as flat or highly curved areas. The second-order segmentation subdivides the surface according to principal curvatures and provides a sufficient foundation for the classification of simple algebraic surfaces. The result of the mesh segmentation is subject to several important parameters, such as the k value (number of neighboring points chosen for estimating surface properties), and prescribed differences in the normal vectors and curvatures (also called sensitivity thresholds) that group the data points or mesh. As an example shown in Fig. 4a, a high sensitive threshold leads to scattered regions of small sizes, and a lower sensitive threshold tends to generate segmented regions that closely resemble the topology of the object, as illustrated in Fig. 4b.

Most of the segmentation algorithms come with surface fitting, which fits a best primitive surface of appropriate type to each segmented region. It is important to specify a hierarchy of surface types in the order of geometric complexity, similar to that of Fig. 5 (Várady et al., 1997). In general, objects are bounded by relatively large primary (or functional) surfaces.

The primary surfaces may meet each other along sharp edges or there may be secondary or blending surfaces which may provide smooth transitions between them.



Fig. 4. Example of mesh segmentation, (a) an object segmented into many small regions due to a high sensitivity threshold, and (b) regions determined with a low sensitivity threshold

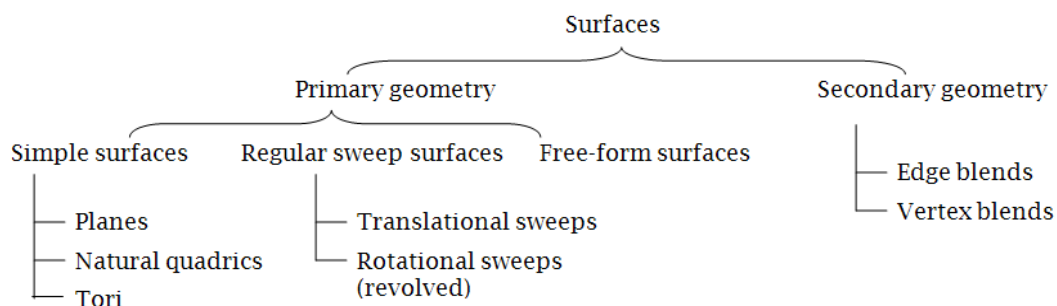


Fig. 5. A hierarchy of surfaces

As discussed above, feature-based segmentation provides a sufficient foundation for the classification of simple algebraic surfaces. Algebraic surfaces, such as planes, natural quadrics (such as sphere, cylinders, and cones), and tori, are readily to be fitted to such regions. Several methods, including (Marshall et al., 2004), have been proposed to support such fitting, using least square fitting.

In addition to primitive algebraic surfaces, more general surfaces with a simple kinematic generation, such as sweep surfaces, revolved surfaces (rotation sweep), extrusion surfaces (translation sweep), pipe surfaces, are directly compatible to CAD models. Fitting those surfaces to segmented data points or mesh is critical to the reconstruction of surface models and support of parameterization (Lukács et al., 1998).

In some applications, not all segmented regions can be fitted with primitives or CAD-compatible surfaces within prescribed error margin. Those remaining regions are classified as freeform surfaces, where no geometric or topological regularity can be recognized. These can be a collection of patches or possibly trimmed patches. They are often fitted with NURB surfaces. Many algorithms and methods have been proposed to support NURB surface fitting, such as (Tsai et al., 2009).

3.3 Solid modeling

Solid modeling is probably the least developed in the shape engineering process in support of reverse engineering. Boundary representation (B-rep) and feature-based are the two basic representations for solid models. There have been some methods, such as (Várady et al., 1998), proposed to automatically construct B-rep models from point clouds or triangular mesh. Some focused on manufacturing feature recognition for process planning purpose,

such as (Thompson, 1999). One promising development in recent years was the geometric feature recognition (*GFR*), which automatically recognizes solid features embedded in the B-rep models. However, none of the method is able to fully automate the construction process and generate fully parametric solid models. Some level of manual work is expected.

3.3.1 Boundary representation

Based on segmented regions (with fitted surfaces), a region adjacent graph is built. This graph reflects the complete topology and serves as the basis for building the final B-rep model, also called stitched models, where the individual bounded surfaces are glued together along their common edges to form an air-tight surface model.

In general, there are three steps involved in constructing B-rep models, flattening, edges and vertices calculations, and stitching (Várady et al., 1998). In flattening step, regions are extended outwards until all triangles have been classified. Note that this step is necessary to remove all gaps between regions. Sharp edges can be calculated using surface-surface intersection routines, and vertices where three surfaces meet are also determined. During the process, a complete B-rep topology tree is also constructed. A B-rep model can then be created by stitching together the faces, edges, and vertices. This operation is commonly supported by most solid modeling kernels.

3.3.2 Solid feature recognition

B-rep models are not feature-based. In order to convert a B-rep model into a feature-based solid model, the embedded solid features must be recognized, and a feature tree that describes the sequence of feature creation must be created.

One of the most successful algorithms for geometric feature recognition has been proposed by (Venkataraman et al., 2001). The algorithm uses a simple four step process, (1) simplify imported faces, (2) analyze faces for specific feature geometry, (3) remove recognized feature and update model; and (4) return to Step 2 until all features are recognized. The process is illustrated in Fig. 6. Once all possible features are recognized, they are mapped to a new solid model of the part (Fig. 6d) that is parametric with a feature tree. This feature tree defines the feature regeneration (or model rebuild) sequence.

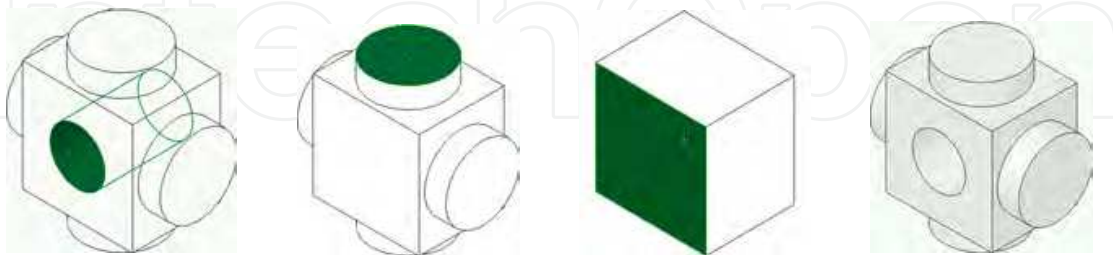


Fig. 6. Illustration of *GFR* algorithm, (a) imported surface model with hole surface selected, (b) hole recognized and removed, extruded face of cylinder selected, (c) cylindrical extrusions recognized, base block extrusion face selected, and (d) all features recognized and mapped to solid model

Venkataraman's method was recently commercialized by Geometric Software Solutions, Ltd. (GSSL), and implemented in a number of CAD packages, including *SolidWorks* and

CATIA, capable of recognizing basic features, such as extrude, revolve, and more recently, sweep. This capability has been applied primarily for support of solid model interchanges between CAD packages with some success, in which not only geometric entities (as has been done by IGES—Initial Graphics Exchange Standards) but also parametric features are translated.

One of the major issues revealed in commercial *GFR* software is design intent recovery. For example, the flange of an airplane tubing would be recognized as a single revolve feature, where a profile sketch is revolved about an axis (Fig. 7a). However, current *GFR* implementations are not flexible. As shown in Fig. 7b, without adequate user interaction, the single sketch flange may be recognized as four or more separate features. While the final solid parts are physically the same, their defining parameters are not. Such a batch mode implementation may not be desirable in recovering meaningful design intents.

