

**A HYBRID CAD MODELER FOR FLEXIBLE GEOMETRIC MODELING OF
PRISMATIC MECHANICAL PARTS**

by

Jack Szu-Shen Chen

B.A.Sc., The University of British Columbia, 2008

M.A.Sc., The University of British Columbia, 2010

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

in

The Faculty of Graduate and Postdoctoral Studies

(Mechanical Engineering)

THE UNIVERSITY OF BRITISH COLUMBIA

(Vancouver)

July 2015

© Jack Szu-Shen Chen, 2015

Abstract

The ease and freedom of shape manipulation achievable through physical modeling materials such as clay for engineering design is steps beyond what is attainable through current computer-aided design (CAD) modelers. With the current CAD modeling paradigm, which creates models through a composition of features constrained by closed-form mathematical formulations, the flexibility in model shape manipulation is limited. As a result, many complex engineering shape designs are either completed in the physical regime then digitized or in the computer-graphics domain where modeling flexibility is significantly higher. Unfortunately, the gain in modeling flexibility is achieved at the expense of well-controlled feature information and modeling precision and accuracy. The resulting models are featureless inexact entities. In order to bring forth geometric modeling flexibility while retaining feature information along with modeling precision and accuracy, this thesis presents a novel hybrid CAD modeler. The hybrid modeler utilizes triangle mesh model representation with the notion of feature imposed as a separate but associated feature information layer. This allows the prismatic features on a model to be modified unrestrictedly without loss of feature information. Users can freely modify the model geometry beyond what is permissible by the current CAD modelers' feature formulations/management. A robust feature segmentation scheme that divides a triangle mesh model into its elementary features automatically categorizes the unconstrained user modifications into prismatic features and updates the associated feature information layer. An idealization module completes the prismatic feature information extraction process and updates the mesh model to accurately reflect the newly detected/extracted features. Feature-based model

editing is incorporated to permit accurate and precise feature-based editing and to maximize the ease-of-use of the hybrid modeler. Accordingly, the hybrid modeler consists of four modules: feature segmentation, feature idealization, unconstrained feature-free model modification and feature-based model editing. With the proposed hybrid modeler, modeling flexibility as well as precision and accuracy are satisfied simultaneously. The hybrid modeler provides the user with a flexible modeling environment for creating and modifying prismatic engineering design models.

Preface

The contents of each chapter in this thesis have been/will be published in peer reviewed journals with the publication schedule directly following the order of the chapters. The technical content of each chapter was all completed by me under the guidance of my research supervisor, Dr. H.Y. Feng. The overall direction of the thesis research was originally recommended by my supervisor but the detailed research subjects as presented in the individual thesis chapters were my original work.

List of Publications:

1. Chen, J.S.S. and Feng, H.Y., 2014, "Automatic prismatic feature segmentation of scanning-derived meshes utilizing mean curvature histograms," *Virtual and Physical Prototyping*, **9**(1), pp. 45-61.
2. Chen, J.S.S. and Feng, H.Y., 2015, "Idealization of scanning-derived triangle mesh models of prismatic engineering parts," *International Journal on Interactive Design and Manufacturing*, in press (DOI: 10.1007/s12008-015-0262-7).
3. Chen, J.S.S. and Feng, H.Y., "Structure-altering feature-based triangle mesh model editing for prismatic engineering parts," submitted.
4. Chen, J.S.S. and Feng, H.Y., "A hybrid CAD modeler for flexible geometric modeling of prismatic mechanical parts," to be submitted.

Chapter 3 is based on article #1, which presents an enhanced method for segmenting scanning-derived triangle mesh models of physical prismatic mechanical parts into elementary features. I was responsible for the problem identification, methodology formulation, algorithm preparation and case study verification. I also wrote the majority of the manuscript with input from my research supervisor.

Chapter 4 is based on article #2. In this chapter, I proposed a mesh idealization method which automatically idealizes scanning-derived mesh models into ideal engineering models in order to better integrate physical 3D geometry capturing with computer-aided design. I developed the idealization scheme, prepared the algorithm, conducted case studies to verify its effectiveness and wrote the manuscript for the journal article.

Chapter 6 is based on article #3. In this chapter, I presented an approach to enable feature-based model editing in the triangle mesh regime. This was previously not realized in any known research work. I was responsible for problem identification, algorithm realization, and case study verification. I also wrote the majority of the journal article's manuscript.

Article #4 outlined the proposed flexible hybrid CAD modeler with much of its content derived from Chapter 7. I performed all the tests and case studies for this work. The novel modeling approach is a collaborative product of my research supervisor and me.

Table of Contents

Abstract.....	ii
Preface.....	iv
Table of Contents	vi
List of Tables	x
List of Figures.....	xi
List of Symbols	xv
List of Abbreviations	xvi
Acknowledgements	xvii
Dedication	xviii
Chapter 1: Introduction	1
1.1 Current CAD modeling technologies.....	3
1.2 Proposed hybrid modeling scheme	12
1.3 Primary components	18
Chapter 2: Literature Review.....	20
2.1 Existing mesh segmentation methods.....	20
2.1.1 STL meshes.....	24
2.1.2 Computer graphics meshes	25
2.1.3 Scanning-derived meshes.....	26
2.2 Existing idealization methods.....	28
2.3 Existing unconstrained feature-free mesh manipulation techniques	30

2.4	Related feature-based model editing work	33
Chapter 3: Feature Segmentation		36
3.1	Methodology	37
3.1.1	Fundamental principle	38
3.1.2	Imperfection effect reduction.....	41
3.2	Detailed procedure	46
3.2.1	Sharpest feature extraction.....	46
3.2.1.1	Mean curvature histogram construction.....	46
3.2.1.2	Mean curvature histogram smoothing	47
3.2.1.3	Extraction of peaks and valleys	48
3.2.1.4	Segmentation.....	49
3.2.1.5	Transition thinning post-processing.....	50
3.2.2	Non-sharp feature segmentation	52
3.3	Case studies.....	52
3.3.1	Ideal meshes.....	53
3.3.2	Synthesized non-ideal meshes	57
3.3.3	Scanning-derived and user manipulated meshes	63
3.4	Summary.....	69
Chapter 4: Mesh Idealization		72
4.1	Methodology	75
4.2	CAD modeling workflow	79
4.3	Detailed procedure	81

4.3.1	Feature identification	82
4.3.2	Idealization: part 1	85
4.3.2.1	Parameter extraction	86
4.3.2.2	Model orientation and global feature alignment.....	89
4.3.3	Idealization: part 2	90
4.3.3.1	Cylinders	93
4.3.3.2	Cones.....	96
4.3.3.3	Spheres	97
4.3.3.4	Tori.....	98
4.4	Case studies.....	101
4.4.1	Synthesized non-ideal mesh models: case 1	101
4.4.2	Synthesized non-ideal mesh models: case 2	104
4.4.3	Synthesized non-ideal mesh models: case 3 – 5	107
4.4.4	Scanning-derived and user manipulated mesh models	109
4.5	Summary	111
Chapter 5: Unconstrained Feature-Free Model Modification.....		114
5.1	Prime candidate.....	114
5.2	Summary.....	116
Chapter 6: Feature-Based Model Editing.....		118
6.1	Non-structure-altering edits	122
6.1.1	Feature parameter update	122
6.1.2	Adaptive meshing	126

6.2	Structure-altering edits.....	127
6.2.1	Information layer mapping	128
6.2.2	Mesh-based feature editing.....	129
6.2.3	Mesh adaptation	134
6.2.4	Genus alteration	135
6.3	Case studies.....	138
6.3.1	Non-structure-altering edits	138
6.3.2	Structure-altering edits.....	140
6.3.3	General edits	143
6.4	Summary	144
Chapter 7: Hybrid CAD Modeler		147
7.1	Methodology recap	147
7.2	Implementation details.....	148
7.3	Case studies.....	150
7.3.1	Case 1: basic modeling via proposed hybrid CAD modeler.....	150
7.3.2	Case 2: complex modeling via proposed hybrid CAD modeler	152
7.3.3	Case 3: complex modeling via proposed hybrid CAD modeler – reverse order	154
7.3.4	Case 4: complex modeling via proposed hybrid CAD modeler – mixed usage	155
Chapter 8: Conclusions and Future Work		158
References		162

List of Tables

Table 3.1 Number of correctly segmented patches for real scanning-derived mesh models.	66
Table 3.2 Computational time comparison.	66
Table 4.1 Feature parameter extraction.	88

List of Figures

Figure 1.1 Difficulty in geometric manipulation due to individualized features.....	4
Figure 1.2 Parametric feature-based modeling.	7
Figure 1.3 Direct modeling.	9
Figure 1.4 Flexibility of shape representation via triangle mesh.....	13
Figure 1.5 Feature as model information rather than model definition.	15
Figure 1.6 Triangle mesh based CAD modeling scheme.	17
Figure 2.1 Typical triangle mesh types and their characteristics.....	24
Figure 3.1 Mean vs. Gaussian curvature.....	39
Figure 3.2 Object identification (top) through mean curvature histogram (bottom).	41
Figure 3.3 Typical curvature histogram for an ideal mesh from a CAD model.	42
Figure 3.4 Mean curvature histogram of a mesh containing chamfered edges and noise.	43
Figure 3.5 Proposed mean curvature histogram based segmentation method.	44
Figure 3.6 Effect of Laplacian smoothing on the mean curvature histogram (sharpest features removed).	45
Figure 3.7 Segmentation via curvature histogram.	50
Figure 3.8 Thinning of a transition section that is more than one-triangle wide.	51
Figure 3.9 An ideal CAD mesh (shaded).....	53
Figure 3.10 Elementary features on the ideal mesh model and the corresponding mean curvature histogram.....	54
Figure 3.11 Sharpest feature extraction for the ideal mesh model.	55

Figure 3.12 Non-sharp feature segmentation for the ideal mesh model.	56
Figure 3.13 Segmentation results: (a) Shape Diameter; (b) Random Walks; (c) Fitting Primitives; (d) K-Means; (e) Feature-based; (f) Local Geometric Filter; and (g) proposed method. (a)-(e) replicated from Wang and Yu (2011).	57
Figure 3.14 A synthesized scanning-derived mesh.....	58
Figure 3.15 Sharpest feature extraction for the synthesized scanning-derived mesh.	59
Figure 3.16 Non-sharp feature segmentation for the synthesized scanning-derived mesh.....	60
Figure 3.17 Sharpest feature extraction for the synthesized user-manipulated mesh.	61
Figure 3.18 Non-sharp feature segmentation for the synthesized user-manipulated mesh.	62
Figure 3.19 Segmentation results of additional synthesized scanning-derived meshes.	62
Figure 3.20 Segmentation results of additional synthesized user manipulated meshes.	63
Figure 3.21 Segmentation results of real scanning-derived mesh models.....	65
Figure 3.22 Segmentation results of real user-manipulated mesh models.....	68
Figure 3.23 Segmentation robustness with regards to surface undulations.	69
Figure 4.1 Proposed mesh-based idealization scheme.....	76
Figure 4.2 Typical CAD modeling workflow.....	80
Figure 4.3 Feature identification: effective zero determination.....	83
Figure 4.4 Feature identification: detailed curvature rules.	85
Figure 4.5 Edge extraction in a transition patch.	92
Figure 4.6 Inflation technique to determine tangentially continuous edges.	92
Figure 4.7 Example of how a cylinder relates to its adjacent planes.....	94
Figure 4.8 Sample cone variations.....	97

Figure 4.9 Sample sphere variations.	98
Figure 4.10 Sample torus variations.	100
Figure 4.11 Case study 1: synthesized non-ideal mesh models with normal geometric complexity: (a), (f), (k) synthesized non-ideal meshes; (b), (g), (l) segmented non-ideal meshes; (c), (h), (m) idealized meshes; (e), (j), (o) deviation map of idealized meshes.	104
Figure 4.12 Case study 2: synthesized non-ideal mesh models with increased geometric complexity: (a), (f), (k) synthesized non-ideal meshes; (b), (g), (l) segmented non-ideal meshes; (c), (h), (m) idealized meshes; (e), (j), (o) deviation map of idealized meshes.	106
Figure 4.13 Additional synthesized non-ideal mesh models.	108
Figure 4.14 Scanning-derived mesh models.	110
Figure 4.15 User manipulated mesh model.	111
Figure 5.1 Unconstrained mesh manipulation for engineering design applications.	117
Figure 6.1 Proposed feature-based mesh editing method flow chart.	120
Figure 6.2 Tangential continuity fulfillment via positional continuity satisfaction given feature orthogonality.	124
Figure 6.3 Direct parameter calculations for maintaining tangential continuities (example): (a) planes are the critical feature; (b) cylinder is the critical feature.	125
Figure 6.4 Adaptive meshing.	127
Figure 6.5 Decal texture extraction and projection.	131
Figure 6.6 Edge extraction through erosive technique.	134
Figure 6.7 Genus increase operation.	135
Figure 6.8 Genus reduction operation.	136

Figure 6.9 Non-structure-altering edits with tangential continuity conditions.	139
Figure 6.10 Non-structure-altering edits with large movements.	139
Figure 6.11 Non-structure-altering edits – supplementary examples.	140
Figure 6.12 Structure-altering edits – step-by-step: (a) decal texture extraction; (b) original location edge extraction and mesh adaptation; (c) new feature location decal projection and mesh adaptation.	141
Figure 6.13 Structure-altering edits.	142
Figure 6.14 Structure-altering edit with genus increase.	143
Figure 6.15 Structure-altering edits – supplementary examples.	144
Figure 7.1 Implementation setup of proposed flexible hybrid CAD modeler.	150
Figure 7.2 Case 1: basic modeling via proposed hybrid CAD modeler.	151
Figure 7.3 Case 2: complex modeling via proposed hybrid CAD modeler.	153
Figure 7.4 Case 3: complex modeling via proposed hybrid CAD modeler – feature-based model editing operations first.	155
Figure 7.5 Case 4: complex modeling via proposed hybrid CAD modeler – mixed usage.	156
Figure 7.6 Case 5: hybrid CAD modeler supplementary example 1.	157
Figure 7.7 Case 6: hybrid CAD modeler supplementary example 2.	157

List of Symbols

Symbol	Definition
α_{cone}	Cone feature defining angle
A_{cone}	Cone feature axis vector
$A_{cylinder}$	Cylinder feature axis vector
A_{plane}	Plane feature axis vector
O_{cone}	Cone feature center point
$O_{cylinder}$	Cylinder feature center point
O_{plane}	Plane feature center point
O_{sphere}	Sphere feature center point
O_{torus}	Torus feature center point
$P_{edge}(j)$	Feature intersection points
R_{cone_max}	Maximum cone feature radius
R_{cyl}	Cylinder feature radius
R_{sphere}	Sphere feature radius
R_{torus_major}	Torus feature major radius
$\hat{n}(j)$	Per-vertex normal
κ_m	Mean Curvature

List of Abbreviations

Abbreviation	Definition
CAD	Computer-Aided Design
STL	Stereolithography

Acknowledgements

I would like to express the deepest of gratitude to my research supervisor Dr. Hsi-Yung Feng who has become one of my most important mentors in both academia and life. My years with him as a graduate student have made me a better researcher and a better person. He has pushed my critical thinking level well beyond what I believed was possible. This ability to process information more critically has a profound effect on my work and my life. I truly appreciate all his support throughout the years.

I would also like to acknowledge the financial support of Natural Sciences and Engineering Research Council of Canada (NSERC) and the Four-Year Fellowship for Doctoral students from UBC Faculty of Graduate and Postdoctoral Studies. Similarly, I wish to thank MeshLab project, AIM@SHAPE Repository, Geomagic, Sculpttris, Graphite, Stratasys and Laser Design Inc. (LDI) to facilitate the completion of my work.

Additionally, I like to extend my gratitude to my supervisory committee for their valuable input, especially at the thesis proposal stage. Their diversified points of view have helped tremendously in shaping this work.

Sincere thanks are due to my family for supporting me without question throughout my graduate studies. In particular, I am grateful to my brother and girlfriend for keeping me sane (or not) in the past few years. I love you all!

The only person you should try to be better than, is the person you were yesterday.

Chapter 1: Introduction

Modern computer-aided design (CAD) modelers make use of the latest computer technologies to assist users in geometric design, creation and representation of engineering parts. The 3D virtual computer environment aims at much improved user-to-model interactions and at the same time, significantly facilitates information extraction and design analysis. Engineering models are no longer represented by sets of complicated and often ambiguous 2D drawings, but rather, they are represented by non-ambiguous 3D computer models. Due to the effectiveness of current CAD technologies, it is used comprehensively in practice ranging from automotive and aerospace industries to a variety of biomedical and architectural applications [1-3]. It is also extensively utilized in the computer animation/movies industry [4].

The novelty for CAD stems from the 3D environment itself. Virtual models are quickly mocked up at the expense of only computational power and designer work hours. By possessing the capability to represent an object in a virtual 3D environment, users are able to interact with geometric design models, which were previously not possible [1]. The 3D virtual CAD environment gives the users ways to handle the geometric model that is more “real life” like. Such interactions made modeling easier and thus, design iterations faster.