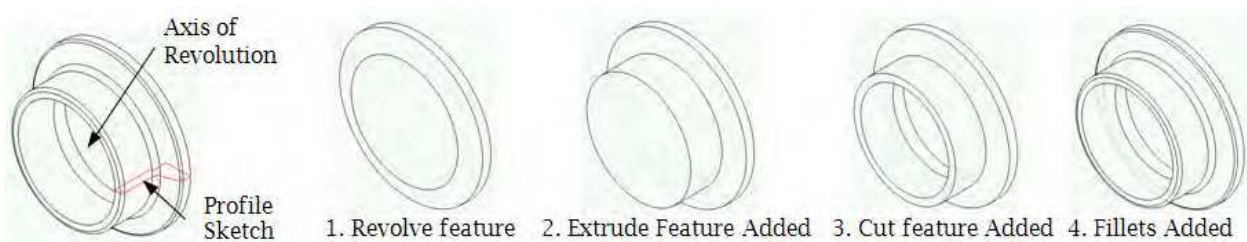


Fig. 7. Feature recognition for airplane tubing flange, (a) single revolved feature, and (b) four features: revolve, extrude, cut, and fillet

3.3.3 Design parameterization

A feature-based parametric solid model consists of two key elements: a feature tree, and fully parameterized sketches employed for protruding solid features. A fully parameterized sketch implies that the sketch profile is fully constrained and dimensioned, so that a change in dimension value yields a rebuilt in accordance with design intents as anticipated. To the author's knowledge, there is no such method proposed or offered that fully automates the process. Some capabilities are offered by commercial tools, such as *Rapidform*, that support designers to interactively create fully parameterized sketches, which accurately conform to the data points and greatly facilitates the solid modeling effort.

3.4 Solid model export

Since most of the promising shape engineering capabilities are not offered in CAD packages (more details in the next section), the solid models constructed in reverse engineering software will have to be exported to mainstream CAD packages in order to support common engineering assignments. The conventional solid model exchanges via standards, such as IGES or STEP AP (application protocols), are inadequate since parametric information, including solid features, feature tree, sketch constraints and dimensions, are completely lost through the exchanges. Although feature recognition capability offers some relief in recognizing geometric features embedded in B-rep models, it is still an additional step that is often labor intensive. Direct solid model export has been offered in some software, such as *liveTransfer*TM module of *Rapidform XOR3* as well as third party software, such as *TransMagic*. More will be discussed for *liveTransfer*TM.

4. Engineering software evaluations

The key criteria for the software evaluations are the capabilities of supporting automatic surface construction from point clouds and parametric solid modeling. We did the first screening on nine leading software tools that are commercially available. This screening was carried out based on the information provided in product brochure, technical reports (for example, Siddique, 2002; Chang et al., 2006), thesis (for example, Gibson, 2004), company web sites, on-line software demo, case study reports, etc. After the screening, we acquired four tools and conducted hands-on evaluations, using five industrial examples. With this, we are able to identify pros and cons in each software tool, make a few observations, and conclude the study.

4.1 Software screening

After extensive research and development in the past decade, software tools for reverse engineering have made impressive advancement. In general, these tools can be categorized into two groups, feature-based and RE-based. The feature-based CAD packages, such as *Pro/ENGINEER*, *SolidWorks*, and *CATIA*, emphasize recovering the original design intents of the parts. Following standard CAD capabilities, such as sketching, extrusion, and Boolean operations, designers are able to create parts with design intents recovered. On the contrary, RE-based packages; such as *Geomagic*, *Rapidform*, and *Paraform*, focus on reconstructing the geometry of the objects from scanned data, usually in the form of NURB surfaces. RE-based packages offer excellent capabilities in editing points, creating meshes, and generating NURB surfaces. In addition, the display performance of mass data offered by the RE-based package is far better than the feature-based CAD software; that is, in the context of reverse engineering.

In this study, we looked for two key engineering capabilities; i.e., surface construction and parametric solid modeling from a point cloud or a polygon mesh. All feature-based and RE-based software tools offer some capabilities for surface constructions. However, manually constructing curves and surfaces from point clouds or polygon meshes are tedious and extremely time consuming. It is critical that a serious RE software must offer auto surfacing; i.e., allowing for creating air-tight, high accuracy, and high quality surface models with only a few button clicks. On the other hand, constructing solid models has to be carried out in an interactive manner, allowing designers to recover original design intents. Software must offer adequate capabilities to assist designers to sketch section profiles and create solid features efficiently, without directly dealing with point clouds or polygon meshes. Certainly, the software will have to be stable and capable of handling massive data. Millions of point data need huge computer resources to process. Zoom, pan or rotate the object, for example, on the screen may take time for software to respond. Speed is the key for modern RE-based software. We are essentially searching for software that offers auto surfacing and parametric modeling capabilities with fast and stable performance.

In addition, several software related criteria are defined, as listed in Table 1. These criteria are categorized into four groups, (1) general capabilities, such as speed; (2) generation of NURB models, including auto surfacing and geometric entity editing capabilities; (3) generation of solid models, including section profiling and parametric capabilities; and (4) usability.

From Table 1, we observe that most surveyed software offers basic capabilities for editing and manipulating points, polygon meshes and NURB curves and surfaces. Particularly, we found both *Geomagic* and *Rapidform* support auto surfacing. Solid modeling using scanned

data can be commonly achieved by creating section sketches from polygon meshes and following feature creating steps similar to CAD packages. Based on the survey, *Rapidform* is found the only software that supports parametric solid modeling. For hands-on evaluations, we selected *Geomagic* and *Rapidform*, in addition to a few CAD packages.

	Geomagic Studio	Rapidform XOR3	SolidWorks	Wildfire	CATIA	Paraform	ICEM	Imageware	CopyCAD
G=good, F=fair, P=poor, U=unknown, N=Not available									
General capabilities:									
Speed: How quickly can surface fitting be done?	G	G	P	P	F	G	F	G	G
Data/file size that the software can process	G	G	P	P	G	G	G	G	G
Screen refresh resolution	G	G	P	P	G	G	G	G	G
Output file format: transferred to other software	G	G	G	G	G	G	G	G	G
Recognition of geometry types (such as circles)	G	G	P	N	F	P	G	G	F
Generation of NURB models									
Auto Surfacing	G	G	P	N	N	N	N	N	N
Modification of points	G	G	P	P	G	G	G	G	G
Hole filling	G	G	G	P	G	G	G	G	G
Modification of spline curves	G	G	P	P	G	G	G	G	G
NURB surfaces deviation analysis	G	G	G	G	G	G	G	G	G
How complicated shapes can be generated?	G	G	F	P	G	G	G	G	G
Surface smoothness	G	G	F	F	G	G	G	G	G
Generation of solid models									
Parametric: Can parametric solid models be generated?	P	G	F	P	F	N	P	F	U
Section sketch	F	G	P	N	P	N	P	G	U
Dimensions and geometric constraints on sketch	P	F	P	N	U	N	U	U	U
Level of human interaction with the geometry required	F	G	P	P	P	N	U	F	U
Is the generated solid model easy to modify?	P	G	P	P	P	N	U	F	U
Parametric model export	P	G	P	P	P	N	U	U	U

Table 1. A summary of commercial software tools surveyed

4.2 Examples for hands-on evaluations

For hands-on evaluations, we carried out two rounds of study; round 1 focuses on auto surfacing, and round 2 is for parametric solid modeling. After surveying most advanced software as discussed in Section 4.1, we selected four candidate software tools for hands-on evaluations. They are RE-based software *Geomagic Studio v.11* and *Rapidform XOR3*; and feature-based CAD software *Pro/ENGINEER Wildfire v.4* and *SolidWorks 2009*. As shown in Table 2, all tools support surface and solid model construction, except for *Wildfire*, which does not support parametric solid modeling using scanned data.

	Surface Reconstruction	Parametric Modeling
Geomagic Studio v. 11	Shape Phase	Fashion Phase
Rapidform XOR3	Auto Surfacing	Solid/ Surface Primitives
SolidWorks 2009	Scan to 3D	Scan to 3D
Wildfire v. 4	Facet + Restyle	Not Available

Table 2. Software selected for hands-on evaluations

For round 1 evaluations, we focus on auto surfacing and the software stability. In round 2, we focus on parametric solid modeling, we look for primitive feature recognition (such as cylinder, cone, etc.), parametric modeling, and model exporting to CAD packages.

We selected five examples for hands-on evaluation, as listed in Table 3. Among the five examples, two are given as polygon meshes and the other three are point clouds. These five parts represent a broad range of applications. Parts like the *block*, *tubing*, and *door lock* are more traditional mechanical parts with regular solid features. In contrast, *sheetmetal* part (Model 3) is a formed part with large curvature, and the *blade* is basically a free-form object.






	Model 1 <i>Block</i>	Model 2 <i>Tubing</i>	Model 3 <i>Sheetmetal</i>	Model 4 <i>Blade</i>	Model 5 <i>Door Lock</i>
Model Pictures					
Scanned data	634,957 points	589,693 polygons	134,089 polygons	252,895 points	207,282 points
Dimensions	5×3×0.5 (inch)	125×93×17 (mm)	16×10×9 (inch)	2×3×4 (inch)	7×3×2 (inch)

Table 3. Examples selected for hands-on evaluations

4.3 Round 1: Auto surfacing

In round 1 evaluation, we are interested in investigating if software tools evaluated are able to support auto surfacing; i.e., automatically constructing air-tight, accurate, and high quality surface models from scanned data. We look for the level of automation, software stability, and capabilities for editing geometric entities (such as points, meshes, and NURB patches).

Based on the evaluations, we found that all software tools evaluated are able to support surface modeling either fully automatically or close to fully automation. Table 4 summarizes the test results. The results show that *Geomagic* is the only software that is able to create surface models for all five examples automatically, without any user interventions. *Rapidform* comes close second. *Rapidform* is able to construct surface models for two out of the five examples fully automatically. For the remaining three examples, only minor interventions or editing from the user are required. However, *SolidWorks* and *Wildfire* are able to support only some of the examples even after spending long hours. It took extremely long time using *SolidWorks* or *Wildfire* to process some of the examples, and yet without achieving meaningful results. Software crashed without giving warning message while conducting triangulation or surface fitting. The size of the scanned data also presents problems for *SolidWorks* and *Wildfire*. They are able to support only up to about 300,000 data points. The software becomes unstable or even crashes while handling more data points.







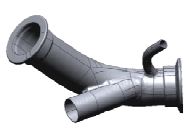













	Model 1 <i>Block</i>	Model 2 <i>Tubing</i>	Model 3 <i>Sheetmetal</i>	Model 4 <i>Blade</i>	Model 5 <i>Door Lock</i>
Geomagic Studio v.11	Completed (Automated) 	Completed (Automated) 	Completed (Automated) 	Completed (Automated) 	Completed (Automated) 
Rapidform XOR3	Completed (Automated) 	Completed (Partial-auto) 	Completed (Partial-auto) 	Completed (Partial-auto) 	Completed (Automated) 
SolidWorks 2009	Fail (Gaps remained, shown in red) 	Software crashed 	Fail (Gaps remained, shown in red) 	Completed (Automated) 	Software crashed 
Wildfire v.4	Software Crashed 	Software crashed 	Completed (Automated) 	Completed (Automated) 	Software crashed 

Table 4. Results of Round 1 evaluations

One important finding worth noting is that the mesh segmentation capability is only available in *Geomagic* and *Rapidform*. This capability allows users to adjust a sensitivity index to vary the size of segmented regions so that the regions match closely to the distinct surfaces of the object. Such segmentation is critical since the properly segmented regions facilitate surface fitting and primitive feature recognition.

Based on the findings, we exclude further discussion on *SolidWorks* and *Wildfire* due to their poor performance in the first evaluation round. In the following we discuss results of *Geomagic* and *Rapidform* for selected examples to consolidate our conclusions.

4.3.1 *Geomagic Studio v.11*

Geomagic demonstrates an excellent surface construction capability with a high level of automation. Based on our evaluations, excellent NURB surface models can be created for all five examples from their respective scanned data in less than 30 minutes. In addition, *Geomagic* offers interactive capabilities that allow users to manually edit or create geometric entities. For examples, *Point Phase* of *Geomagic* supports users to edit points, reduce data noise, and adjust sampling to reduce number of point data. After point editing operations, polygon meshes are created by using *Wrap*. In *Mesh Phase*, self-intersecting, highly creased edges (edge with sharp angle between the normal vectors of the two neighboring polygonal faces), spikes and small clusters of polygons (a group of small isolated polygon meshes) can be detected and repaired automatically by *Mesh Doctor*. Mesh editing tools; such as smooth polygon mesh, define sharp edges, defeature and fill holes; are also provided to support users to create quality polygon meshes conveniently. Once a quality mesh is generated, *Shape Phase* is employed to create NURB surfaces best fit to the polygon mesh.

Auto Surface consists of a set of steps that automatically construct surface models. The steps include *Detect Contour*, *Construct Patches*, *Construct Grids* and *Fit Surfaces*. Before using *Auto Surface*, users only have to consider the quality of the surface model (for example, specifying required tolerance) and the method (for example, with or without mesh segmentation). For the *block* example, we set *surface tolerance* to 0.01 inch and construct NURB surface model with *Detect Contours* option (which performs mesh segmentation) using *Auto Surface*. A complete NURB surface model was created in 5 minutes (Fig. 8). Average deviation of the NURB model is 0.0 inch and the standard deviation is 0.0003 inch. The deviation is defined as the shortest distance (a signed distance) between the polygon mesh and the NURB surfaces. Note that in Figure 8d, green area indicates deviation close to 0 and red spot indicates the max deviation, which is about 0.017 inch in this case.



Fig. 8. Results of the *block* example tested using *Geomagic*, (a) point cloud model (634,957 points), (b) polygon mesh (1,271,924 triangles), (c) NURB surface model, and (d) deviation analysis

Two more examples, *tubing* and *sheetmetal*, are processed following the same steps. Results are shown in Figs. 9 and 10, respectively. These examples demonstrate that *Auto surface* of *Geomagic* offers reliable, viable and extremely efficient capability for automated surface reconstruction.



Fig. 9. Results of the *tubing* example tested using *Geomagic*, (a) polygon mesh (589,693 triangles), (b) NURB model (1,107 patches), and (c) deviation analysis



Fig. 10. Results of the *sheet metal* example tested using *Geomagic*, (a) polygon mesh (126,492 triangles), (b) NURB model (91 patches), and (c) deviation analysis

4.3.2 *Rapidform XOR3*

Like *Geomagic*, *Rapidform* offers excellent capabilities for point data editing and polygon mesh generation, including data sampling, noise reduction, wrap, mesh repair, defeature, and fill holes. *Auto Surfacing* for NURB surface construction in *Rapidform* contains two methods, (1) *Feature Following Network* (with mesh segmentation), and (2) *Evenly Distribution Network* (without mesh segmentation).

Feature Following Network is a very good option for surface reconstruction in *XOR3*. Segmentation was introduced into *Auto Surfacing* to overcome problems of surface transition across sharp edges, especially dealing with mechanical parts with regular features. Using *Feature Following Network* sharp edges can be detected and retained in the surface model. *Feature Following Network* is usually more successful in surface construction. For example, in Fig. 11a, several gaps (circled in red) are found in the *block* example, mostly along narrow and high curvature transition regions, while using *Evenly Distribution Network* option for constructing surfaces. Using *Feature Following Network* option the surface model constructed is air-tight with sharp edges well preserved, as shown in Fig. 11b. Note that large size NURB surfaces (therefore, less number of NURB surfaces) shown in Fig. 11b tend to be created due to incorporation of mesh segmentation.