However, the novelty (for geometric manipulation) does not stem far from the environment itself. Models for the current CAD systems are represented as a combination of associated features with individualized mathematical definitions [5,6]. This makes the design model modification procedure rigid and cumbersome, causing CAD unable to maximize its ease-of-use for geometric design modeling. Geometric design modeling is restricted by the parametric

representation which leaves the users with complicated CAD systems. Even with newest Direct Modeling capabilities, outlined in the next section, models are still composed of sets of associated features with unique feature definitions, making design alteration difficult. The interactions with the model are improved for the users but the modification flexibility is still insufficient for design modeling. The current CAD systems require a great deal of expertise from the users in order to efficiently translate geometric designs and design changes envisioned by the users on to the computer screen.

Hence, this thesis proposes a triangle mesh based hybrid CAD modeler to combat these issues. The models are no longer represented through a composition of restrictive individualized separate feature entities but are represented by a flexible homogeneous model representation (triangle mesh) with features information imposed as a separate but associated information layer. By utilizing a homogeneous model representation with a separate associated feature information layer, modeling flexibility is greatly increased while modeling precision and accuracy are appropriately retained. The triangle mesh can be altered to represent any desired geometry with the feature information layer adapting to suit. The notions of features do not restrict the editability of the models. This allows the integration of flexible model editing capabilities (that are not realizable in the current CAD modeling system) with that of the current parametric Direct Modeling editing scheme, creating the flexible hybrid modeler.

With the proposed hybrid modeling system, the users can either freely manipulate the model to the desired geometry without any restrictions or can directly push-and-pull the existing features to the locations of choice. The hybrid modeling system would transparently and automatically identify and idealize the geometries of these edited regions, allowing the consistent realization of

ideal 3D parameterized design models. The end products of the proposed hybrid modeling scheme can be seen as equivalent to that of the current parametric CAD models (feature information is retained and accuracy and precision are guaranteed), but the path taken to derive the model is considerably less restrictive. With the proposed hybrid modeling system and its integration of previously unrealized “freely” editing capabilities, the users now have the ability to model beyond the rigidity imposed by the current parametric CAD modelers. The advantages and the necessity of the proposed approach is further outlined in the subsequent section via the short-comings of the current CAD systems.

1.1 Current CAD modeling technologies

Currently, CAD modelers are predominately one type, Parametric Feature-Based Modeling. This form of geometric modeling has dominated the CAD scene for the past two decades and have a very specific technique of modeling which the user has to follow [7,8].

For parametric feature-based modeling, the user creates the model through the idea of “design intent”. Rather than focusing on the geometry, the user focuses on the intellectual arrangements of features and dimensions [8-10]. Feature types and parameters and inter-feature relationships are explicitly outlined and precisely constrained in the parametric modeling systems. This brings intelligence and control to the practice of building CAD models.

The paradigm has been viewed as one of the most powerful method of creating geometric design models since it fundamentally possesses the trait to “put a number on everything”. This allows geometric design model to be an accurate digital reflection of the desired design model [11,12]. The features on the models are defined through accurate mathematical expressions, which

suitably facilitate the encapsulation of design information. However, parametric modeling is not built with the action of geometric design in mind, where objects would undergo numerous number of design iterations before arriving at the final product. By utilizing individualized mathematical-equations-based model definition to insure models carry the required information, the ease and flexibility in geometric modification is lost. Model geometries are constrained by the types of features and number of features on the model. Typically a feature on a model is not readily modified to a different feature type and the number of features is not straightforwardly increased or decreased without cumbersome modeling or remodeling processes. If the model is a composition of a sphere and a block, as shown in Fig. 1.1, it cannot be modified to the shown arbitrary shape. Features have to be deleted and remodeled in order for the object to arrive at the desired geometry. In the case illustrated, the complete geometry is to be remodeled given the current parametric feature-based modeling paradigm. This is a severe hindrance in conceptual work and in design modifications during design iterations.

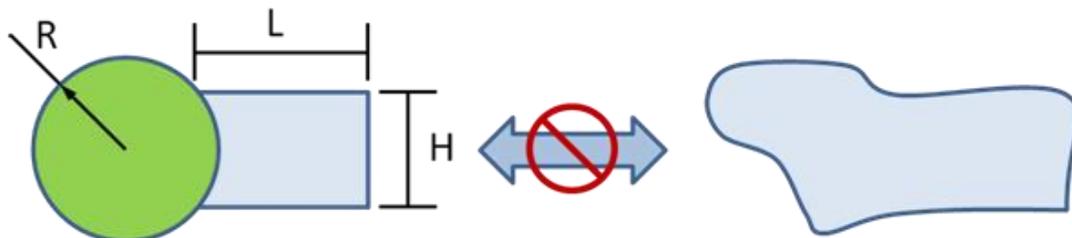


Figure 1.1 Difficulty in geometric manipulation due to individualized features.

Such issue with geometry modification is highly noticeable in the automotive industry. A majority of the shape development work of a vehicle is performed via automotive modeling clay due to its modeling flexibility. The geometry of the vehicle is continuously modified through

clay until the intended design of the vehicle is reached. The shape is then digitized and used to construct the CAD models necessary for the downstream development of the vehicle. In fact, such issue exists universally and has become very prominent in the design world as of late. With the fast emergence of 3D printing and 3D shape capture devices [14] for both fast prototyping and end user part manufacturing purposes, the need for fast CAD design changes has become vitally necessary. Prototypes during design development are no longer mocked up and edited physically until the design is ready to be translated into CAD models. Prototypes are mocked up in the digital world and rapidly printed for testing purposes. Design changes required are implemented in the CAD software and the new design is printed for further physical testing. Optimal design is found through the iteration of these steps.

With 3D printing and 3D shape capture technologies, large number of design variations are quickly realized in the physical domain enabling enhanced searches for the optimal design. With the current new trend of fast iterative design enabled by 3D printing and 3D shape capturing devices, efficient CAD modeling methods are crucially necessary for optimizing the complete design process. Design changes needs to be quick and straightforward. Model modifications should be flexible and undemanding. However, this is far from the case for the current CAD systems. For the current CAD systems, time consuming remodeling with high expertise requirement is still a necessary part of the geometric design process. Geometric changes that involve alterations to the model's design intent usually leave the designer with the cumbersome job of remodeling a significant portion of the model. Most models found in libraries can only be efficiently re-used if the new model's design intent is extremely similar to the original. If not,

inflexible CAD software is used to perform the modifications. As a result, the current parametric CAD modeling paradigm serves as a hindrance to geometric design and modeling.

To better outline and illustrate the inherent issue regarding model modification possessed by the current parametric modelers, the modeling flow of current systems with a modification approach is shown in Fig. 1.2. The first step is to load a similar object from the model library with the highest number of usable parametric features. The unusable features are deleted while the usable features are kept. New bulk parametric features (sphere, planes, cones, etc.) necessary to complete the part are then created through the specification of the individual feature's type and their corresponding feature parameters. The relationships (constraints) of these new bulk features with respect to other features on the model are also established via the user. The new bulk features' feature parameters and constraints have to be associated appropriately in this step [13], in order to establish the proper design intent. The newly created bulk parametric features and the usable features are combined to form the associative bulk feature-based Model. The needed detail features (blends, chamber, etc.) are then added through the specification of their types and their corresponding parameters. These detailed features refine the boundaries and the connections between the features. Again, detailed feature parameters and constraints have to be associated appropriately to further completely the design intent. The result is the associative detailed feature-based model. At this stage, the object can proceed to become the associative final feature-based Model if the design is deemed satisfactory (with possible minor parameters adjustment) or the object can proceed to be modified, in terms of feature type and numbers, if the design is deemed unsatisfactory. If modification is needed, which is the case for most models in their design stages, the features-to-be-changed and their associated features are removed. The

new features and their associated features needed in place of the old is modeled from ground up and combined to the existing parametric features. The detailed feature operations are then performed again on the new associative bulk feature-based model. This inefficient process of deletion and recreation repeats during the design stage until the final design is derived.

Parametric Feature-Based Modeling

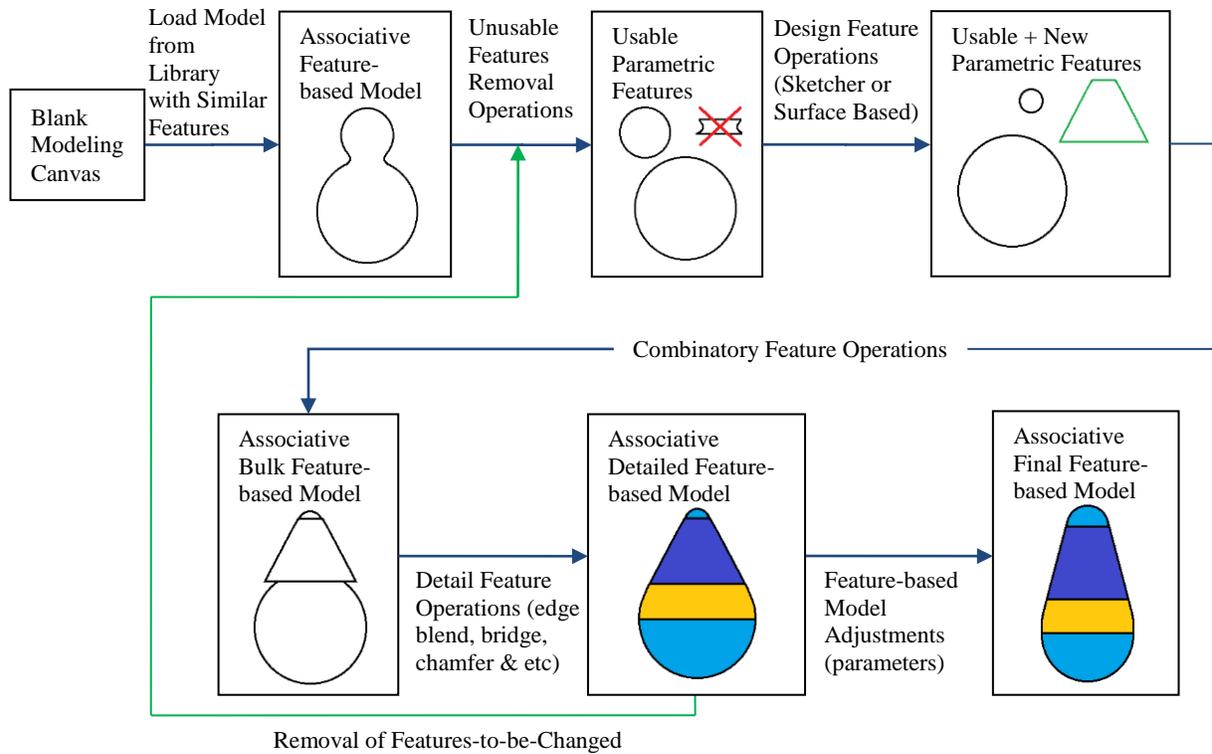


Figure 1.2 Parametric feature-based modeling.

The need to constantly remodel significantly hinders design work [15,16]. However, it is a necessity with the current parametric modelers since the features are strictly represented by separate mathematical definitions. Each feature in the model is considered as an entity of its own. The geometry of the model is not treated as a homogenous entity but rather a combination and intersection of mathematical definitions. In order to change models with such setup,

redefinition is always required. For example, a simple change of a cylindrical feature to that of two cones tangentially continuous to one sphere is not realizable without remodeling. Thus, the basis of the parametric modeling paradigm is limiting the growth of geometric modeling. Design modifications are labor intensive rituals resulting in time-consuming design iterations. Most often, these feature-based parametric modelers are utilized to create the final geometry of the design rather than to perform the design itself. Parametric feature-modeling is a powerful documentation-based CAD modeling paradigm. It is, however, a very rigid approach to geometric design and modeling.

In recent years, Direct Modeling operators in various CAD package have emerged to battle the rigidity of the parametric CAD modeling paradigm. Siemens NX introduced Synchronous Technologies [17] while PTC Creo introduced FMX [18] and Autodesk introduced Inventor Fusion Technology [19]. Various other CAD software companies have also presented their own form of direct modeling operators which, as mentioned previously, allow users to directly manipulate the CAD models. Feature parameters on a previously created model are interactively modified via “push-and-pull” techniques (illustrated in Fig. 1.3). Features can be moved and resized simply through grabbing the features-of-interest and moving them, which was not possible prior to direct modeling operators. This type of interactions increased the ease-of-modification for previously created models. Users interacts directly with the model instead of menus and dialogs. However, direct modeling is only just an extension of the current parametric modeling paradigm and thus suffers the same issues. The adherence of features to its feature types is still necessary. Features are still treated as separate entities leading to limited and incomplete user modeling interactions. Features on a previously completed model can be

modified simply, but the modification is limited to and bounded by its feature type and design intent. When feature type change is required, local remodeling is typically used to perform the alteration. This means that a revolved feature's defining curve has to be redrawn, and/or its revolved operator has to be changed in order to facilitate the correct design modifications. Direct modeling better the ease-of-interaction in modeling; however, modification shown in Fig. 1.1 is still not possible. Direct modeling is still only a part of the parametric CAD paradigm; thus, highly flexible geometric modifications are not possible.

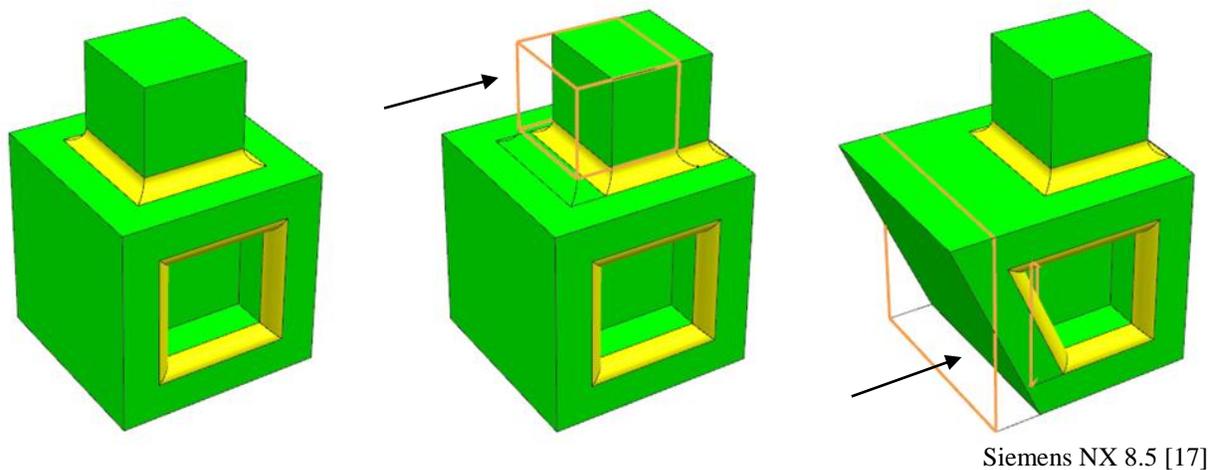


Figure 1.3 Direct modeling.

Another approach that has gain traction to also better the ease-of-interaction in the parametric CAD modeling paradigm is sketch-based modeling [20-23]. Instead of dealing with complicated menus and dialogs or interacting directly with features via push-and-pull operators, the users use a stylus with a screen or drawing pad. To create a model, the users simply draw the intended design on the screen as they would on paper. Different sketch-based modeling algorithm calls for different number of sketches for a single model, but overall, these algorithms all attempts to extract a set of defining curves that fully describes the design-of-interest. The system translates the sketches into a set of fitted curves which are then used to create the features. These features,

created individually, together represent the final model: parametric CAD model. For sketch-based modeling approach, the ease-of-interaction for model and feature creation have been improved compared to the traditional approach of using mouse and keyboard. However, it is still based on the current parametric CAD systems where modification flexibility is lacking significantly. The created model can still not be modified freely by the user. The model is still heavily bounded by individualized feature bounded/defined by extracted sketch-curves.

Due to the rigidity of the current parametric CAD systems, designers have looked to use 3D computer graphics modelers [24-27] for the needed modeling flexibility. These modelers, such as 3ds Max [28], Maya [29], Blender [30] and MeshMixer [31] are developed with modeling efficiency and flexibility in mind instead of modeling precision and accuracy. Therefore, designers have increased freedom in shape manipulation via these modelers compared to the current parametric CAD modelers. Models can be created and modified efficiently without the restriction of individualized features. The lack of features allows users to quickly derived visually pleasing designs. The rigidity of parametric modeling is not seen in these computer graphics modelers.

However, the gain in modeling flexibility for 3D computer graphics modelers is done at the expense of well-defined design features [32-35]. Models do not carry feature information that not only guarantees model precision and accuracy but facilitates downstream engineering analysis and simulation. Geometry that closely mimics prismatic mechanical features seen in the parametric CAD modelers can be created and used in 3D computer graphics modelers, but these features are only approximations without mathematical definitions [33,35]. The mathematical

formulations that appropriately define these features are non-existent, making these “pseudo” features only rough visual approximations.