The NURB surface model of the *block* example (Fig. 12a) was successfully created using *Feature Following Network* option in just about 5 minutes (Fig. 12b). The accuracy measures; i.e., the deviation between the surface model and the polygon mesh, are 0.00 inch and 0.0006 inch in average and standard deviation, respectively, as shown in Fig. 12c.



Fig. 11. NURB surface models generated using two different options in *Rapidform*, (a) *Evenly Distribution Network* option, and (b) *Feature Following Network* option

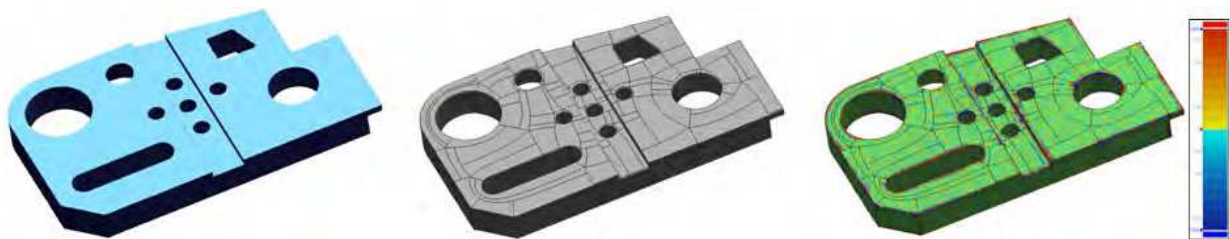


Fig. 12. Results of the *block* example tested using *Rapidform*, (a) polygon mesh (1,062,236 triangles), (b) NURB surface model (273 patches), and (c) deviation analysis

While evaluating *Rapidform* for surface construction, some issues were encountered and worth noting. First, as discussed earlier, *Rapidform* tends to create large size NURB patches that sometimes leave unfilled gaps in the surface model, especially in a long narrow region of high curvature. This happened even with *Feature Following Network* option. As shown in Fig. 13, almost half of the small branch of the *tubing* is missing after auto surfacing with *Feature Following Network* option. When such a problem appears, *Rapidform* highlights boundary curves of the gaps that are not able to be filled. In general, users can choose to reduce the gap size, for example, by adding NURB curves to split the narrow regions, until NURB patches of adequate size can be created to fill the gaps with required accuracy.

For the *tubing* example, the repair process took about 45 minutes to finish. The final surface model was created with some manual work. The average and standard deviation between the surface model and the polygon mesh are -0.0003 mm and 0.0189 mm, respectively, as shown in Fig. 14.

The sheet metal example shown in Fig. 15 also presents minor issues with *Rapidform*. The boundary edge of the part is not smooth, as common to all scanned data. *Rapidform* created a NURB curve along the boundary, and then another smoother curve very close to the boundary edge. As a result, a very long and narrow region was created between these two curves, which present problems in auto surfacing. Similar steps as to the *tubing* example were taken to split the narrow region by adding NURB curves. The final model was split in four main regions and several smaller regions shown in Fig. 16, which allows NURB surfaces to be generated with excellent accuracy (average: 0.0 in, standard deviation: 0.0002 in).

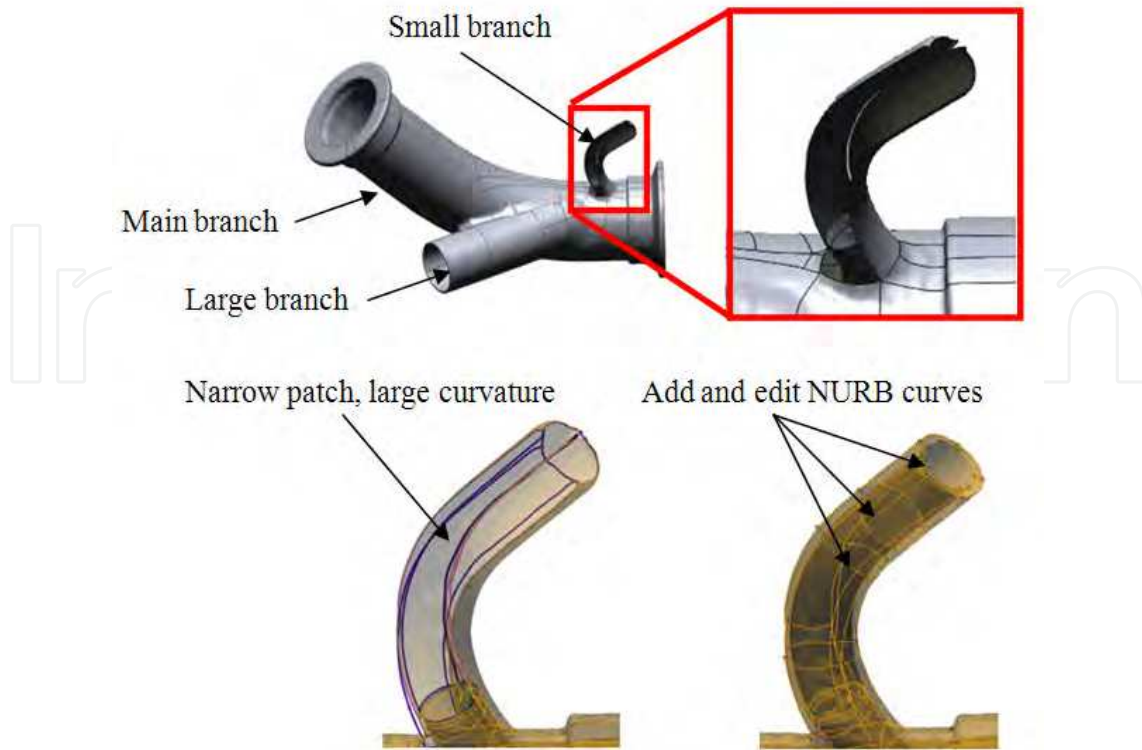


Fig. 13. Incomplete NURB surface model created by *Rapidform*



Fig. 14. Results of the *tubing* example tested using *Rapidform*, (a) polygon mesh (589,693 triangles), (b) NURB surface model (185 patches), and (c) deviation analysis

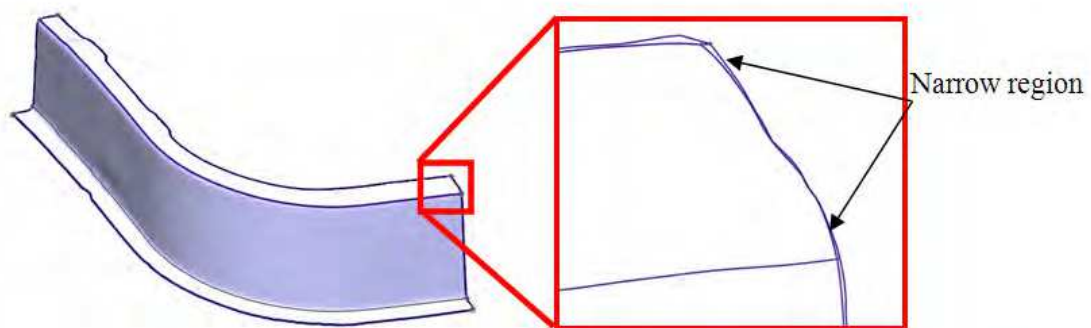


Fig. 15. Narrow regions failed for auto surfacing using *Rapidform*

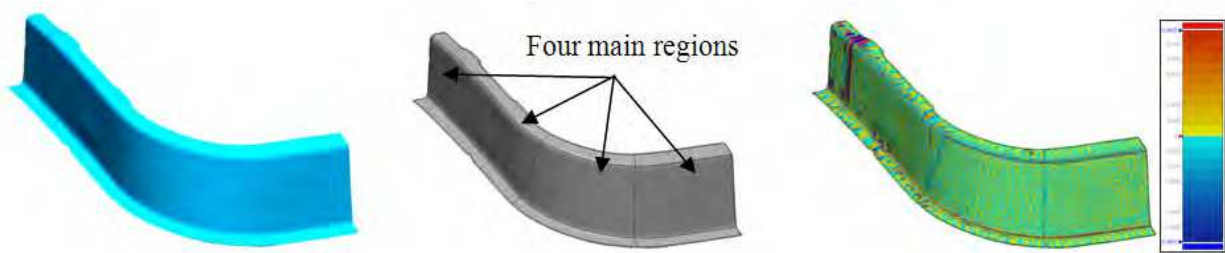


Fig. 16. Results of the sheet metal example tested using *Rapidform*, (a) polygon mesh (126,492 triangles), (b) NURB surface model (43 patches), and (c) deviation analysis