Without feature definitions, these pseudo features become inexact finite entities. They are inaccurate and individually bounded by false feature boundaries which should not exist [33,35]. Feature boundaries are the product of feature intersection and therefore cannot be defined for an individual feature. For example, a cylindrical surface is infinite in the direction of its defining axis. Its height/length is only defined when two features, not tangent to the cylindrical surface’s axis, intersects the cylindrical surface. For a cylindrical pseudo surface feature, its height/boundary is pre-defined, even though no other features intersect with it. Therefore, the pseudo features seen in 3D computer graphics modelers do not behave as features should. Pseudo features can only intersect other features within its finite boundary while parametric features can intersect other features anywhere within its definition. Thus, due to this finite characteristic caused by the featureless representation, direct modeling techniques seen in the parametric paradigm is absent in the current 3D computer graphics software. Interactions of features derived through feature definitions is not possible.

These 3D computer graphics modeling packages offers flexible shape manipulation abilities at the cost of not only modeling precision and accuracy but also feature-based model modification capabilities. Without the existence of features and their corresponding definitions, users can only create approximate models that are visually good. Model precision and accuracy cannot be guaranteed and downstream analysis and simulated cannot be readily facilitated for common mechanical parts.

To appropriately facilitate mechanical engineering design, the notion of (prismatic) features is necessary. However, current CAD systems handle features in a way that drastically lowers the modeling flexibility. CAD has always attempted to aid and streamline design processes, but with its current approach to geometric manipulation, such goal is hindered by its own inner workings. Even with high-level of learning and training, parametric CAD systems are still difficult to use to translate a 3D design idea into a 3D digital computer model. Users are still very often left “scratching their heads” wondering how best to realize their designs in the CAD system they are using. The current CAD system functions well for design documentation and performs adequately for model creation but it is far from satisfactory in facilitating flexible 3D modeling for mechanical parts.

1.2 Proposed hybrid modeling scheme

Consequently, to realize a flexible CAD geometric modeler the very basis of CAD systems, model representation, needs to be revamped. There is no doubt that features, its parameters and constraints, needs to be part of design modeling, but it is not necessary to handle it in such a way that flexibility in geometric modification is lost. As mentioned previously, the current parametric CAD systems fall short when it comes to geometry editability largely due to its model representation scheme and its approach to feature management. By utilizing parametric mathematical representation with each feature individually realized, 3D shape modification flexibility is greatly sacrificed [36]. It is not possible to freely manipulate the model of interest. Hence, rather than attempting to increase the geometry modification capability of a model representation that is fundamentally limited, an alternate scheme that does not rigidly treat models as compositions of separate feature entities is proposed: the triangle mesh based hybrid

representation. The hybrid representation scheme utilizes the flexible triangle mesh as the base representation with prismatic features imposed as an associated additional information layer in order to give rise to a flexible, but still featured, modeling scheme.

Triangle mesh is chosen as the base because it can represent any given physical object in a discrete sense without the need of different geometric formulation for different shapes. Figure 1.4 shows two objects represented by triangle meshes. The only difference between these two triangle meshes is the spatial location of its triangles vertices.

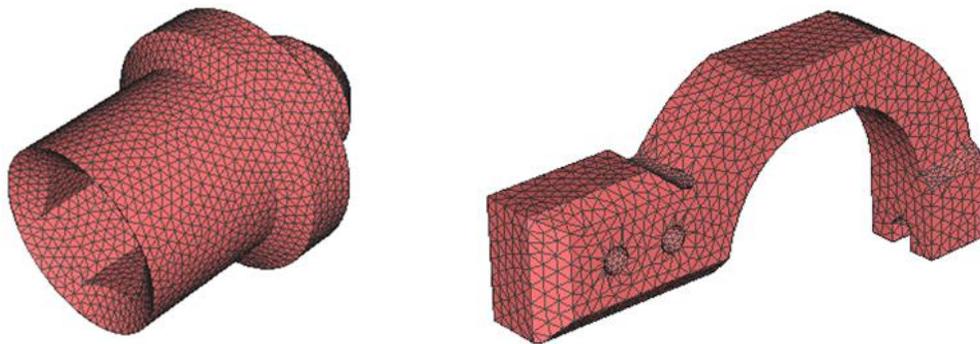


Figure 1.4 Flexibility of shape representation via triangle mesh.

Thus, such piece-wise linear representation offers exceptionally high shape representation and modification capability. The triangle mesh geometry is not strictly restricted by mathematical formulations. The mesh is an entity that can take on any shape required by the design [37]. When formulation is required, local definition of the user's choice is temperately created and imposed on the mesh [38]. There is no restriction on the type of formulation that can be imposed on the mesh. Complete geometric freedom is available for design purposes via triangle mesh. Furthermore, triangle is the simplest polygon primitive which is always planar and convex. It facilitates straightforward triangle rasterization (conversion of 3D model to viewable image) and

linear interpolation, which allows for the development of efficient geometric processing algorithms. Also, generation of triangle mesh from point cloud derived from 3D geometric capture devices is inherently simpler. Various fully automated mesh generation from point cloud algorithms exists in literatures and industrial software [39]. Triangle mesh is also the prime model representation for 3D printing and various engineering machining simulation software packages. Thus, triangle mesh is the base model representation for the proposed hybrid CAD modeler.

Another promising candidate is quad mesh [40]. Quad mesh is very similar to triangle mesh in terms of model representation benefits; however, it does not straightforwardly facilitate efficient geometric processing algorithms. Compared to triangle mesh, similarly purposed algorithms in the quad mesh domain is more computationally demanding. Furthermore, generation of quad mesh from point cloud is much more difficult. User often needs to provide boundary conditions such as feature curves or singularity information. These information greatly affects the quad mesh quality. Various quad mesh generation methods also suffer from robustness issues when processing noisy inputs. Interactive user repair is more extensively required for quad mesh compared to triangle mesh. Overall, quad mesh is inherently more difficult to process compared to triangle mesh due to its non-planar and non-convex characteristics. The requirement for the quads to be nearly rectangular with the need for them to follow prescribed directions in order to achieve proper arrangement or to create suitable patches makes geometric processing problematic [40]. Triangle mesh is a much more developed mesh representation technique with an abundance of efficient and robust geometric processing algorithms making it the base model representation choice for the proposed hybrid CAD modeling scheme. Nevertheless, quad mesh

can be the base representation of choice in the future for the proposed hybrid modeler if the above outlined issues are addressed robustly. The algorithms proposed in this chapter can be adapted to quad meshes, since the methodology for this work is mesh driven.

By adapting triangle mesh representation as the base of the hybrid representation, the model is no longer strictly defined by separate mathematical feature formulations. The restrictive representation is removed and replaced with connecting vertices that can represent any physical shape. Features is imposed as an additional information layer in order to bring about model precision and accuracy. With such hybrid representation, both shape modifications free of feature-based restrictions and feature-based modifications with feature-based restriction is realizable, providing users with increased geometric modeling flexibility. An abstract illustration of the difference between the current CAD model representation and the proposed triangle mesh based hybrid representation with associated feature information is shown in Fig. 1.5.

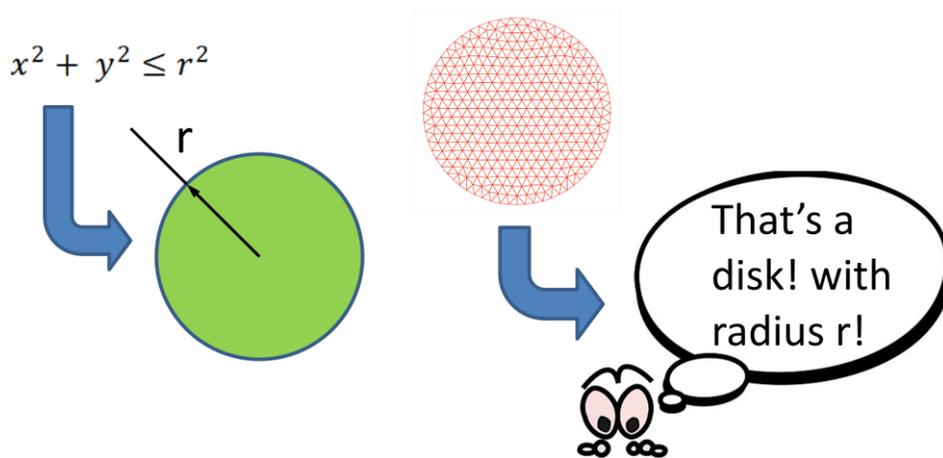


Figure 1.5 Feature as model information rather than model definition.

Therefore, a flexible triangle mesh based geometric modeling scheme as shown in Fig. 1.6 named Triangle Mesh Based Hybrid CAD Modeling Scheme is proposed. This modeling scheme first imports a water-tight triangle mesh model from a mesh library and subsequently applies feature segmentation and identification to bring the notion of feature on to the model. If the input model possesses imperfections, feature idealization is applied. The model is idealized (perfected) based on the identified features and their continuity constraints. Once this is completed, the idealized triangle mesh possesses feature information such as feature types and parameters and thus, a feature-based mesh model is derived. Such model is comparable to the associative feature-based model from parametric feature-based modeling (Fig. 1.2) or in short, such model is comparable to a parametric CAD model. The user can modify the idealized object based on parameters just like the current CAD systems. However, the user can also modify the model without abiding to any feature-based constraints. Unconstrained feature-free model modification can be applied to any portion of the model in order to freely alter the geometry. The user can deform/change any portion of the model through mesh deformation operators without dealing with constraints or feature definitions. Unconstrained modeling interactions with the mesh model brings the model close to the desired shape.

Triangle Mesh Based Hybrid CAD Modeling Scheme

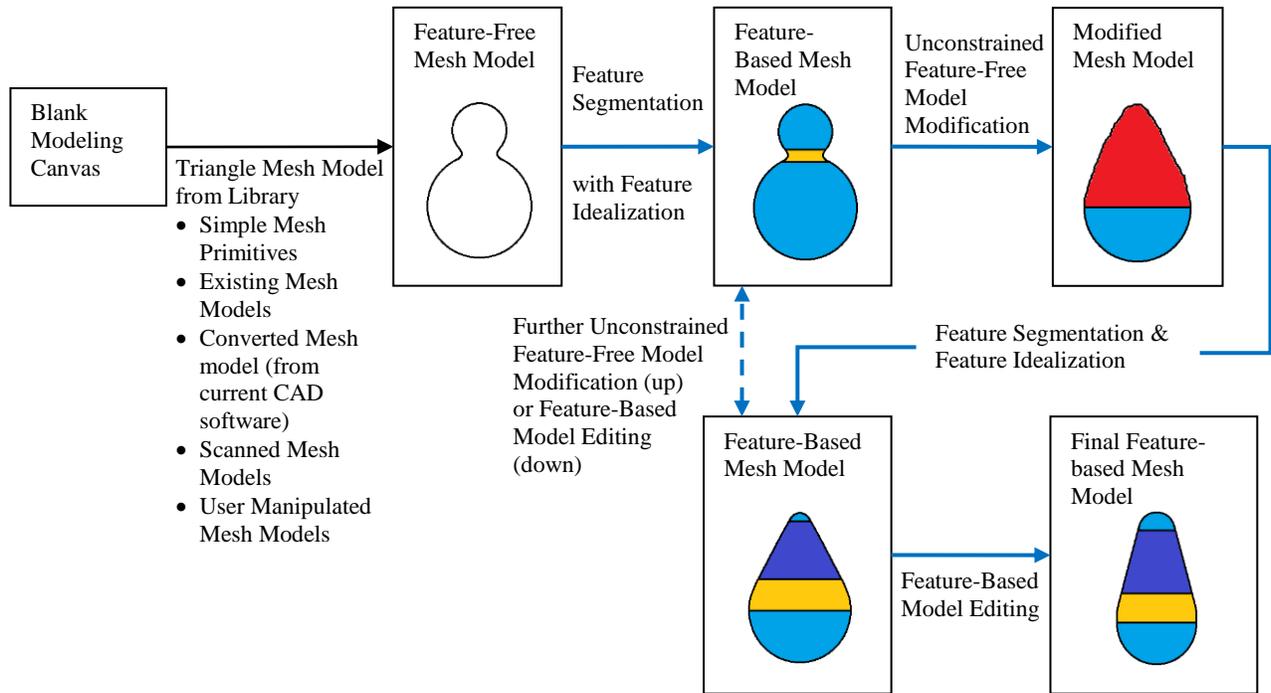


Figure 1.6 Triangle mesh based CAD modeling scheme.

Once such unconstrained feature-free modifications are imposed, feature segmentation and identification is applied on the modified portion of the mesh to extract the user intended geometry. The type and parameters are automatically determined. The modified mesh, which possesses imperfections caused by the user during the unconstrained modification stage, is idealized to that of the identified user intended geometry. Further modification can be performed quickly and efficiently either through CAD like modeling manipulation or through unconstrained feature-free model manipulation. Once the design is deemed as intended, the mesh model becomes the Final Feature-Based Mesh Model.

By handling the notion of features as geometrical information rather than individualized geometrical definitions, the users are not constrained by feature definitions and can modify the

model much more freely than previously possible. At the same time, the notion of feature and their parameters are not lost. With the proposed triangle mesh based hybrid modeling scheme, the notion of features is retained while a pro shape modification modeling scheme that offers unconstrained shape manipulation capability is realized. Feature-based modification is still present while unconstrained feature-free shape manipulation is now available. Modeling flexibility is increased without sacrificing precision and accuracy. Less restrictive design alterations can be achieved through the proposed triangle mesh based hybrid modeling scheme.

1.3 Primary components

To best outline the proposed modeling scheme in detail, the four main components of the modeling scheme are elaborated (corresponding to the operations shown in Fig. 1.6). Each of the components and their corresponding objective necessary to realize the proposed modeling scheme (and thus the modeler) is summarized below. The subsequent chapter then expands on these components by first outlining the relevant state-of-the-art methods then followed by the proposed approaches themselves in the later chapters.

- Feature Segmentation: autonomously and robustly divide input triangle-mesh model geometry and unconstrained feature-free user manipulated mesh patches into elementary prismatic features (planes, cylinder, sphere, cone and torus). Feature continuity conditions also need to be identified.
- Mesh Idealization: automatically perfects the input geometry and the user manipulated patches based on the segmented features and continuities. Features and their intersecting edges all need to be perfected while abiding to prior detected continuity conditions.

- Unconstrained Feature-Free Model Modification: Straightforwardly allows user to perform unconstrained feature-free model modifications. The used operators are borrowed from the computer-graphics community.
- Feature-based Model Adjustments: Robustly enables feature-based model editing for triangle mesh models. Parametric CAD Direct Modeling like adjustments are realized for the mesh regime.

Please note that since the proposed novel modeling scheme (modeler) is in its initial stage, the scope of the four primary modules, in terms of geometry, is restricted to the processing of prismatic engineering design models. Only prismatic features of planes, cylinders, cones, spheres and tori, are considered in this work. Also, only positional and tangential continuity constraints are handled in this thesis. A modeling scheme capable of processing comparable geometry as the current parametric CAD modelers with the proposed much increased modeling flexibility is very challenging and beyond the scope of this thesis. Such implementation/realization is part of the future work. This thesis focusses primarily on demonstrating the feasibility and capabilities of the novel hybrid modeling scheme on prismatic mechanical parts.

Chapter 2: Literature Review

With the hybrid modeling scheme and its four modules (Feature Segmentation, Mesh Idealization, Unconstrained Feature-Free Model Modification and Feature-Based Model Editing) and their goals outlined, the following sections review the relevant literature pertaining to each of the four modules. The suitability of the existing methods to the proposed work is analyzed here. If a suitable candidate is found for a specific module, the subsequent chapter dedicated to the specific module elaborates on the chosen method. If an appropriate candidate does not currently exist for a specific module, the corresponding subsequent chapter elaborates on the proposed approach.

2.1 Existing mesh segmentation methods

As outlined previously, one of the first tasks in realizing the proposed flexible hybrid modeling scheme is to acquire the capability to bring features into the featureless triangle mesh model representation. A robust method to deduce and extract prismatic features directly from the triangle mesh model geometries is necessary. It is with such segmentation capability that the modeling precision and accuracy seen in parametric feature-based CAD modeling can be readily reflected and combined with the flexibility of mesh modeling techniques. Therefore, a robust and automatic segmentation method is required for the proposed hybrid modeling scheme.

For the current triangle-mesh segmentation algorithms, two broad categories exist: surface-based methods and part-based methods. The surface-based methods segment the input triangle mesh models into a set of connected surface patches with each individual patch representing a

particular feature type (for example, cylindrical and planar surfaces). For the part-based methods, the algorithms segment the input mesh into sets of feature that together represents a part (for example, cylindrical and planar surfaces which together represents a cylinder). In order to maximize modeling flexibility, surface-based segmentation is of particular interest. Part-based segmentation does not allow for maximum modeling editing flexibility since the input models are not reduced down into its elementary form. Parts can always be further decomposed into features. To allow for maximum geometric manipulation capability with maximum shape control, parts need to be decomposed into its elementary form. Each elementary feature on the mesh needs to be discovered in order for the modeling system to enable the highest level of control. Therefore, for the purpose of this research, only surface-based segmentation techniques are considered since their goal is to decompose the triangle mesh models into their most basic feature compositions. Segmentation methods in this thesis refer to surface-based segmentation. Furthermore, please note, the words cylinder, torus, sphere and cone in this thesis represent features not parts. On occasion, cylinder and cylindrical surfaces are used interchangeably.

In the past decade, a good number of surface-based triangle mesh segmentation methods are reported in the literature. These methods, however, need to be carefully examined in order to confirm their applicability to the present work. Referring back to Fig. 1.6, it is shown the proposed method processes various input triangle mesh models. The inputs ranges from ideal triangle mesh models directly outputted from CAD or computer graphics software to user manipulated (unconstrained) triangle meshes and to scanning-derived triangle meshes. Although from different sources, the triangle mesh models have similar characteristics. As shown in Figure 2.1, the different types of triangle mesh inputs are seemingly just variations of the same mesh

model. They only differ in quality. The mesh triangle size and aspect ratios are all similar. The triangle mesh derived directly from CAD packages and computer-graphics software effectively represent the ideal (best quality) mesh model types, while the user manipulated and the scanning-derived mesh models represent two of the least ideal mesh model types: with the scanning-derived mesh model being the worst in terms of high frequency defects and the user manipulated mesh model/area being the worst in terms of low frequency defects.

Thus, in order to fully realize the hybrid modeling scheme, the segmentation algorithm needs to target the two worst case scenarios, scanning-derived meshes and user manipulated meshes. Since the different quality triangle mesh inputs are similar, just variations of the same geometry, it is believed that by targeting the worst variations of the input triangle mesh types, the developed algorithm will perform well for the better quality input meshes. The geometric measures on the better quality triangle mesh inputs can only be healthier than the worst quality triangle mesh input. This hypothesis will be tested and verified in the case studies section for the proposed segmentation method. The existing segmentation techniques of interest are thus those developed for scanning-derived mesh models and user manipulated mesh models/regions, two of the most difficult to segment mesh types.

However, to date and to the knowledge of the author, segmentation method aimed specifically to process unconstrained feature-free user manipulated mesh models/regions for engineering objects do not exist explicitly. This type of mesh models contains non-uniform low frequency surface undulation and chamfered edges caused by users' freehand manipulation as illustrated in Fig. 2.1 [41]. Various works focuses explicitly on the segmentation of computer-graphics-based engineering triangle meshes and scanning-derived engineering triangle meshes, but not

unconstrained feature-free user manipulated engineering triangle-mesh models. This is understandable since the need to reliably segment user manipulated triangle-mesh model for engineering design purposes is a product of the proposed hybrid modeler. The goal to combine flexible triangle-mesh modeling techniques with the precision and accuracy of parametric CAD approach resulted in the need for segmenting user manipulated mesh models/regions. Therefore, there are no explicit methods aimed to segment unconstrained feature-free user manipulated triangle-mesh engineering models in current literatures. With this, the literature review puts its main emphasis on the state-of-the-art scanning-derived mesh segmentation techniques for engineering parts; one of the two worst mesh input variations for the hybrid modeling scheme. Computer-graphics based segmentation methods are also reviewed as part of the existing method section for completeness. But as mentioned previously, the segmentation work in this thesis is targeting the most difficult to segment input triangle-mesh models (scanning-derived triangle-mesh engineering models and unconstrained feature-free user manipulated triangle-mesh models) in order to ensure the segmentation method developed functions well for all mesh variations seen in the proposed hybrid modeling scheme.

When reviewing current work regarding scanning-derived mesh segmentation methods, care is required. Scanning-derived meshes have specific and unique characteristics such as sampling aliasing (by-product of 3D scanning) known as the chamfered-edge effect (sharp edges not being sharp) and high frequency defects, scanning noise. A scanning-derived mesh is also characterized by fairly uniform tessellation throughout the mesh. As shown in Figure 2.1, these characteristics separate the applicable methods for segmenting scanning-derived meshes from those for segmenting chordal-error based stereolithography (STL) meshes and computer graphics

meshes. Quite often, methods developed for segmenting STL or computer graphics meshes claim to be useful for scanning-derived triangle mesh models. In fact, they only have limited applicability in segmenting scanning-derived meshes. Inappropriate or even invalid segmentation results are commonly produced. The following section outlines and review the works that have made claims regarding segmenting scanning-derived mesh models.

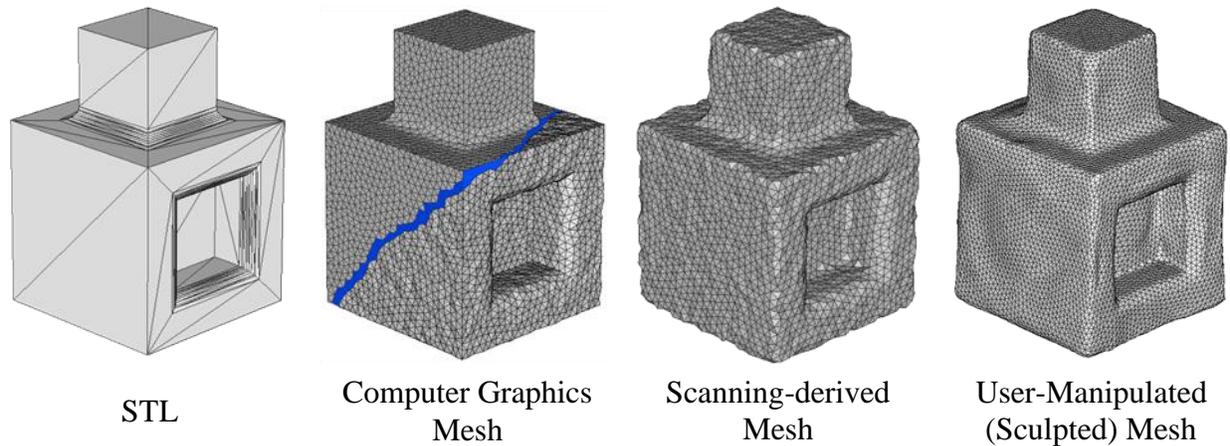


Figure 2.1 Typical triangle mesh types and their characteristics.

2.1.1 STL meshes

One of these sometimes misleading types of work regarding segmentation are works developed for STL meshes. STL is a widely used format to represent triangle mesh models with the triangle mesh being created from a parametric CAD model based on specified chordal errors. To be consistent with the specified chordal error constraint, planes would consist of large triangles whereas highly curved features would consist of small triangles. Segmenting STL meshes is not the same as segmenting scanning-derived meshes. Some researchers have referred to STL meshes as CAD meshes and introduced segmentation algorithms specifically tailored to STL meshes [42-45]. These algorithms use the triangle size and aspect ratio as the segmentation

measure and this makes the algorithms mesh-specific. The algorithms would fail when the input mesh is composed of triangles whose sizes and aspect ratios cannot be used to infer feature information. This is the case for the input models which this work processes. The triangle meshes for the proposed hybrid modeling work possess triangle size and aspect ratios that cannot be used to infer its features. Consequently, STL mesh segmentation methods are not applicable for the proposed work.

2.1.2 Computer graphics meshes

For algorithms that are developed to deal with computer graphics meshes, the working principle is not dependent on the triangle size and aspect ratio. These meshes, as shown in Figure 2.1, possess characteristics that are much closer to those of scanning-derived meshes. As mentioned previously, computer graphics derived meshes can be treated just as a better quality variation of scanning derived meshes and is one of the inputs to the proposed hybrid modeling scheme. However, methods developed specifically for computer graphics meshes puts inadequate emphasis on chamfered-edge effect and noise. Some reported algorithms to segment computer graphics meshes do not account for the presence of chamfered edges and scanning noise at all [46-48]. Others only account for the noise but not the chamfered-edge effect [49-51]. For those algorithms that disregard the presence of both chamfered-edge effect and noise, segmenting a scanning-derived mesh results in a highly over-segmented mesh. For those algorithms that take into account the noise but not the chamfer-edge effect, segmenting a scanning-derived mesh results in poor feature and boundary extraction [51].

The chamfered-edge effect is indeed the key characteristic that differentiates a scanning-derived mesh from a computer graphics mesh. When combined with noise, it makes the scanning-derived mesh very difficult to segment. It renders the sharp features unsharp and likely causes their segmentation indicator values to be undistinguishable relative to its adjacent noisy features. Often, this leads to poor segmentation results with poor segmented patch boundaries. In addition, some computer graphics oriented methods are simply not applicable to common mechanical part geometry since they are specifically tailored to organic geometry [52,53]. If the methods outlined in this section are used for this work, much labor-intensive manual editing is needed [54-56].

2.1.3 Scanning-derived meshes

In order to deal with chamfered edges and noise (high-frequency defects) in scanning-derived meshes for CAD applications, existing segmentation methods employ fitting. Fitting, either global or local, is seemingly the most straightforward and natural approach when the mesh is non-ideal, especially since the goal here is to segment the mesh into distinct features. The mathematical expressions of the involved features are readily available and can be directly used to perform the feature segmentation. Various fitting-based methods were introduced in the past several years [57-61]. However, there are two fundamental issues with these fitting-based methods. The first and most important issue is that these methods are not completely automatic. Various user-specified case-specific parameters are needed to implement these segmentation methods. Users have to tune various parameters such as patch number, threshold values, and sensitivity in order to attain the appropriate results and this makes the segmentation process labor-intensive and not robust. Thus, these methods do not satisfy the automatic segmentation requirement necessary to realize the hybrid CAD modeling scheme. The proposed modeling

scheme is meant to be an automatic tool for CAD modeling where the user only needs to focus on geometric design. The second issue with the state-of-the-art methods is that detail features such as chamfers, blends, and fillets are often inappropriately segmented [60]. These detail features are usually grouped into adjacent features or segmented with inaccurate borders. This is due to the difficulty in setting suitable thresholds. Subsequent labor-intensive editing is needed to correct these incorrect segmentation results. Even so, fitting is still the most widely employed method to segment scanning-derived meshes for CAD applications since with much user input, acceptable results can be obtained. However, robust fully automatic algorithms have not yet been realized.

For completeness, there is another class of segmentation methods that is also relevant to the present work, the direct segmentation methods of point clouds without mesh reconstruction [62]. The basic principles of these methods are in fact very similar to those of the existing methods to segment scanning-derived mesh models. Effectively, the point cloud segmentation methods utilize temporary local meshes (neighboring points) to extract needed geometric information from the point cloud, while the scanning-derived mesh segmentation methods utilize a permanent mesh constructed from the same set of points to extract the geometric information. Since the point set is the same, the inherent characteristics of chamfered-edge effect and noise in a point cloud and a scanning-derived mesh are identical. As a result, existing point cloud segmentation methods [63-70] all suffer from the same issues as the existing scanning-derived mesh segmentation methods. The segmentation results are heavily dependent on user-specified parameters. Even for automatic methods such as those developed by Cai *et al.* [63] and Weber *et al.* [69], a good initial estimate

is required to ensure proper convergence and appropriate termination conditions are also needed to attain desired results.

Therefore, through the above literature review of the current segmentation methods, it is shown that a robust and automatic triangle mesh segmentation algorithm does not currently exist. The needed ability to, without user intervention, segment a triangle mesh model into its elementary prismatic features in order to realize the hybrid CAD modeling scheme has not yet been realized. Thus, a robust and automatic segmentation method has been developed and outlined in the Feature Segmentation chapter of this thesis. The ability to appropriately segment a mesh into its elementary prismatic features is a must for the hybrid modeling scheme.

2.2 Existing idealization methods

The second module necessary to realize the hybrid modeling scheme is Mesh Idealization. In order to transform imperfect meshes and/or unconstrained feature-free model modifications into perfect mechanical models, the hybrid modeling scheme needs to identify and update the imperfect meshes/modified mesh regions into prismatic features with correct relationships.

Existing studies on model idealization have been predominantly presented in the research field of reverse engineering as a constrained parametric surface fitting problem [70-74]. These studies generally involve finding a set of regularities that fully constraint the surface fitting problem for CAD parametric models without violating the solvability of the problem. Often in these studies, constraints (regularities) were added to the constraint problem one-by-one with the solvability checked for every addition. Once the system is fully constrained and solvable, a constraint solver is used to solve the problem. The end product is an idealized parametric CAD model. Some

variations of the work studies and uses the likelihood of appearance of regularities on engineering models [70]. The regularities are prioritized based on their likelihood of appearance and are added to the constraint problem based on the priority order. However, the priority only acts as a guide since the problems are often unsolvable if the priority is strictly followed [70]. To summarize, these existing studies place their main emphasis on ensuring the solvability of the constrained problem but not on guaranteeing the geometrical similarity and topological identity of the idealized model to the original model.

There were a number of reverse engineering research works attempting to idealize models through projections [75-79]. These methods segmented objects into sub regions that can be described by single 2D contours. Constraint 2D contour fitting is applied to each of these sub regions to determine these contours. The object is then reconstructed through extruding or sweeping these 2D contours. Due to the methods being based on projections, their applications in terms of shape has been limited. Vast number of prismatic engineering objects cannot be readily divided into projectable sub-regions. Furthermore, the methods only consider the inter-feature relationship within each sub-region. Continuities between sub-regions are ignored during the idealization process.

There were also methods that took a hybrid approach by combining 3D constraint surface fitting and 2D contour fitting [80,81]. By incorporating 3D constraint surface fitting, those sub regions of the model that were not readily projectable into 2D contours could be processed. However, due to the complexity of the problem, good initial conditions were required. The methods are extremely sensitive to initial conditions (feature parameters and constraints) and therefore, not robust.

For method developed strictly for meshes [82-84] non-parameterizing approaches are typically proposed. The models are perfected without identifying the underlying geometry and therefore are only visually ideal. Additionally, the end products of the idealization were still parameter-less mesh models. Parameter-less mesh models are not parameterized and thus, do not satisfy the requirement of an engineering design model.

As reflected by the review of the current methods, the needed method to assist users in creating the ideal mechanical model is non-existent. Existing studies either focus primarily on the solvability of the idealization problem or on the visual appearance of the final model. Precision and accuracy of the model is not guaranteed through these approaches.

2.3 Existing unconstrained feature-free mesh manipulation techniques

As for unconstrained feature-free mesh model modification, there is an abundant number of research studies currently available, most if not all, stemming from the computer-graphics domain [85-90]. A large number of these methods treat the meshes as homogenous bodies and deform them similar to rubberized entities in order to achieve natural and organic like deformations. This leads to approaches that attempt to preserve mesh volume and to minimize mesh distortion. Mesh entities are deformable such that specific physical characteristics are well mimicked such as the natural bending of a tree branch, the proper movement of an octopus' tentacle and the correct motion of human appendages. For animation purposes, these approaches are quite suitable. However, for the purpose of engineering design modeling, these methods are neither appropriate nor effective.

The objective for this work is not natural movement but rather, efficient shape alteration. The goal is to select a mesh deformation technique that effectively alters shape, or in other words, effectively changes mesh volume and performs mesh distortion. Therefore, the required method needs to target the change of rather than the preservation of mesh volume.

Three primary types of techniques exist in the mesh domain for the shape alteration purpose: parametric CAD like editing, sketch-based mesh editing and mesh sculpting. The parametric CAD like editing approaches essentially mimic the pre-direct modeling technique of parametric modeling. Users create models through the utilization of 2D drawings, combination of “pseudo” features and adjustment of mesh control points. Mesh models can be derived through any combination of these methods. This type of approach can be seen in all current 3D computer graphics modelers such as 3ds max [28], Maya [29], Blender [30] and MeshMixer [31]. Essentially the method simply resembles parametric feature-based modeling without utilizing mathematically defined features. The features or rather pseudo features are the individually created mesh entities that only visually look like the features. Due to the closeness of this type of modeling approach to current parametric CAD modeling, adding such operations to the proposed hybrid modeler only adds non-precise, inaccurate and duplicated modeling operations but does not add to the flexibility of CAD modeling.

Current parametric CAD modeling performs similar modeling operations with well-defined features, guaranteeing model precision and accuracy. Again, it should be noted that complete parametric CAD modeling operations are not fully implemented in the current prototype of the hybrid modeler that focuses on prismatic mechanical features only. However, non-prismatic features can be integrated at a later stage of the hybrid modeler development. Since similar

operations exist in current parametric CAD modelers and the proposed hybrid modeler uses a feature information layer to facilitate parametric CAD modeling, the integration of this category of modeling operations is possible. In short, direct modeling operations of parametric modeling is the only part that is to be included in the current hybrid modeler prototype.