4.3.3 Summary of round one evaluations

Based on the software evaluated and examples tested, we concluded that *Geomagic* and *Rapidform* are the only viable software tools for automated surface constructions. Between these two, *Geomagic* offers more flexible and easier to use capabilities in editing NURB curves and surfaces, as well as smoothing NURB surfaces. On the other hand, *Rapidform* offers more quality measurement functions, such as continuity and surface reflection, on the constructed surface model. In addition, *Rapidform* provides feature tree that allows users to roll back and edit geometric entities created previously, which is extremely helpful in dealing with complex models. However, *Rapidform* tends to create larger NURB surfaces that could sometimes lead to problems. Overall, either tool would do a very good job for surface constructions; *Geomagic* has a slight edge in support of editing geometric entities.

4.4 Round 2: Parametric solid modeling

Although NURB surface models represent the part geometry accurately, they are not parametric. There are no CAD-like geometric features, no section profiles, and no dimensions; therefore, design change is impractical with the NURB surface models. In some applications, geometry of the parts must be modified in order to achieve better product performance, among other possible scenarios.

In round 2, we focus on evaluating parametric modeling capabilities in four software tools, including *Geomagic*, *Rapidform*, *SolidWorks*, and *Wildfire*. More specifically, we are looking for answers to the following three questions:

1. Can geometric primitives, such as cones, spheres, etc., be automatically recognized from segmented regions? How many such primitives can be recognized?
2. Whether a section sketch of a geometric feature can be created from a polygon mesh or point cloud (or segmented regions)? This is mainly for generating solid models interactively.
3. Whether a section sketch generated in (2) can be fully parameterized? Can dimensions and geometric constraints, such as concentric, equal radii, etc., be added to the section profile conveniently?

Solid modeling capabilities in the context of reverse engineering for the four selected software are listed in Table 5, based on the first glance. Among these four, *Geomagic*, *Rapidform*, and *SolidWorks* are able to recognize basic primitives, such as plane, cylinder,

sphere, etc., from segmented regions. *Wildfire* does not offer any of the modeling capabilities we are looking for; therefore, is excluded from the evaluation. Although some primitives can be recognized automatically, they often result in a partially recognized or misrecognized solid model. It takes a good amount of effort to interactively recover the remaining primitives or correct misrecognized primitives. Overall, it often requires less effort yet yielding a much better solid model by interactively recovering solid features embedded in the segmented regions. The interactive approach mainly involves creating or extracting section profiles or guide curves from a polygon mesh, and following CAD-like steps to create solid features, for example, sweep a section profile along a guide curve for a sweep solid feature.

	Q1: Recognition of geometric primitives	Recognized primitives	Q2: Section sketch	Q3: Adding dimensions and constraints
Geomagic Studio v.11	Yes (Solid + Surface)	Plane, Cylinder, Cone, Sphere, Free form, Extrusion, Revolve	Yes (Poor)	Yes (Poor)
Rapidform XOR3	Yes (Solid + Surface)	Plane, Cylinder, Cone, Sphere, Torus, Box	Yes (Excellent)	Yes (Fair)
SolidWorks 2009	Yes (Surface only)	Plane, Cylinder, Cone, Sphere, Torus, Free form, Extrusion, Revolve	Yes (Poor)	Yes (Poor)
Wildfire v.4	No	No	No	No

Table 5. Feature primitive recognition capabilities of selected software

Among the remaining three, *SolidWorks* is most difficult to use; especially in selecting misrecognized or unrecognized regions to manually assign a correct primitive type. The system responds very slowly and only supports surface primitive recognition. Therefore, *SolidWorks* is also excluded in this round of evaluations.

4.4.1 Geomagic Studio v.11

Geomagic automatically recognizes primitive surfaces from segmented regions. If a primitive surface is misrecognized or unrecognizable, users are able to interactively choose the segmented region and assign a correct primitive type. Often, this interactive approach leads to a solid model with all bounding surfaces recognized. Unfortunately, there is no feature tree, and no CAD-like capabilities in *Geomagic*. Users are not able to see any sketch or dimensions in *Geomagic Studio v.11*. Therefore, users will not be able to edit or add any dimensions or constraints to parameterize the sketch profiles. Section sketches only become available to the users after exporting the solid model to a selected CAD package supported by *Geomagic*.

The *block* example (3in.×5in.×0.5in.) of 634,957 points shown in Fig. 4 is employed to illustrate the capabilities offered in *Geomagic*. As shown in Fig. 17a, primitive surfaces in most regions are recognized correctly. However, there are some regions incorrectly recognized; for example, the hole in the middle of the *block* was recognized as a free-form primitive, instead of a cylinder. There are also regions remained unrecognized; e.g., the middle slot surface.

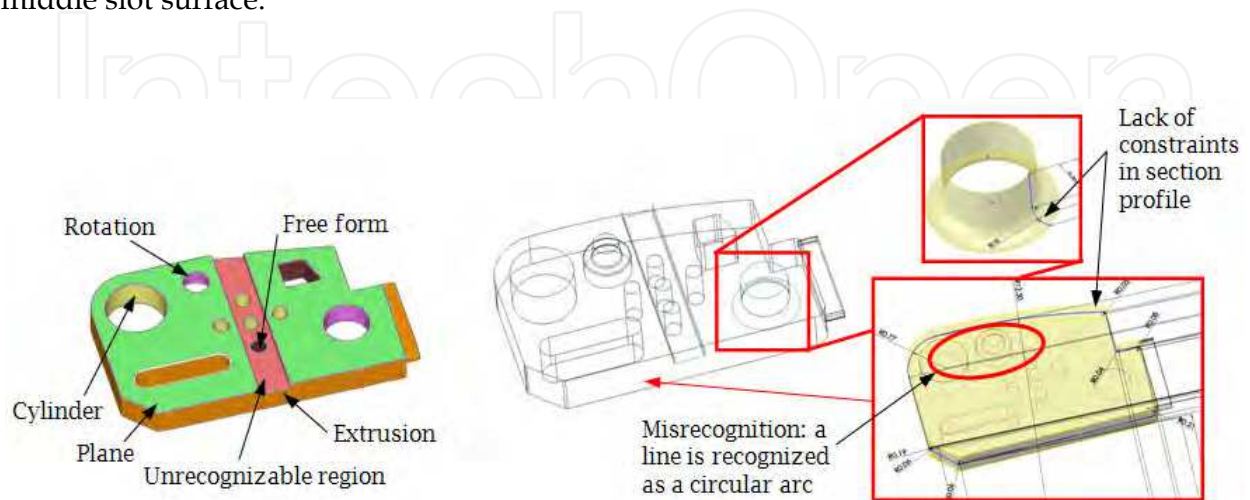


Fig. 17. Primitive surfaces recognized in *Geomagic*, (a) recognized regions, and (b) extracted primitive surfaces in *SolidWorks*

Although most primitives are recognized in *Geomagic*, there are still issues to address. One of them is misrepresented profile. One example is that a straight line in a sketch profile may be recognized as a circular arc with a very large radius, as shown in Fig. 17b (this was found only after exporting the solid model to *SolidWorks*). The sketch profile will have to be carefully inspected to make necessary corrections, as well as adding dimensions and constraints to parameterize the profile. Unfortunately, such inspections cannot be carried out unless the solid model is exported to supported CAD systems. Lack of CAD-like capability severely restricts the usability of the solid models in *Geomagic*, let alone the insufficient ability for primitive surface recognition.