Sketch-based approaches for mesh editing [91-93], on the other hand, add to the flexibility of CAD modeling. They allow users to straightforwardly create or modify a mesh object through sketching. Users simply draw on the screen the changes they wish to realize. The modeler alters the mesh based on the user drawing commands. This type of interactions with the design model is not a duplicate operation even though similar sketch-based approaches have already been developed in the parametric CAD regime, as mentioned previously. Due to the sketch-based editing operators being imposed on a mesh representation, the method is not constrained by the features. The sketch contours do not together transform into a rigid parametric CAD model but rather the contours assist in editing a flexible triangle mesh model. The mesh model can be altered into any shape realizable via sketching. However, sketching-based approaches have one significant drawback. The usage of sketch contours often leads to shape alteration ambiguity [93]. A set of sketch contours are frequently non-unique in terms of the shape they together represent. Multiple solutions can be produced, leading to ambiguity during shape alterations. User intervention is required in order to ensure accuracy of result. Nevertheless, even with the issue of ambiguity, if a sketched-based approach is used as the proposed hybrid modeling scheme's unconstrained feature-free model modification method, modeling flexibility will be increased for engineering CAD modeling. Unconstrained feature-free modification is achieved.

Fortunately, this is also true for a less ambiguous approach, the mesh sculpting. Mesh sculpting methods [37,94,95] aim to modify the mesh in a way that is similar to working with physical clay. Instead of modeling through sketching or rather, the usage of “pen strokes”, mesh sculpting gives the user the ability to interactively and directly manipulate the 3D mesh models with instant alteration feedback. Users essentially incrementally “massage” the mesh model/regions into the desired shape. This form of incremental 3D mesh manipulation with instant feedback avoids ambiguity seen in sketch-based methods where sets of contour can be non-unique in terms of the shapes which they attempt to represent [96]. Every motion/stroke in mesh sculpting leads to instant shape alteration. Therefore, mesh sculpting avoids the need to process ambiguous sets of contours into shape alteration operations. Without ambiguity presented by sketch contours, mesh sculpting methods provide users with more predictable outcomes [97]. Due to this advantage and its shape alteration capability, mesh sculpting is the prime candidate for the unconstrained feature-free model modification module of the proposed hybrid modeling scheme. Modeling flexibility is increased with mesh sculpting. Further elaboration of the specific mesh sculpting software tool chosen for this work is outlined in Chapter 5.

2.4 Related feature-based model editing work

The last module of the hybrid modeling scheme is Feature-Based Model Editing. The goal of this module is to realize the direct modeling capability seen in the current parametric CAD modeler for the proposed hybrid CAD modeler. This means the direct modeling capability needs to be realized with proper association to the triangle mesh representation. The feature-based editing method not only needs to be able to solve for feature parameters necessary to reflect the user imposed feature editing operations, but also needs to be able to propagate these edits to the

triangle mesh representation appropriately. Such work currently does not exist; however, some related work has been published and is reviewed below.

As a reminder, the methods outlined in the previous section treat mesh models as homogeneous entities and deform them similar to rubberized bodies [85-90]. These methods are not feature-based. For feature-based modeling, the focus is quite different. Deformation of the model needs to follow specific rules which are implicitly set by the prismatic features present in the mesh model and by the feature dependencies of these prismatic features. During editing, feature types need to be preserved and constraints need to be kept, unless the edits require changes in the constraints [98]. The previously cited work does not take these into account and thus deforms the mesh with inadequate regards to the features. Therefore, the results from these editing methods do not accurately facilitate feature-based modeling.

Work that focuses on satisfying the above requirements have been attempted for mesh. However, the approaches either have very limited editing capability (small movements only) [99-101] or can only preserve the features and their constraints visually [90,102,103]. The features are a close but not an accurate representation of the intended design. Work such as [104] attempts to preserve features and constraints more accurately; however, in order to facilitate easy interaction and light-weight analysis, the preservation is not rigorously kept. Furthermore, the editing capability of the method is limited since the author's notion of editing involves the preservation of the essence of shape or rather, the preservation of the structure of the model. The arrangement (structure) of the features in the model cannot be re-established and the dependencies amongst the features cannot be changed.

It is evident that structure-altering editing is required in order to fully realize prismatic triangle mesh model feature-based editing. Typically, separate techniques using Boolean operations are utilized for editing the structure of prismatic triangle mesh models [31,35]. However, these methods involve the complete separation of the critical features (feature-of-interest) from the rest of the mesh model and the eventual reattachment of the critical features elsewhere [105]. This cuts the mesh at the new feature location and creates a hole that needs to be filled at the original feature location. The repeated cutting and filling operations are non-robust and detrimental to the quality of the triangle mesh. Cutting and patching of mesh should only be done when necessary and not used as an editing technique. Therefore, given the editing limitation of existing methods, users often still rely on remodeling as their primary method of design betterment when working with prismatic engineering mesh models.

With the inadequacy of the existing attempts to feature-based model editing, a method to mimic direct modeling seen in the parametric CAD regime is currently unavailable for the hybrid modeling scheme. Therefore, this thesis work sets out to develop a robust feature-based model editing method that gives rise to direct modeling capability for the hybrid modeler.

Chapter 3: Feature Segmentation

The first step in realizing the hybrid modeling system is to acquire the capability to bring the notion of features into a featureless flexible model representation. A reliable method to deduce and extract features directly from the mesh model geometries is necessary since without the ability to consistently identify features on mesh models, the precision and accuracy seen in parametric feature-based CAD modeling cannot be readily reflected and combined with the flexibility of mesh modeling techniques. The proposed flexible hybrid modeler is not realizable. Thus, a method to robustly, autonomously and exhaustively segment all prismatic features (planes, cylinder, sphere, cone and torus) on any given prismatic engineering watertight manifold triangle-mesh model into separate feature patches is required for the realization of the triangle mesh based flexible hybrid modeler.

Furthermore, with the feature segmentation module, shape deduction capability that aids to detect and to identify the user's intent during unconstrained feature-free modeling is achieved. The manipulated regions are segmented into feature patches that approximates the prismatic features which the deformed triangle-mesh area likely represents. This facilitates the utilization of the precision and accuracy of parametric CAD modeling while giving the user increased freedom and flexibility for geometric modeling. The user can deform the mesh model at any region into any rough prismatic feature shape with the segmentation algorithm automatically and transparently segments the deformed area into prismatic feature patches. Geometric manipulation restriction imposed by individualized feature entities is avoided and the precision and accuracy of parametric CAD modeling system is retained.

3.1 Methodology

The goal of this work is to develop a segmentation method that can automatically and reliably segment input triangle meshes of mechanical engineering parts into individual patches of design features (planes, cylinders, cones, and spheres) and detail features (chamfers, fillets, and constant-radius blends) to facilitate the imposition of feature information on featureless mesh models, and overall, to realize the proposed hybrid modeler. It is assumed that the mechanical part of interest in this work is made up of multiple prismatic features (listed above) bounded by convex or concave edges. As stated before and repeated here, the method has to be able to handle both scanning-derived meshes and unconstrained user manipulated meshes since these are the most error packed triangle mesh types experience by the hybrid modeler (see Figure 2.1 for mesh types and errors). Scanning-derived meshes possess chamfered-edge effects, noise, and very often real physical defects capture by the laser scanners. While, unconstrained user manipulated meshes contains chamfered-edge effects and large surface undulations (low frequency defects). Lastly and most importantly, any user involvement during the segmentation process is to be avoided so as to ensure full automation, robustness and consistency. Since not only is the segmentation method used to impose the notion of feature on the featureless triangle mesh inputs, but for the proposed hybrid modeling method, it is also used to deduce the features which the user is modeling during unconstrained feature manipulation. In this case, user transparency becomes very important. The segmentation algorithm should be completely transparent to the user. As the user manipulates the geometry, the segmentation algorithm has to automatically segment the manipulated area into features, which is later idealized. The modeling method will not work if the user is left heavily editing the segmentation results at every modeling step.

3.1.1 Fundamental principle

The mesh segmentation method proposed in this work is based on analyzing the histogram constructed from the discrete mean curvature at each mesh vertex [106]. Mean curvature (average of the principal curvatures: $\kappa_m = (1/R_1 + 1/R_2) / 2 \text{ mm}^{-1}$) is chosen as the segmentation indicator because it is distinctive at each mesh vertex and more importantly, it is able to realize the inherent geometric difference between adjacent prismatic features and edges of common mechanical parts.

Even though both mean curvature and Gaussian curvature (product of the principal curvatures) are needed to uniquely identify the shape of a prismatic surface [107], Gaussian curvature is not as useful to signal the difference in shape that exists between a particular patch and its neighboring patches. For example, the surface shown in Figure 3.1 consists of a cylindrical surface patch and a plane surface patch. A difference exists in the mean curvature values of the two features. And this difference in mean curvature can be exploited to segment the surface into distinct feature patches. Gaussian curvature, however, is zero for both the cylindrical and the plane surface patches, rendering Gaussian curvature an ineffective segmentation indicator. By using a single capable measure, mean curvature, the complexity of the segmentation approach is minimized.

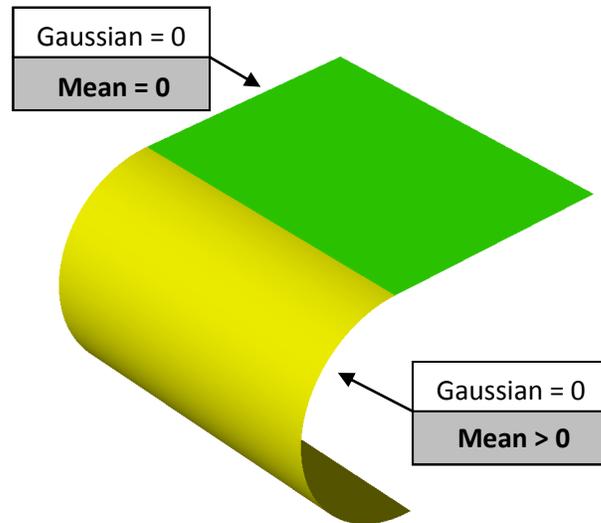


Figure 3.1 Mean vs. Gaussian curvature.

In principle, the proposed approach is similar to the interactive curvature histogram based point-cloud segmentation method introduced by Demarsin *et al.* [108]. This method is capable of segmenting point clouds with simple and smooth underlying features. It attempts to automatically determine a single threshold value through analyzing the point cloud's curvature histogram and segment the point cloud based on the determined threshold. The user then edits this threshold value in order to improve the segmentation results. If more than one threshold value is required, which is mostly the case for typical engineering objects, the user has to add more threshold values manually. The method also requires the scan data to have no or very low noise. This is necessary in order to ensure that the resulting curvature histogram is well-behaved and thus, usable for segmentation. However, this expectation is not applicable to real scan data, which often do not yield well-behaved curvature histograms.

The method by Demarsin *et al.* did not aim at fully automatic segmentation. Nonetheless, the method did demonstrate that the curvature histogram could be analyzed to extract appropriate threshold values necessary for the segmentation of mechanical parts. The proposed method in the present work builds on the potential of the mean curvature histogram and extracts all the thresholds necessary to completely segment scanning-derived meshes and unconstrained user manipulated meshes for CAD applications. Its fundamental principle is to identify the valleys in the mean curvature histogram and subsequently use the corresponding curvature values to segment the mesh. The curvature range covered by each pair of neighboring valleys adjacent to a peak in the mean curvature histogram is indicative of a possible feature. As shown in Figure 3.2, if there are four cylinder features of different radii in a triangle mesh model, the mean curvature histogram will ideally be made up of four distinct spikes at each of the four mean curvature values corresponding to the four cylinders. Thus, to segment the four cylinders, the proposed algorithm simply groups triangles according to the particular curvature values of the four cylinders. The cylinders are then easily extracted from the valleys in the histogram.

If the four cylinder features are of the same radius, then only one single peak exists in the mean curvature histogram. One single peak only entails a single feature even though four features exist. Inappropriate segmentation results are thus produced. In the present work, however, all distinct patches in the mesh are treated as features. Features include not only planes, cylinders, cones and spheres but also sharp edges, blends, chamfers, and convex/concave corners. In other words, the mean curvature histogram of a four-cylinder object with different cylinder radii will yield more than four peaks. As shown in Figure 3.2, a fifth peak exists on the left. This fifth peak represents the intersected edges at the joints of the cylinders. The proposed segmentation

method uses this information and segments the object not only through grouping regions of similar mean curvature values but also by dividing the mesh into patches via edges. And this is achieved by applying only one geometric measure, the mean curvature.

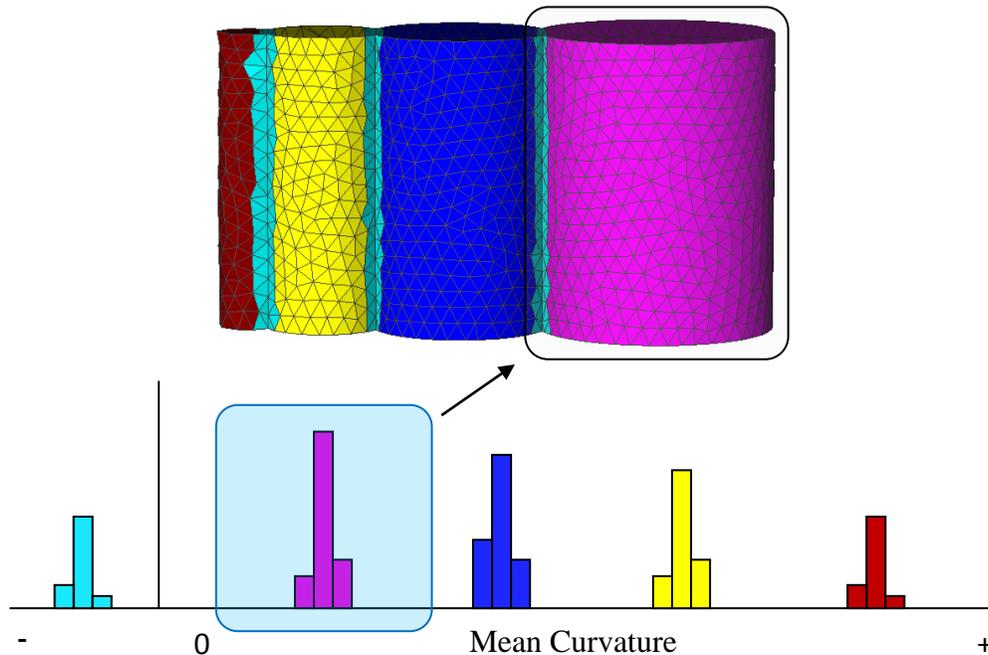


Figure 3.2 Object identification (top) through mean curvature histogram (bottom).

3.1.2 Imperfection effect reduction

In a mean curvature histogram, positive mean curvatures correspond to “mainly” convex features and negative mean curvatures correspond to “mainly” concave features. The word “mainly” is employed here since mean curvature is the average of principal curvatures. A point on an object surface can have positive curvature in one principal direction and negative curvature in the other principal direction. As the average of the two principal curvatures is the mean curvature value,

the larger principal curvature dominates the resulting mean curvature value. A surface of positive mean curvatures can in fact be “mildly” concave along the negative principal curvature direction.

As noted previously, the mean curvature histogram of scanning derived meshes and user manipulated meshes are generally not well-behaved. Even for an ideal mesh created from a CAD model (edges being sharp and containing no noise), the curvature histogram does not exhibit perfect spikes. Figure 3.3 shows an ideal mesh and its corresponding mean curvature histogram. The peaks are not perfect spikes but slightly spread out. This spread is still quite acceptable since each peak and its adjacent valleys are identifiable. In this case, the method by Demarsin *et al.* would work well.

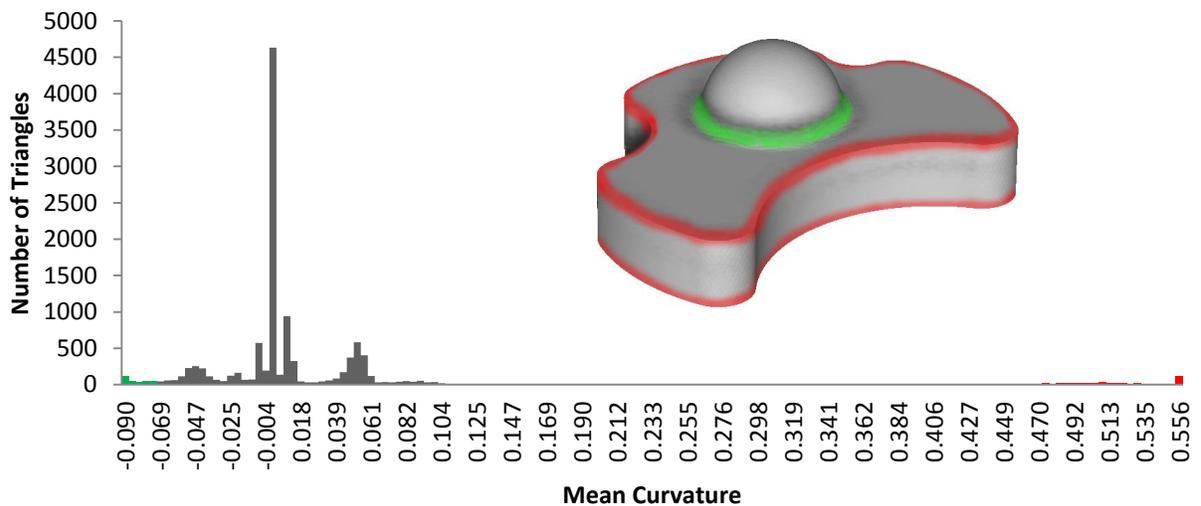


Figure 3.3 Typical curvature histogram for an ideal mesh from a CAD model.

Unfortunately, the peaks would become increasingly outstretched when chamfered edges are introduced into the ideal mesh, making it more difficult to identify the valleys in the histogram. And with the addition of scanning noise and/or surface undulations, the peaks widen to a point such that the exact valley locations are no longer visible. Figure 3.4 shows the mean curvature

histogram of the ideal mesh shown in Figure 3.3 with superimposed chamfered edges and scanning noise for illustration. The histogram conveys little or no information at all regarding the features present in the mesh model being segmented. Fortunately, the needed information is recoverable. And to perform the recovery, a method to condition the histogram such that peaks and valleys are guaranteed to be visible is developed.

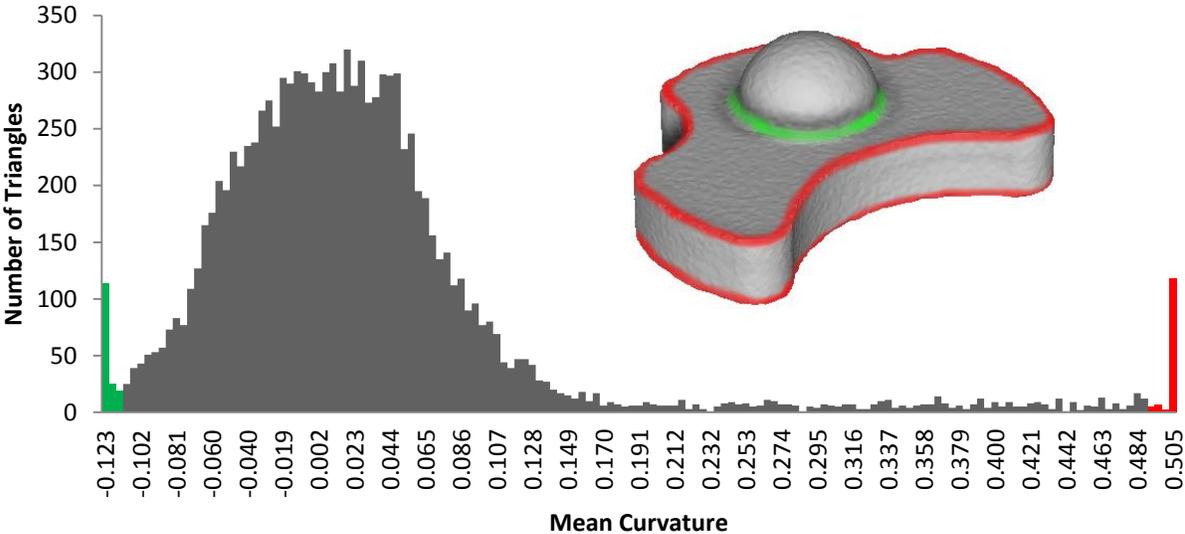


Figure 3.4 Mean curvature histogram of a mesh containing chamfered edges and noise.

To condition the mean curvature histogram of scanning-derived meshes containing chamfered edges and noise and also of user manipulated meshes containing chamfered edges and surface undulations (in order to reliably recover the peaks and valleys), a two-part algorithm is proposed. Figure 3.5 shows a flow chart of the algorithm. The first part is to extract the sharpest features which, in the mean curvature histogram, are indicated by the extremity peaks (the -0.123 and 0.505 bins in Figure 3.5). These peaks can be identified quite reliably for scanning-derived meshes and user manipulated meshes since the practical scanning noise levels are relatively low [109] and the surface undulations are quite large. The scanning noise typically translates into

0.01 mm^{-1} of mean curvature noise, which is equivalent to 100 mm in curvature radius. Since the surface undulations are of lower frequency defects than the scanning-noise, the surface undulations must be characterized by curvature radius values of larger (typically significantly larger) than 100 mm. For an edge to be considered as a sharp edge, its radius has to be significantly less than 100 mm. This means that the mean curvature of a sharp edge will be much larger than 0.01 mm^{-1} or if negative curvature, much smaller than -0.01 mm^{-1} . Because of this, the extremity peaks that represent edges should always be visible and identifiable on a mean curvature histogram and situated distinctly away from the rest of the peaks.

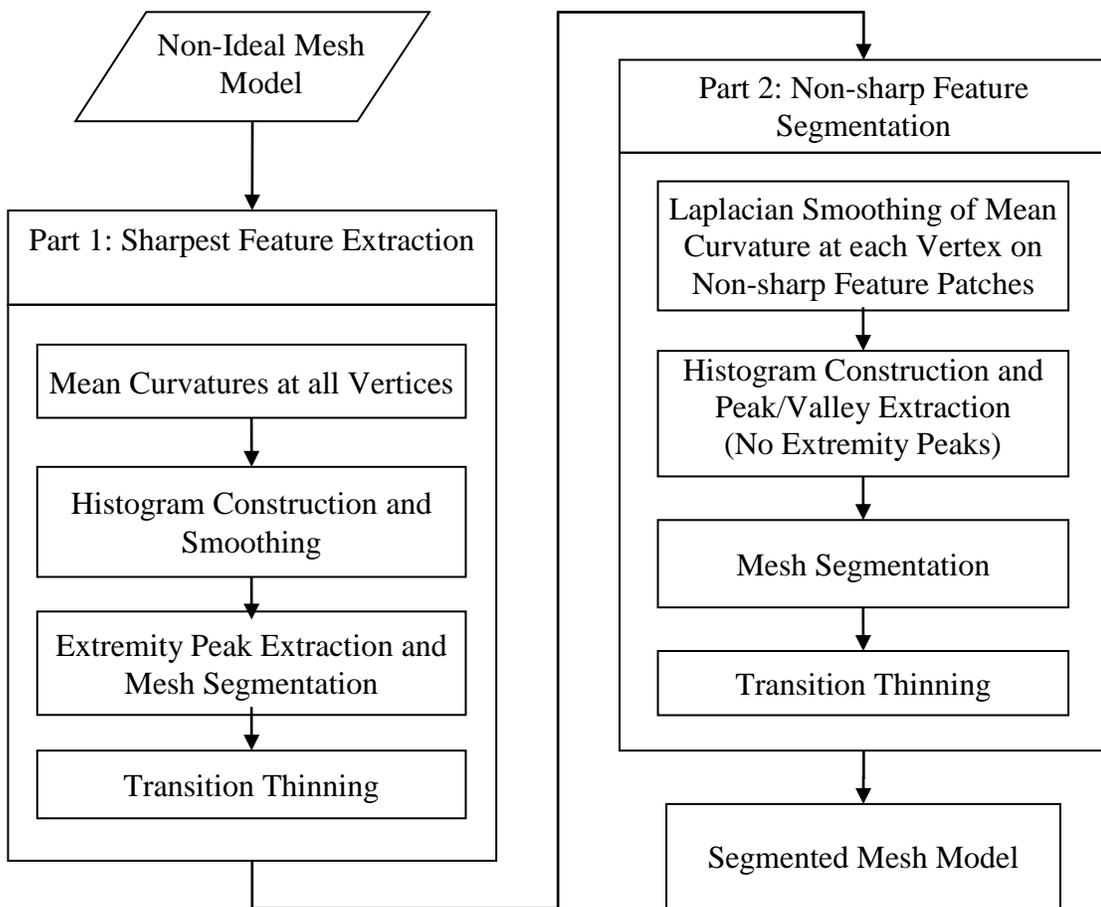


Figure 3.5 Proposed mean curvature histogram based segmentation method.

Once the sharpest features are extracted, vertex-based Laplacian smoothing [110] is applied aggressively to the remainder of the mesh as the second part of the proposed segmentation algorithm in order to robustly deal with surface noise and undulations. The portion of the mesh that is classified as sharp is not smoothed by the Laplacian smoothing operator. The method keeps the sharp and smooth regions separate in order not to distort the mean curvature values when recovering the “ideal” histogram. With the Laplacian smoothing, the mean curvature histogram is reconstructed. As shown in Figure 3.6, the peaks that are indicative of individual features are appropriately recovered. The valley curvature values are identified and usable for segmentation. The two-part algorithm recovers the ideal mean curvature histogram which in turn enables reliable and automatic segmentation of scanning-derived and unconstrained user manipulated meshes. Users do not have to interactively tune any parameters to deal with the presence of chamfered edges, noise and surface undulations.

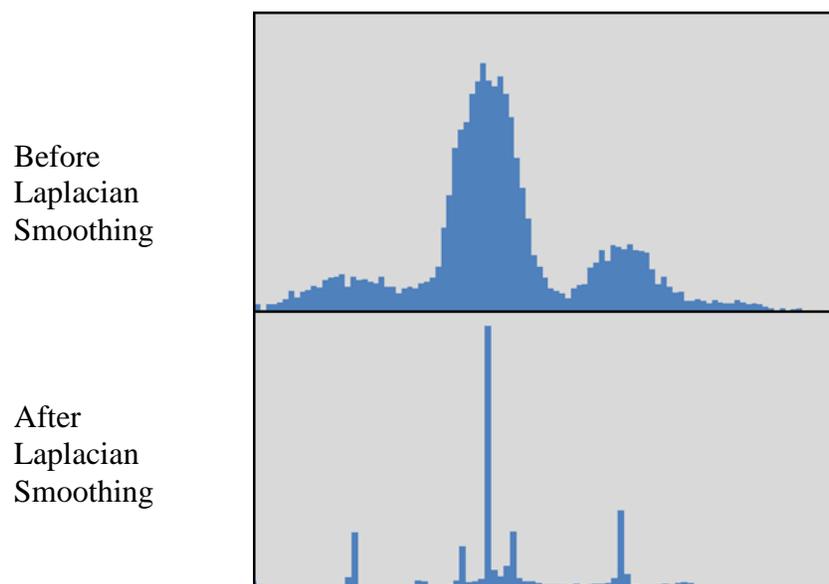


Figure 3.6 Effect of Laplacian smoothing on the mean curvature histogram (sharpest features removed).

3.2 Detailed procedure

The following subsections present the two-part algorithm from an implementation point of view. The first part of the algorithm is described in detail. Since the fundamental functions of both parts are quite similar, only the principal differences are given for the description of the second part. The two parts both start with histogram construction and conditioning and terminate with segmentation and post-processing.

3.2.1 Sharpest feature extraction

The first part of the algorithm only segments or extracts the sharpest features on the mesh. Details are given below. The required mean curvature value at each vertex on the mesh is calculated using the method of Meyer *et al.* [106].

3.2.1.1 Mean curvature histogram construction

To construct the mean curvature histogram, only one parameter needs to be set, which is the bin size. Any choice of the bin size will allow the construction of a histogram; however, different levels of detail are captured depending on the employed bin size. As a general guideline, the number of histogram bins should be sufficiently small such that high-frequency noise is avoided, but is not overly small such that the underlying shape of the histogram is lost. The bin size of a histogram in effect acts very closely as a low-pass filter, which allows low-frequency contents to pass through while attenuates the high-frequency contents. To determine the suitable bin size for this work, a straightforward trial-and-error procedure was used. It was found that with the subsequent proposed smoothing operation, the bin size selection is in fact not critical. A bin size

of 100 to 500 that spans the associated minimum and maximum mean curvature values can be used for most objects (of varying sizes and mesh densities) with almost identical results. A bin size of 150 is set for this work. This is done only to reduce the computational load.

3.2.1.2 Mean curvature histogram smoothing

With the histogram constructed, histogram smoothing is imposed in order to facilitate robust extraction of histogram peak and valleys. Noise always exists in the constructed histogram even with good pre-processing before construction. Therefore, a smoothing algorithm to remove the noise in the constructed histogram is proposed in this work. The algorithm evaluates the average fluctuation in the frequency of the adjacent histogram bins first. The average fluctuation is simply the average of the absolute difference in frequency between the i th histogram bin and the i th + 1 histogram bin for a total of n histogram bins:

$$Average_Fluctuation = \frac{\sum_{i=0}^{n-1} abs[hist_freq(i) - hist_freq(i + 1)]}{n} \quad (1)$$

Then, for a given bin in the curvature histogram, it is smoothed if it is a peak or a valley and is characterized by a fluctuation with respect to both of its adjacent bins which is less than the average fluctuation. This means that any monotonically increasing or decreasing bins of the histogram are not smoothed and bins characterized by higher than average fluctuation are not smoothed. The necessity of not smoothing the monotonic regions is to ensure that the location of the peaks and valleys are not distorted. The proposed smoothing algorithm works very well and allows reliable extraction of the peaks and valleys in the histogram. More importantly, the algorithm appears to be asymptotically stable. Empirically, once the algorithm is used to smooth

a given histogram for about 100 times, further smoothing does not modify the smoothed histogram further. This implies that, as long as the histogram is smoothed aggressively (above 100 times), further smoothing no longer affects the outcome of the two-part segmentation method.

Note that, for the development of this smoothing algorithm, no assumption is made regarding the characteristics of the histogram. As long as the bin size is sufficiently large to capture the shape of the histogram, the proposed smoothing algorithm allows for the appropriate extraction of the histogram peaks and valleys.

3.2.1.3 Extraction of peaks and valleys

Once the histogram is smoothed, extraction of valleys is just a matter of finding bins which are adjacent to larger bins on both sides. This works well in regions with high-amplitude peaks and valleys; however, in the flatter regions with low-amplitude bins, an overly large number of peaks and valleys can be identified. This is because low-amplitude oscillation is always present no matter how much smoothing is applied. To alleviate this issue, an initial set of peaks and valleys are first automatically identified. A bin is considered a peak when its adjacent bins are both lower in magnitude. For valleys, both adjacent bins are larger in magnitude. The valleys are then compared against their adjacent peaks and the peaks compared against their adjacent valleys. Similar to Eq. (1), an average fluctuation between m pairs of adjacent peaks and valleys is evaluated:

$$Av_Peak_to_Valley_Fluctuation = \frac{\sum_{i=0}^{m-1} abs[peak(i) - valley(i)]}{m} \quad (2)$$

Then, for those valleys whose fluctuations with adjacent peaks are below the average peak-to-valley fluctuation, they are considered part of the low-amplitude oscillations and thus not considered. This helps with the rejection of user manipulation caused surface undulations and scanned real-life part defects. The remaining valleys are then employed for segmentation. The average fluctuation is effectively used here as a reference to remove excessive peaks and valleys because the differences between adjacent bins in the histogram are usually small in most regions except at feature spikes. Thus, peaks and valleys that are representative of features are retained and only small oscillating peaks and valleys that are not representative of features are removed. Again, ignoring these small oscillations helps the rejection of surface undulations and real-life part defects as features.

3.2.1.4 Segmentation

The segmentation step is straightforward. For each peak, there are two adjacent valleys. The mean curvatures corresponding to these two adjacent valleys are used as the bounding values for the particular peak. Accordingly, those triangles on the mesh that are within these bounding curvature values can be identified. In order for a triangle to be considered as within the specified bound, the curvatures at all three vertices have to be within the bounding range. If any of the three curvature values does not fall within the bound, the triangle is regarded as a transition triangle. These transition triangles are in effect triangles between features, or rather, represent the border of elementary feature patches. The segmentation approach is shown in Fig. 3.7. It should be noted here again that for the first part of the algorithm, the mean curvature histogram is only divided into three sections: the two end extremity sections that represent the sharpest features and the middle section that consists of the non-sharp features. As mentioned previously, only the

extraction of the sharpest features is reliable at this part of the algorithm. It is possible that the mean curvature histogram only contains positive (or negative) mean curvature values. In this case, the negative (or positive) extremity peak is simply ignored. Only the positive (or negative) extremity peak is extracted and the mean curvature histogram is just divided into two sections.

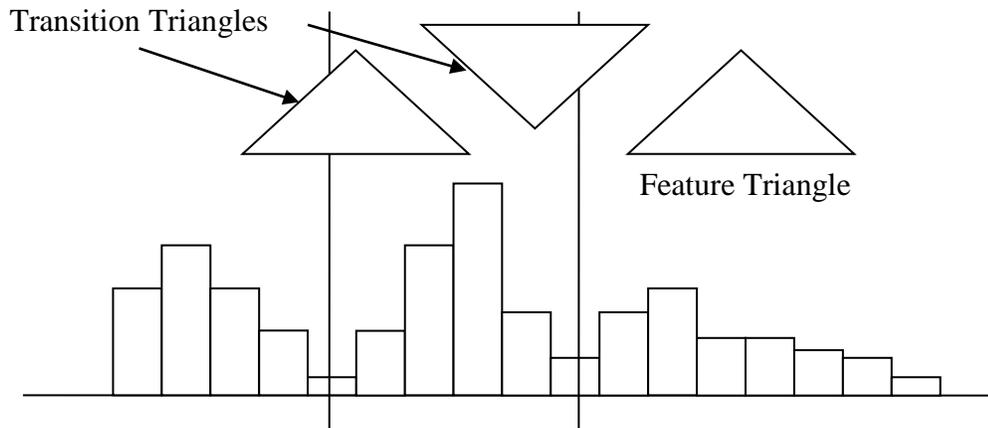


Figure 3.7 Segmentation via curvature histogram.