4.4.2 *Rapidform XOR3*

Rapidform offers much better capabilities than *Geomagic* for parametric solid modeling. Very good CAD-like capabilities, including feature tree, are available to the users. These capabilities allow users to create solid models and make design changes directly in *Rapidform*. For example, users will be able to create a sketch profile by intersecting a plane with the polygon mesh, and extrude the sketch profile to match the bounding polygon mesh for a solid feature. On the other hand, with the feature tree users can always roll back to previous entities and edit dimensions or redefine section profiles. These capabilities make *Rapidform* particularly suitable for parametric solid modeling. *Rapidform* offers two methods for solid modeling, *Sketch*, and *Wizard*, supporting fast and easy primitive recognition from segmented mesh. The major drawback of the *Wizard* is that some guide curves and profile

sketch generated are non-planar spline curves that cannot be parameterized. Users can use either or both methods to generate solid features for a single part.

Method 1: Sketch

In general, there are six steps employed in using the sketch method, (1) creating reference sketch plane, (2) extracting sketch profile by intersecting the sketch plane with the polygon mesh, (3) converting extracted geometric entities (usually as planar spline curves) into regular line entities, such as arcs and straight lines, (4) parameterizing the sketch by adding dimensions and constraints, (5) extruding, revolving, or lofting the sketches to create solid features; and (6) employing Boolean operations to union, subtract, or intersect features if necessary.

Rapidform provides *Auto Sketch* capability that automatically converts extracted spline curves into lines, circles, arcs, and rectangles, with some constraints added. Most constraints and dimensions will have to be added interactively to fully parameterize the sketch profile. Steps 4 to 6 are similar to conventional CAD operations. With capabilities offered by *Rapidform*, fully constrained parametric solid models can be created efficiently.

For the *block* example, a plane that is parallel to the top (or bottom) face of the base block was created first (by simply clicking more than three points on the surface). The plane is offset vertically to ensure a proper intersection between the sketch plane and the polygon mesh. The geometric entities obtained from the intersection are planar spline curves. The *Auto Sketch* capability of *Rapidform* can be used to extract a set of regular CAD-like line entities that best fit the spline curves. These standard line entities can be joined and parameterized by manually adding dimensions and constraints for a fully parameterized section profile, as shown in Fig. 18a.

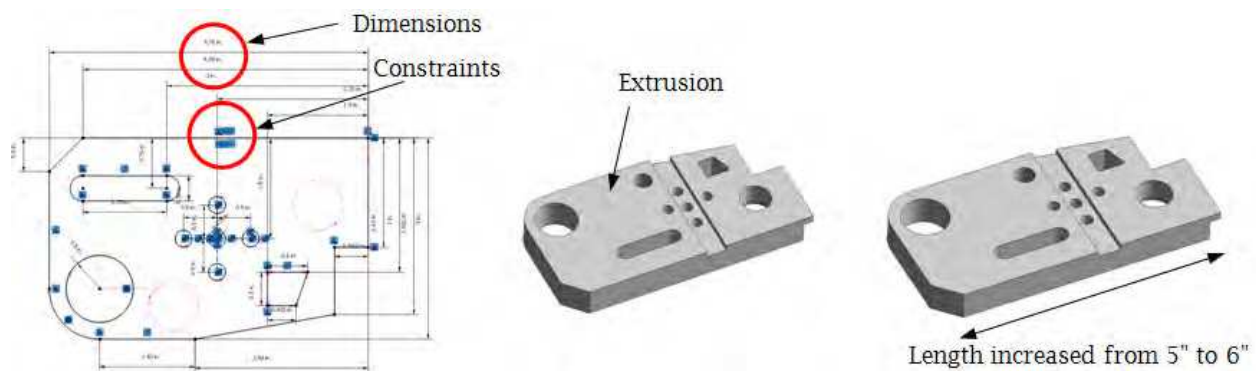


Fig. 18. A parametric solid model of the *block* example created using *Rapidform*, (a) fully parameterized section sketch, (b) extrusion for the base block, and (c) design change

Once the sketch profile is parameterized, it can be extruded to generate an extrusion feature for the base block (Fig. 18b). The same steps can be followed to create more solid features, and Boolean operations can be employed to union, subtract, or intersect solid features for a fully parameterized solid model. The final solid model is analyzed by using *Accuracy*

Analyzer. The solid model generated is extremely accurate, where geometric error measured in average and standard deviation is 0.0002 and 0.0017 in., respectively (between the solid model and point cloud). Since the model is fully parameterized, it can be modified by simply changing the dimension values. For example, the length of the base block can be increased for an extended model, as shown in Fig. 18c.

Method 2: Wizard

Wizard, or *Modeling Wizard*, of *Rapidform* automatically extracts *Wizard* features such as extrude, revolve, pipe, and loft, etc., to create solid models from segmented regions. Note that a *Wizard* feature can be a surface (such as pipe) or a solid feature. There are five *Wizard* features provided: *extrusion*, *revolution* for extracting solid features; and *sweep*, *loft*, and *pipe* for surface features. There are three general steps to extract features using *Wizard*, (1) select mesh segments to generate individual features using *Wizard*, (2) modify the dimensions or add constraints to the sketches extracted in order to parameterize the sketches, and (3) use Boolean operations to union, subtract, or intersect individual features for a final model if needed.

The same *tubing* example shown in Fig. 19 is employed to illustrate the capabilities offered in *Wizard*. We start with a polygon mesh that has been segmented, as shown in Fig. 19a. First, we select the exterior region of the main branch and choose *Pipe Wizard*. *Rapidform* uses a best fit pipe surface to fit the main branch automatically, as shown in Fig. 19b. Note that the *Pipe Wizard* generates section profile and guide curve as spatial (non-planar) spline curves, which cannot be parameterized. Also, wall thickness has to be added to the pipe to complete the solid feature. Next, we choose *Revolution Wizard* to create revolved features for the top and bottom flanges, as shown in Fig. 19c. Note that each individual features are extracted separately. They are not associated. Boolean operations must be applied to these decoupled features for a final solid model.

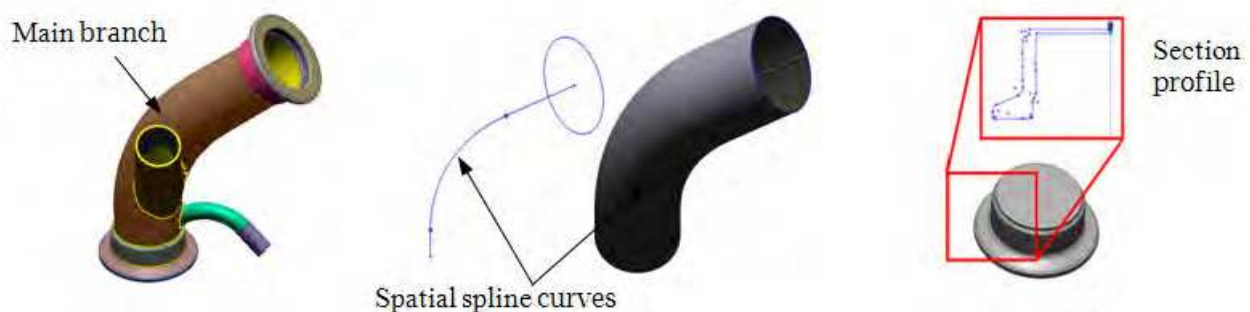


Fig. 19. Feature extraction for the *tubing* example using *Wizard*, (a) selected main branch region, (b) surface created using *Pipe Wizard*, and (c) flange created using *Revolution Wizard*

Although *Wizard* offers a fast and convenient approach for solid modeling, the solid models generated are often problematic. The solid models have to be closely examined for validation. For example, in this *tubing* model, there are gap and interference between

features, as indicated in Fig. 20. This is not a valid solid model. It is inflexible to edit and make changes to the *Wizard* features since the sketch profile is represented in spatial spline curves that cannot be constrained or dimensioned.