3.2.1.5 Transition thinning post-processing

After segmentation, one important post-processing task is necessary. It is called transition thinning in this work. It is done to ensure that all sections of connected transition triangles are no more than one-triangle wide. This is because the proposed segmentation method is based on the consistency in the mean curvature values at the three vertices of a triangle. As a result, the transition/border section surrounding any elementary feature patch should be only one-triangle wide. Conceptually, since transitions bound features, features should bound transitions and transitions should not be bounded by further transitions. Nonetheless, because the presence of mesh-imperfections, a triangle adjacent to an ideal transition section can also be considered a

transition triangle. This causes practical transition sections to widen. Thus, there are at times, transition sections that are more than one-triangle wide.

To ensure that the transitions are reliably extracted, the identified transition sections are examined and processed if necessary. Figure 3.8 illustrates the steps involved in the proposed transition thinning task. First, the triangles in the transition section that are one-triangle wide are identified. This is simply done by identifying transition triangles where all the three vertices are shared with adjacent feature patch vertices. These triangles do not require thinning. However, the interface edges with more-than-one-triangle wide section patch are marked. These interface edges can be thought of as the entry or exit edges that connect adjacent one-triangle wide patches with the more-than-one-triangle wide patch. To form a one-triangle strip successively from the entry to the exit edge, triangle edges that are characterized by the higher average mean curvature (of their two end vertices) are employed and marked as the growing edge for the next triangle. Using this method, complete transition sections that surround elementary feature patches are easily identified and shown as yellow triangles in Figure 3.8.

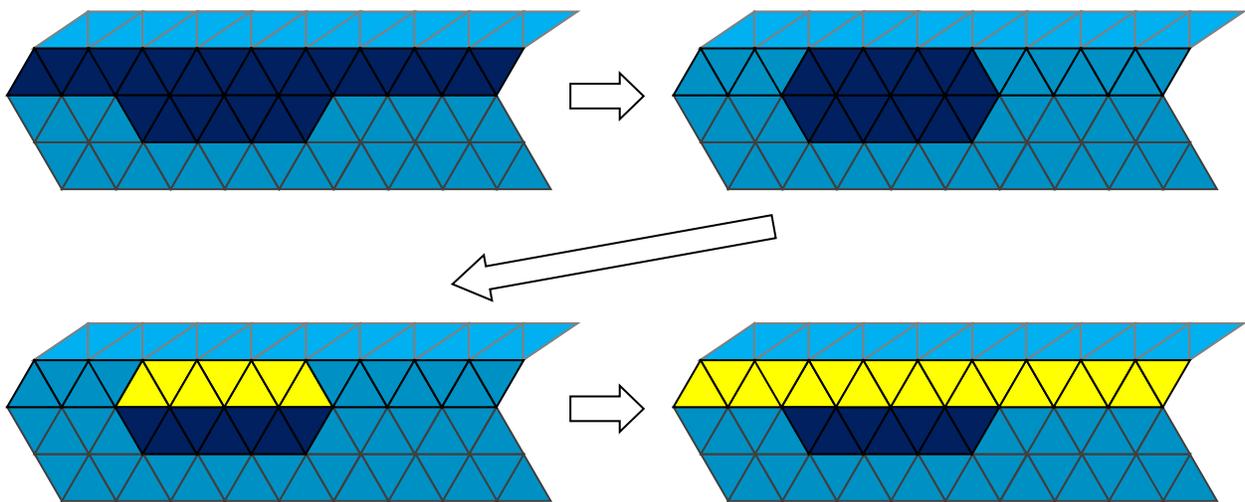


Figure 3.8 Thinning of a transition section that is more than one-triangle wide.

3.2.2 Non-sharp feature segmentation

After the sharpest features are extracted, vertex-based Laplacian smoothing is applied to the feature patches. It should be noted that the feature patches usually need to be further segmented since at this point only the sharpest features are segmented from the overall mesh model. However, as stated previously, the histogram valleys (other than the extremity ones) cannot be reliably determined due to the presence of noise and surface undulations. The presence of these surface imperfections spreads the peaks out and distorts the shape of the histogram. To mitigate this effect, Laplacian smoothing is applied to feature patches to restore the histogram shape.

Similar to the first part of the algorithm, the smoothed mean curvature histogram is constructed and processed. However, unlike the first part of the algorithm, all the extracted valleys except the extremity ones are used for segmentation. The feature patches that possess non-sharp features and need further segmentation are segmented into multiple distinct patches. The transition thinning post-processing is then applied as the last step of the two-part algorithm. At this point, the input scanning-derived mesh is segmented into individual design and detail features with transitions clearly established between these features in the form of one-triangle strips. And the goal of automatically segmenting scanning-derived and unconstrained user manipulated meshes of mechanical parts to enable feature imposition on featureless triangle model is achieved.

3.3 Case studies

Case studies have been performed to illustrate the effectiveness of the two-part algorithm in automatically and reliably segmenting mesh models of prismatic mechanical parts. Details are given in the subsections below.

3.3.1 Ideal meshes

A typical case of an ideal mesh, which is one of the healthiest mesh input types for the proposed flexible hybrid modeling system, is shown in Figure 3.9 and is directly obtained from a commercial CAD software package. This ideal mesh does not possess noise, chamfered-edges, surface undulations and real-life imperfections. Thus, the mean curvature at a vertex (approximated from the mesh geometry) should be very close to its ideal value on the CAD model. As a result, the mesh should yield a very well-behaved mean curvature histogram. Figure 3.10 shows the mean curvature of each elementary feature on the mesh model and the corresponding mean curvature histogram. The histogram is very well-behaved with peaks (color coded) that directly correspond to the mean curvature values of the underlying features. This validates that a mesh can be segmented reliably by analyzing its mean curvature histogram when it is well-behaved.

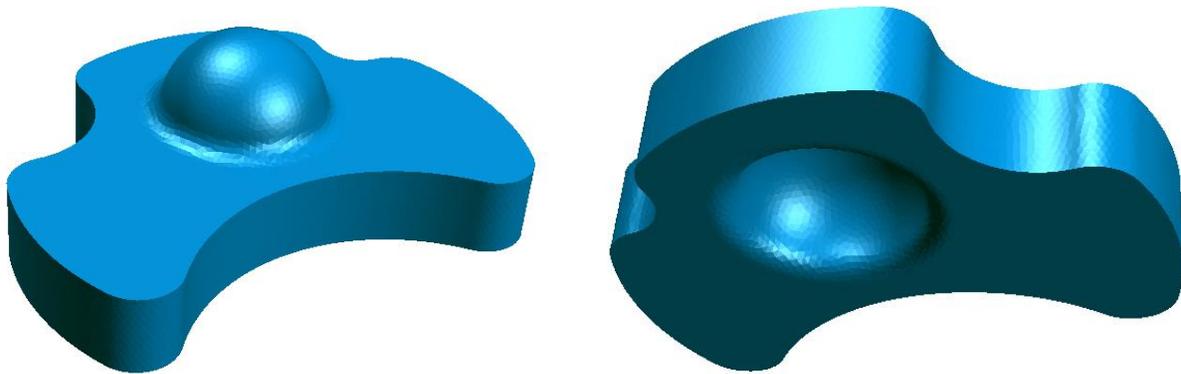


Figure 3.9 An ideal CAD mesh (shaded).

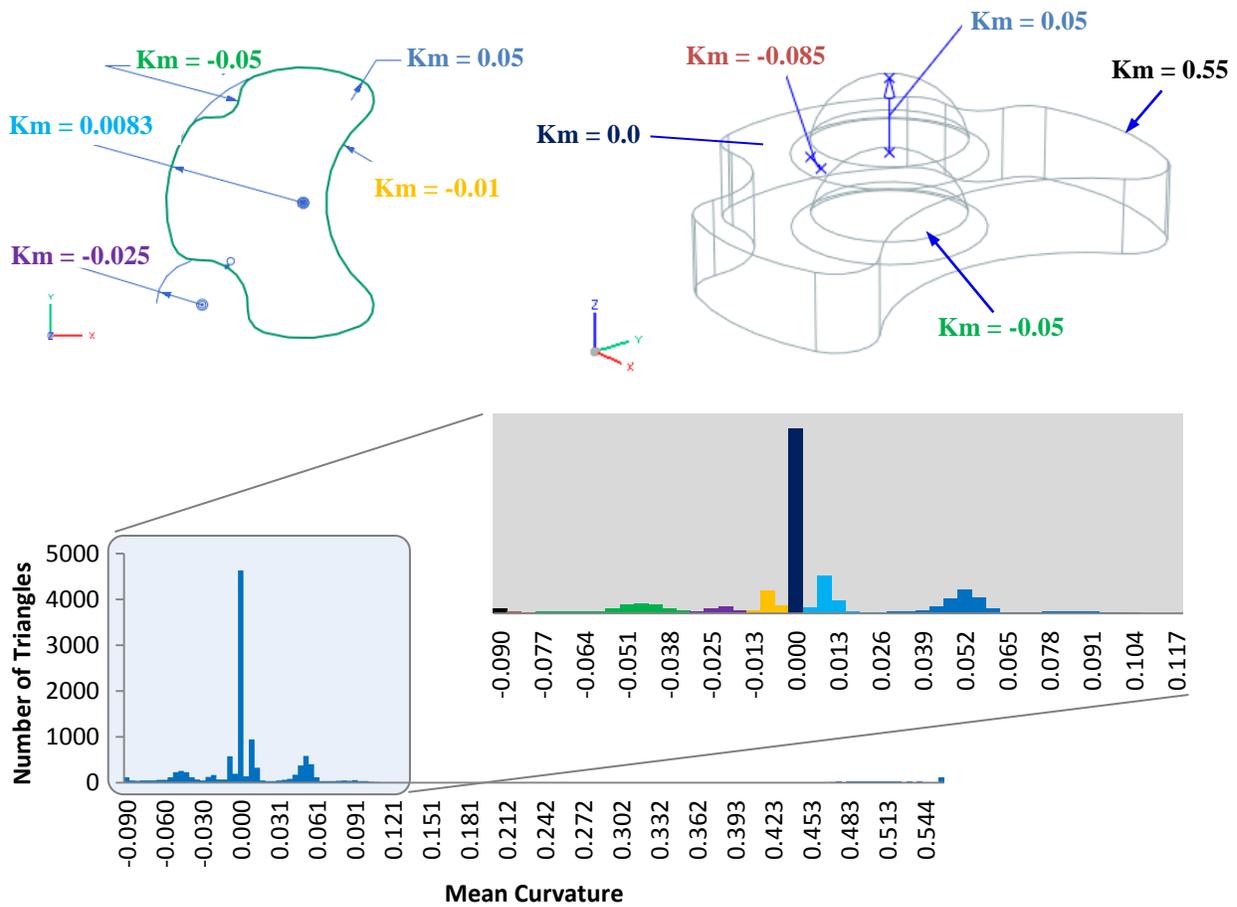


Figure 3.10 Elementary features on the ideal mesh model and the corresponding mean curvature histogram.

Per the proposed algorithm, the sharpest features are extracted first. In this ideal case, the algorithm does not need to be carried out in two parts since the peaks and valleys of the histogram are distinctly identifiable already. This is due to the mesh being ideal. However, it is not possible to yield such a well-behaved mean curvature histogram from an actual scanning-derived mesh or a user manipulated mesh. Since this algorithm aims to process these two worst case types of meshes, the two parts are still performed to better illustrate the proposed method.

For part one of the algorithm, the sharpest features are extracted as shown in Figure 3.11 along with the corresponding mean curvature histogram. As noted previously, the peaks that represent the sharpest features are distinctly far from those representing the elementary features. Extraction for the valleys of these sharp-feature peaks is thus trivial. As shown in the figure, both the positive-curvature sharp edges and the negative-curvature edge blends are segmented and extracted straightforwardly. It should be noted that the negative-curvature edge blends are segmented since they are the sharpest negative-curvature features on the mesh.

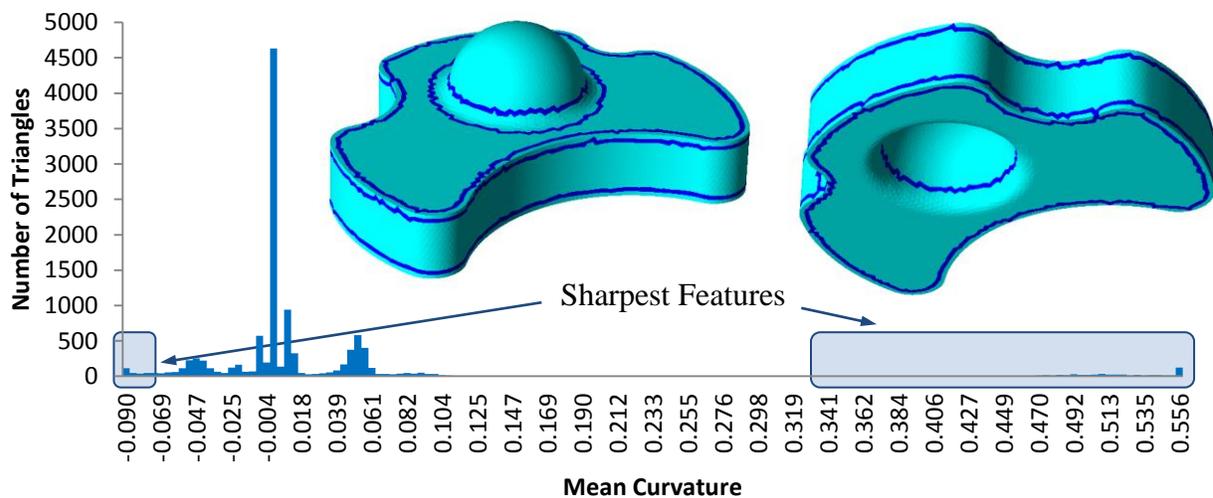


Figure 3.11 Sharpest feature extraction for the ideal mesh model.

With the sharpest features extracted, part two of the algorithm takes place and the non-sharp features are segmented to complete the segmentation process. The first step in part two of the algorithm is to apply per-vertex Laplacian smoothing. As mentioned previously, the smoothing blends the per-vertex mean curvature values of non-sharp feature patches in order to reduce the curvature value fluctuation caused by noise. Recall that such fluctuation spreads out and reduces the peaks in the histogram, making the valleys hard to identify. In this case study of using an

ideal mesh model, the mesh is already well-behaved. With Laplacian smoothing, Figure 3.12 shows that the peaks in the histogram became much sharper and much closer to the ideal mean curvature histogram where each feature is represented by a single impulse peak with no spread. The underlying features in the original CAD model can thus be easily segmented and correctly identified.

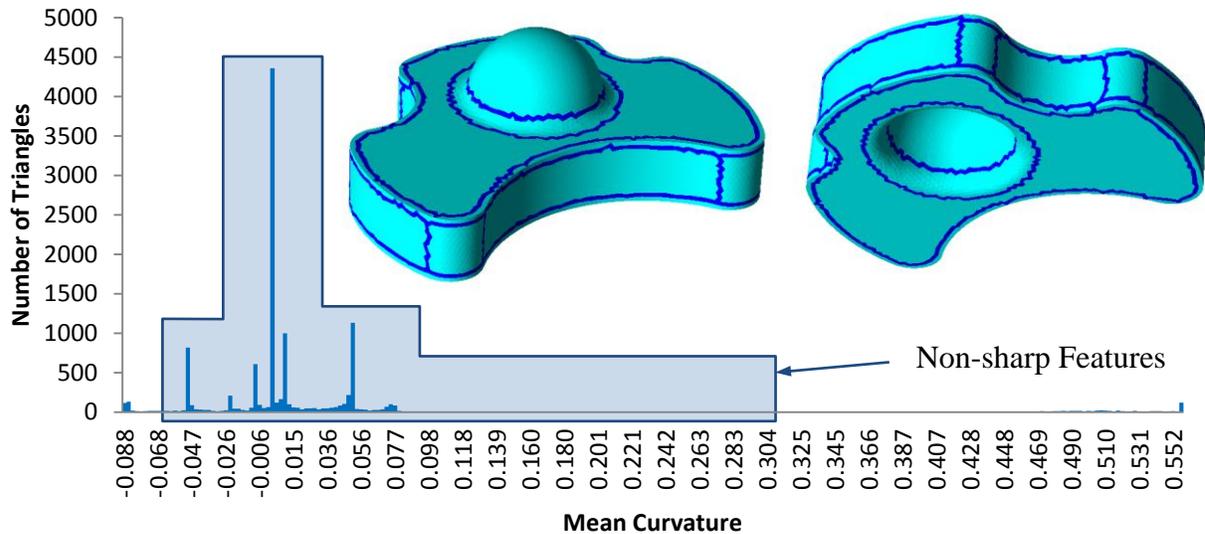


Figure 3.12 Non-sharp feature segmentation for the ideal mesh model.