Fig. 20. Gap and interference between solid features in the *tubing* model

In summary, *Rapidform* is the only reverse engineering software that supports for creating parametric solid models from scanned data. *Rapidform* offers CAD-like capabilities that allow users to add dimensions and constraints to sketches and solid features for a fully parametric solid model. In addition, *Rapidform* provides two modeling methods, *Sketch* and *Wizard*. Design intent and model accuracy can be achieved using the *Sketch* method, which is in general a much better option for creating parametric solid models.

4.5 Solid model export

The solid models created in specialized software, such as *Rapidform* and *Geomagic*, have to be exported to mainstream CAD systems in order to support engineering applications. Both *Rapidform* and *Geomagic* offer capabilities that export solid models to numerous CAD systems.

4.5.1 Parametric Exchange of Geomagic

The solid model of the *block* example created in *Geomagic* was exported to *SolidWorks* and *Wildfire* using *Parametric Exchange* of *Geomagic*. For *SolidWorks*, all seventeen features recognized in *Geomagic* (see Fig. 21a) were exported as individual features, as shown in Fig. 21b. Note that since there are no Boolean operations offered in *Geomagic Studio v.11*, these features are not associated. There is no relation established between them. As a result, they are just "piled up" in the solid model shown in Fig. 21c. Subtraction features, such as holes and slots, simply overlap with the base block. Similar results appear in *Wildfire*, except that one extrusion feature was not exported properly, as shown in Fig. 21d and 21e.

4.5.2 *liveTransfer*TM module of *Rapidform XOR3*

The *liveTransfer*TM module of *Rapidform XOR3* exports parametric models, directly into major CAD systems, including *SolidWorks* 2006+, Siemens NX 4+, *Pro/ENGINEER Wildfire* 3.0+, CATIA V4 and V5 and *AutoCAD*.

The *block* example that was fully parameterized in *Rapidform* was first exported to *SolidWorks*. All the solid features were seamlessly exported to *SolidWorks*, except for some datum entities, such as datum points. Since entities such as polygon meshes and segmented regions are not included in *SolidWorks* database, they cannot be exported. As a result, geometric datum features associated with these entities are not exported properly. The dimensions and constraints added to the sketches and solid features in *Rapidform* are exported well, except again for those referenced to entities that are not available in *SolidWorks*. Fortunately, it only requires users to make a few minor changes (such as adding or modifying dimensions or constraints) to bring back a fully parametric solid model in *SolidWorks*. As shown in Fig. 22, the length of the base block was increased and the solid model is rebuilt in *SolidWorks* (Fig. 22b). Similar results were observed in *NX*. However,



Fig. 21. The *block* model explored to *SolidWorks* and *Wildfire*, (a) seventeen features recognized in *Geomagic*, (b) features exported to *SolidWorks* (wireframe), (c) features "piled up" in *SolidWorks*, (d) features exported to *Wildfire* (wireframe), and (e) features "piled up" in *Wildfire*



Fig. 22. *Block* exported from *Rapidform* to *SolidWorks*, (a) solid model exported to *SolidWorks*, and (b) design change made in *SolidWorks*

model exported to *Wildfire* 4.0 is problematic, in which numerous issues, such as missing and misinterpretation portion of the section profile, are encountered. In general, parametric solid models created in *Rapidform* can be exported well to *SolidWorks* and *NX*. The export is almost seamless. Although, there were minor issues encountered, such as missing references for some datum points, those issues can be fixed very easily.

5. Discussion

The most useful and advanced shape engineering capabilities are offered in specialized, non-CAD software, such as *Geomagic*, *Rapidform*, etc., that are intended to support reverse engineering. Some CAD packages, such as *SolidWorks*, *Pro/ENGINEER Wildfire*, and *CATIA*, offer limited capabilities for shape engineering. In general, capabilities offered in CAD are labor intensive and inferior to specialized codes while dealing with shape engineering.

After intensive review and survey (Chang & Chen, 2010), to the authors' knowledge, the best software on the market for reverse engineering is *Geomagic Studio* v.11 and *Rapidform XOR3*. This was determined after a thorough and intensive study, following a set of prescribed criteria including auto surfacing, parametric solid modeling, and software usability. Between the two, *Geomagic* has a slight edge in geometric entity editing, which is critical for auto surfacing. In terms of solid modeling, *Geomagic* stops short at only offering primitive surfaces, such as plane, cylinder, sphere, etc., from segmented regions.

Rapidform is superior in support of solid modeling (in addition to excellent auto surfacing) that goes beyond primitive surface fitting. *Rapidform* offers convenient sketching capabilities that support feature-based modeling. As a result, it often requires less effort yet yielding a much better solid model by interactively recovering solid features embedded in the segmented regions. The interactive approach mainly involves creating or extracting section profiles or guide curves from the polygon mesh, and following CAD-like steps to create solid features.

6. Conclusions

In this chapter, technology that enables 3D shape engineering and design parameterization for reverse engineering was reviewed. Software that offers such capabilities was also evaluated and tested using practical examples. Based on the evaluations, we observed that *Rapidform* is the only viable choice for parametric solid modeling in support of 3D shape engineering and design parameterization. *Rapidform* offers CAD-like capabilities for creating solid features, feature tree for allowing roll back for feature editing, and very good sketching functions. In addition, the *liveTransfer*TM module offers model exporting to mainstream CAD systems almost seamlessly.

After research and development in decades, technology that supports 3D shape engineering and design parameterization is matured enough to support general engineering applications. The ideal scenario can now be realized by using software such as *Rapidform* for shape engineering and parameterization, where labor intensive tasks, such as managing point cloud, triangulation, etc., is taken care of in an automated fashion; and design intents

can be recovered interactively as desired. One area that might require more work is to incorporate more CAD packages for model export. Major CAD packages, such as *SolidWorks* and *NX*, have been well supported. However, software such as *CATIA* is yet to be included and software like *Wildfire* needs to be streamlined.

7. References

- Alrashdan, A.; Motavalli, S.; Fallahi, B.: Automatic Segmentation of Digitized Data for Reverse Engineering Applications, *IIE Transactions*, 32(1), 2000, 59-69. DOI: 10.1023/A:1007655430826
- Bernardini, F.; Mittleman, J.; Rushmeier, H.; Silva, C.; Taubin, G.: The Ball-Pivoting Algorithm for Surface Reconstruction, *Visualization and Computer Graphics*, 5(4), 1999, 349-359. DOI:10.1109/2945.817351
- Besl, P.J.; Jain, R.C.: Segmentation Through Variable-Order Surface Fitting, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(2), 1988, 167-192. DOI: 10.1109/34.3881
- Blais, F.: Review of 20 Years of Range Sensor Development, *Journal of Electronic Imaging*, 13(1), 2004, 231-240. DOI:10.1117/1.1631921
- Chang, K.H.; Magdum, S.; Khera, S.; Goel, V.K.: An Advanced Computer Modeling and Prototyping Method for Human Tooth Mechanics Study, *Annals of Biomedical Engineering*, 31(5), 2003, 621-631. DOI:10.114/1.1568117
- Chang, K.H. and Chen, C., Research and Recommendation of Advanced Reverse Engineering Tools, Final Technical Report, Reference SOW # QIB09-008, September 2010.
- Chang, K.H. and Siddique, Z., "Re-Engineering and Fast Manufacturing for Impact-Induced Fatigue and Fracture Problems in Aging Aircrafts," Final Report, AFOSR Grant F49620-02-1-0336, January 31, 2006.
- Cuccuru, G.; Gobbetti, E.; Marton, F.; Pajarola, R.; Pintus, R.: Fast Low-Memory Streaming MLS Reconstruction of Point-Sampled Surfaces, *Graphics Interface*. 2009, 15-22.
- Edelsbrunner, H.; Kirkpatrick, D.G.; Seidel, R.: On the Shape of A Set of Points In The Plane, *IEEE Transactions on Information Theory*, 29(4), 1983, 551-559. DOI:10.1109/TIT.1983.1056714
- Geomagic, www.geomagic.com
- Gibson, D., Parametric feature recognition and surface construction from digital point cloud scans of mechanical parts, MS Thesis, The University of Oklahoma, December 2004
- Hansen, G.A.; Douglass, R.W; and Zardecki, Andrew: *Mesh enhancement*. Imperial College Press, 2005, ISBN-13: 978-1860944871
- Hoppe, H.; DeRose, T.; Duchamp, T.; McDONALS, J., and Stuetzle, W.: Mesh Optimization, SIGGRAPH '93 Proceedings of the 20th annual conference on Computer graphics and interactive techniques, ACM New York, ISBN:0-89791-601-8
- Huang, J.; Menq, C.-H.: Automatic Data Segmentation for Geometric Feature Extraction From Unorganized 3-D Coordinate Points, *IIE Transactions on Robotics and Automation*, 17(3), 2001, 268-279. DOI:10.1109/70.938384

InnovMetric, www.innovmetric.com

Kazhdan, M.; Bolitho, M.; Hoppe, H.: Poisson Surface Reconstruction, The Fourth Eurographics Symposium on Geometry Processing, 2006.

Li, Z.; Ma, L.; Jin, X.; and Zheng, Z.: A new feature-preserving mesh-smoothing algorithm, *The visual computer* 2009, vol. 25, n°2, pp. 139-148

Liu, Y.; Dong, X.; Peng, W.: Study on Digital Data Processing Techniques for 3D Medical Model, *Bioinformatics and Biomedical Engineering*, 4, 2010, 1-4. DOI:10.1109/ICBBE.2010.5518168

Lorensen, W.E.; Cline, H.E.: Marching Cubes: A High Resolution 3D Surface Construction Algorithm, *Computer Graphics*, 21(4), 1987. DOI:10.1145/37402.37422

Lukács, G.; Martin, R.; Marshall, D.: Faithful Least-Squares Fitting of Spheres, Cones and Tori for Reliable Segmentation, *Lecture Notes in Computer Science*, 1406, 1998, 671-686. DOI:10.1007/BFb0055697

Lv, Y.; Yi, J.; Liu, Y.; Zhao, L.; Zhang, Y.; Chen, J.: Research on Reverse Engineering for Plastic Operation, *Information Technology and Applications*, 2009, 389-391. DOI: 10.1109/IFITA.2009.123

Marshall, D.; Lukacs, G.; Martin, Ralph.: Robust Segmentation of Primitives from Range Data in the Presence of Geometric Degeneracy, *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 23(3), 2001,304-314. DOI:10.1109/34.910883

Raja, V.; Fernandes, K.J.: *Reverse Engineering: An industrial Perspective*, Springer-Verlag, London, 2008. DOI:10.1007/978-1-84628-856-2

Rapidform, INUS Technology, Inc., www.rapidform.com

Schroeder, W.J.: A topology modifying progressive decimation algorithm, Eighth IEEE Visualization 1997 (VIS '97), Phoenix, AZ, October 19-October 24, ISBN: 0-8186-8262-0

Siddique, Z.: Recommended Tool Kit for Reverse Engineering, CASI Summer Research Program 2002, OC-ALC, August 28, 2002

Silva, J.S.; Chang, K.H.: Design Parameterization for Concurrent Design and Manufacturing of Mechanical Systems, *Concurrent Engineering Research and Applications (CERA) Journal*, 2002, 10(1), 3-14. DOI:10.1177/1063293X02010001048

SolidWorks, Inc., www.solidworks.com

Sun, Q.; Chang, K.H.; Dormer, K.; Dyer, R.; Gan, R.Z.: An Advanced Computer-Aided Geometric Modeling and Fabrication Method for Human Middle Ear, *Medical Engineering and Physics*, 24(9), 2002, 595-606. DOI:10.1016/S1350-4533(02)00045-0

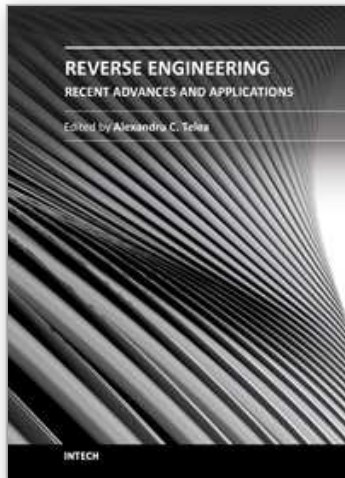
Thompson, W.B.; Owen, J.C.; de St. Germain, H.J.; Stark, S.R., Jr.; Henderson, T.C.: Featured-Based Reverse Engineering of Mechanical Parts, *IEEE Transactions on Robotics and Automation*, 15(1), 1999, 57-66. DOI:10.1109/70.744602

Tsai, Y.C.; Huang, C.Y.; Lin, K.Y.; Lai, J.Y.; Ueng, W.D.: Development of Automatic Surface Reconstruction Technique in Reverse Engineering, *The International Journal of Advanced Manufacturing Technology*, 42(1-2), 2009, 152-167. DOI:10.1007/S00170-008-1586-2

Vanco, M.; Brunnett, G.: Direct Segmentation of Algebraic Models for Reverse Engineering, *Computing*, 72(1-2), 2004, 207-220. DOI:10.1007/S00607-003-0058-7

- Várady, T.; Benkő, P.; Kós, G.: Reverse Engineering Regular Objects: Simple Segmentation and Surface Fitting Procedures, *International Journal of Shape Modeling*, 4(3-4), 1998, 127–141. DOI:10.1142/S0218654398000106
- Várady, T.; Martin, R.R.; Cox, J.: Reverse Engineering of Geometric Models - An Introduction, *Computer-Aided Design*, 29(4), 1997, 255–268. DOI:10.1016/S0010-4485(96)00054-1
- Venkataraman, S.; Sohoni, M.; Kulkarni, V.: A Graph-Based Framework for Feature Recognition, *Sixth ACM Symposium on Solid Modeling and Applications*, 2001, 194-205. DOI:10.1145/376957.376980
- ZCorp, www.zcorp.com

IntechOpen



Reverse Engineering - Recent Advances and Applications

Edited by Dr. A.C. Telea

ISBN 978-953-51-0158-1

Hard cover, 276 pages

Publisher InTech

Published online 07, March, 2012

Published in print edition March, 2012

Reverse engineering encompasses a wide spectrum of activities aimed at extracting information on the function, structure, and behavior of man-made or natural artifacts. Increases in data sources, processing power, and improved data mining and processing algorithms have opened new fields of application for reverse engineering. In this book, we present twelve applications of reverse engineering in the software engineering, shape engineering, and medical and life sciences application domains. The book can serve as a guideline to practitioners in the above fields to the state-of-the-art in reverse engineering techniques, tools, and use-cases, as well as an overview of open challenges for reverse engineering researchers.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Kuang-Hua Chang (2012). A Review on Shape Engineering and Design Parameterization in Reverse Engineering, *Reverse Engineering - Recent Advances and Applications*, Dr. A.C. Telea (Ed.), ISBN: 978-953-51-0158-1, InTech, Available from: <http://www.intechopen.com/books/reverse-engineering-recent-advances-and-applications/3d-shape-engineering-and-design-parameterization>

INTECH
open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2012 The Author(s). Licensee IntechOpen. This is an open access article distributed under the terms of the [Creative Commons Attribution 3.0 License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

IntechOpen

IntechOpen