To demonstrate the effectiveness of the proposed method against the existing methods, segmentation results of ideal CAD meshes are compared. The methods employed for the comparison are: Shape Diameter [48], Random Walks [53], Fitting Primitives [58], K-Means [46], Feature-based [60] and Local Geometric Filter [57]. As shown in Figure 3.13, methods (a)-(d) do not yield results in which all features of the CAD mesh model are appropriately extracted. Although the method of Fitting Primitives appears to be promising, the detail features (blends) are not segmented. This is an inherent issue for the fitting type of methods as discussed in

Section 2.1. Feature-based and Local Geometric Filter methods, which are essentially still based on fitting, are able to appropriately identify the individual design features as well as the detail features. However, a total of four parameters need to be tuned individually for the Feature-based method and a total of three parameters need to be tuned together for the Local Geometric Filter method (as implemented in the Geomagic software) in order to attain the appropriate fitting result. The proposed method (Figure 3.13g), on the other hand, is able to appropriately segment the CAD mesh model and generate the result automatically. No user input or parameter tuning is needed.

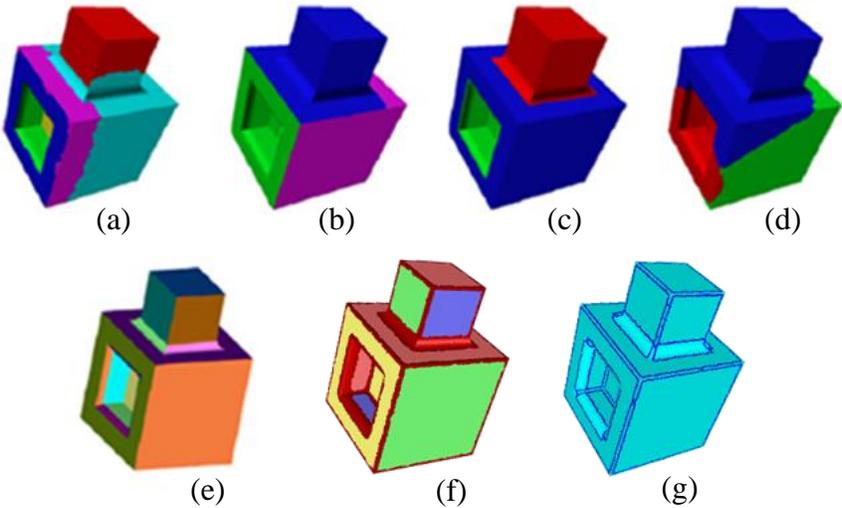


Figure 3.13 Segmentation results: (a) Shape Diameter; (b) Random Walks; (c) Fitting Primitives; (d) K-Means; (e) Feature-based; (f) Local Geometric Filter; and (g) proposed method. (a)-(e) replicated from Wang and Yu (2011).

3.3.2 Synthesized non-ideal meshes

Synthesized meshes that imitate the characteristics of scanning-derived meshes and user manipulated meshes are employed in this case study. By synthesizing the meshes, imperfections in the mesh model are controllable. For synthesized scanning-derived meshes, noise and chamfer

edges are added to the mesh resulting in a non-ideal mesh as shown in Figure 3.14. The mesh no longer possesses sharp edges and the local mean curvatures deviate from their original CAD values. As a result, the associated mean curvature histogram is now ill-behaved as shown in Figure 3.15. The peaks have spread out so significantly that the valleys are no longer visible. However, the two extremity peaks are still clearly visible, which means the sharpest features can still be reliably extracted. After finding the valleys that correspond to these extremity peaks, the sharpest features are extracted as illustrated in Figure 3.15. It should be noted that the segmented result of sharpest feature extraction (part one of the algorithm) is the same as that for the ideal CAD mesh shown in Figure 3.11. Thus, even with noise and chamfered edges, the sharpest features can still be reliably extracted using the mean curvature histogram.

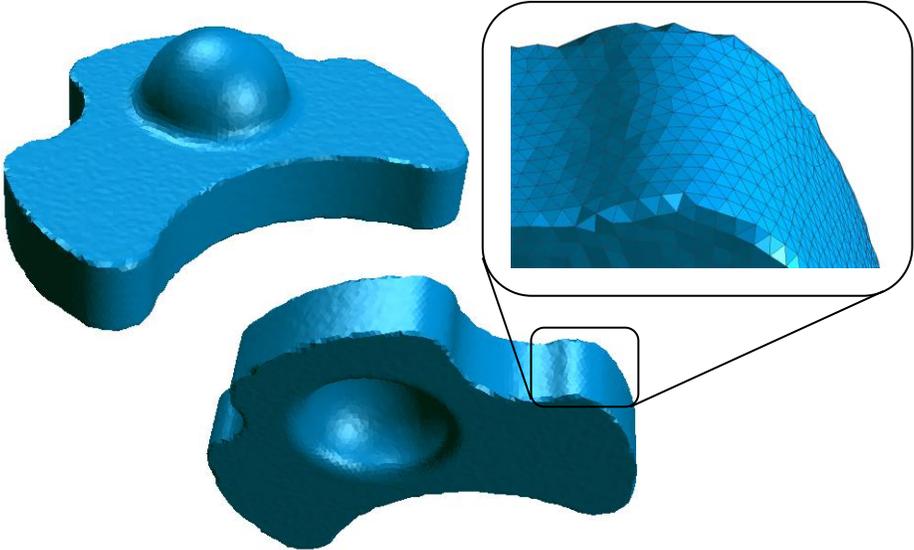


Figure 3.14 A synthesized scanning-derived mesh.

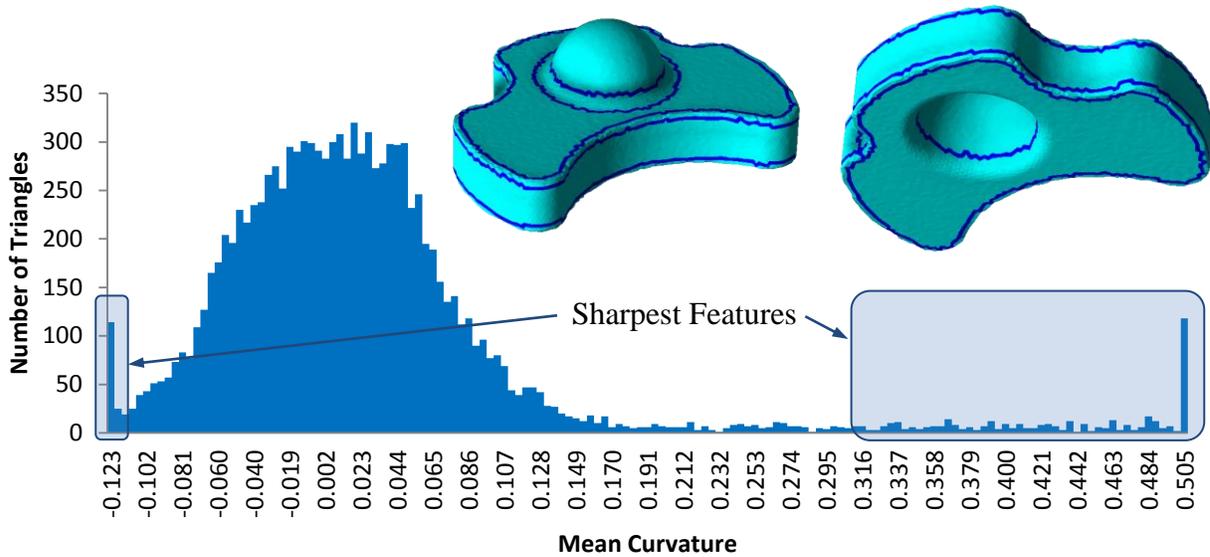


Figure 3.15 Sharpest feature extraction for the synthesized scanning-derived mesh.

To segment the non-sharp features, Laplacian smoothing is applied. For the ideal CAD mesh, not much difference is noted due to the mesh being ideal. For the current synthesized scanning-derived mesh, a large change in the mean curvature histogram is observed, before and after the Laplacian smoothing. The bell-shaped histogram with indistinguishable peaks in Figure 3.15 is now shown with distinguishable peaks and valleys in Figure 3.16. Very similar histograms are observed in Figures 3.12 and 3.16, which suggests that the smoothing, performed on the non-sharp feature patches, has conditioned the mean curvature values such that the effect of noise is largely alleviated. It is clearly seen that the segmentation results between the ideal mesh and the synthesized mesh are very similar with each segmented feature patch distinctly identified. In other words, the proposed two-part algorithm is effective in segmenting the mesh into its underlying features under conditions that are similar to those of a scanning-derived mesh (noise and chamfered edges). The design and detail features are all properly segmented.

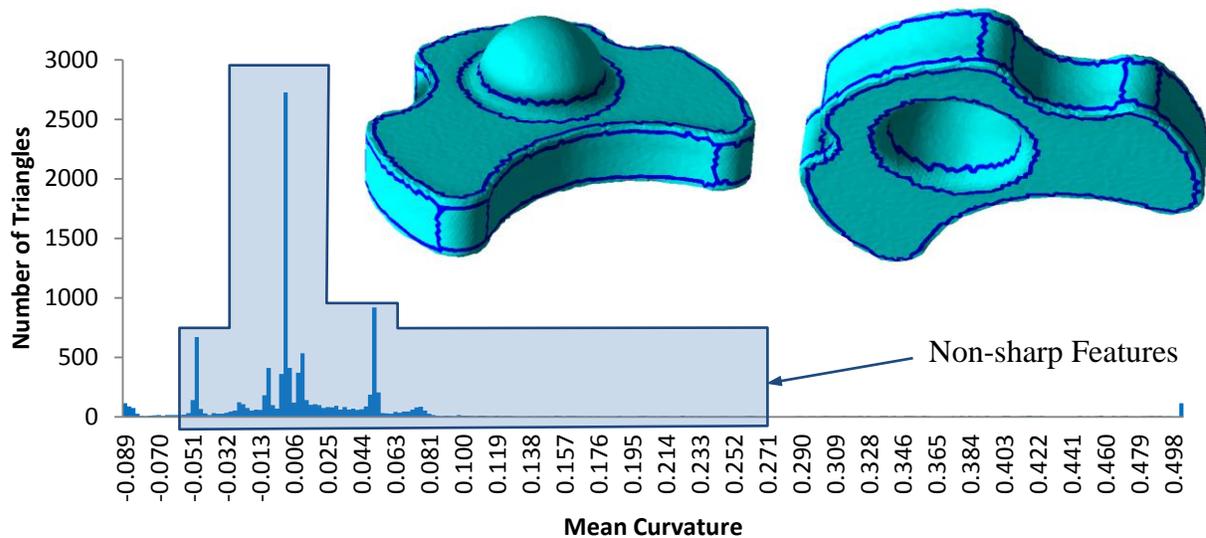


Figure 3.16 Non-sharp feature segmentation for the synthesized scanning-derived mesh.

For unconstrained feature-free user manipulated mesh, chamfered-edge effect and large surface undulations are added to the ideal mesh model in order to simulate the manipulated mesh (Fig. 3.17). Much like the synthesized scanning-derived mesh model, the associated per-vertex mean curvature histogram is ill-behaved. Some of the peaks are indistinguishable as shown in Fig 3.17. However, due to the lower frequency characteristic of the surface undulations, the shown histogram is better-behaved compared to the scanning-derived case's initial histogram. Nevertheless, the histogram still does not convey the information necessary to perform a proper segmentation. Some peaks that are representative of features are missing in the unprocessed histogram. As expected, the extremity peaks can be reliably used from the unprocessed per-vertex mean curvature histogram to perform segmentation. The extremity peaks that are representative of the sharp features are far enough removed from the center of the curvature histogram such that they reliably facilitate sharpest feature segmentation.

Utilizing the proposed two part segmentation algorithm, the sharpest features are extracted from the synthesized user manipulated mesh model (Fig. 3.17). Laplacian smoothing is performed post sharpest feature segmentation in order to recover the mid portion of the per-vertex mean curvature histogram. For the synthesized user manipulated mesh, as shown in Fig. 3.18, the mid portion peaks are well recovered; thus, all features on the mesh are reliably and automatically segmented through the proposed segmentation algorithm. The final segmentation result of the synthesized user manipulated mesh is shown in Fig. 3.18. It is seen that the segmentation result is equal in terms of feature types and features numbers to that of the segmented ideal mesh results (Fig. 3.12).

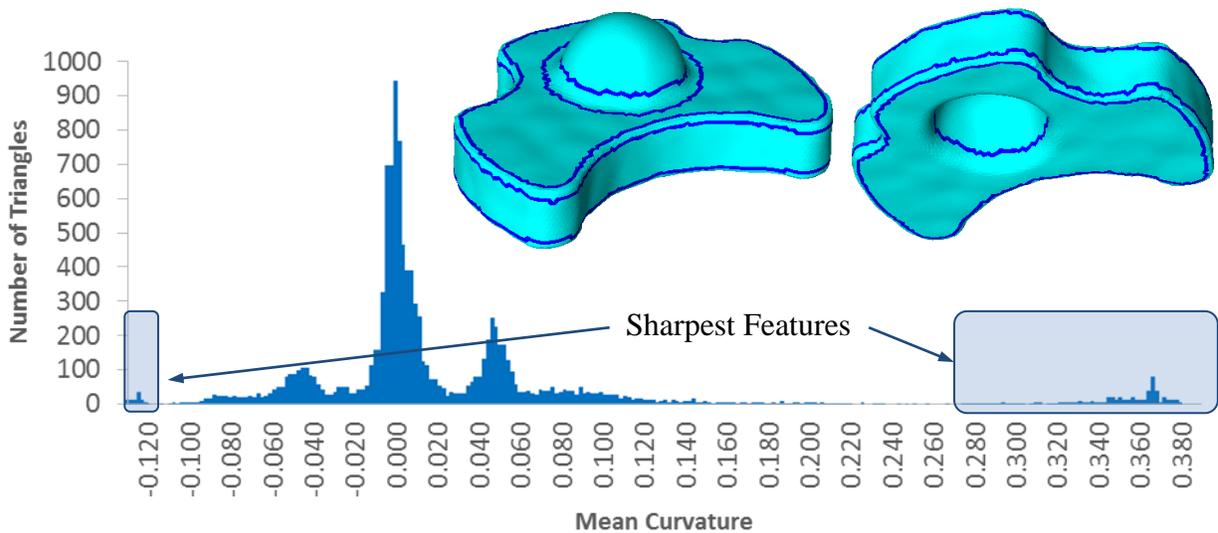


Figure 3.17 Sharpest feature extraction for the synthesized user-manipulated mesh.

Figure 3.19 and Figure 3.20 shows the segmentation results of four more synthesized scanning-derived meshes and four more unconstrained feature-free user manipulated meshes, respectively, to further illustrate the effectiveness of the proposed algorithm. It can be seen that the primary design features (planes, cylinders, spheres and cones) and the detail features (blends and fillets)

are all well extracted and segmented for both mesh type variations. The underlying features of the meshes are all present in the segmentation results.

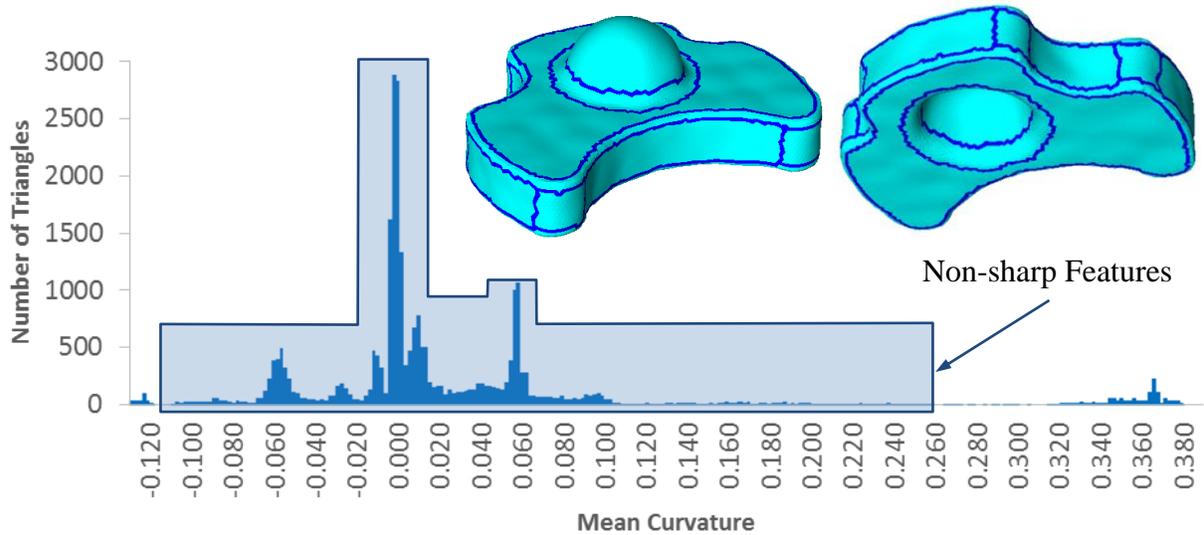


Figure 3.18 Non-sharp feature segmentation for the synthesized user-manipulated mesh.

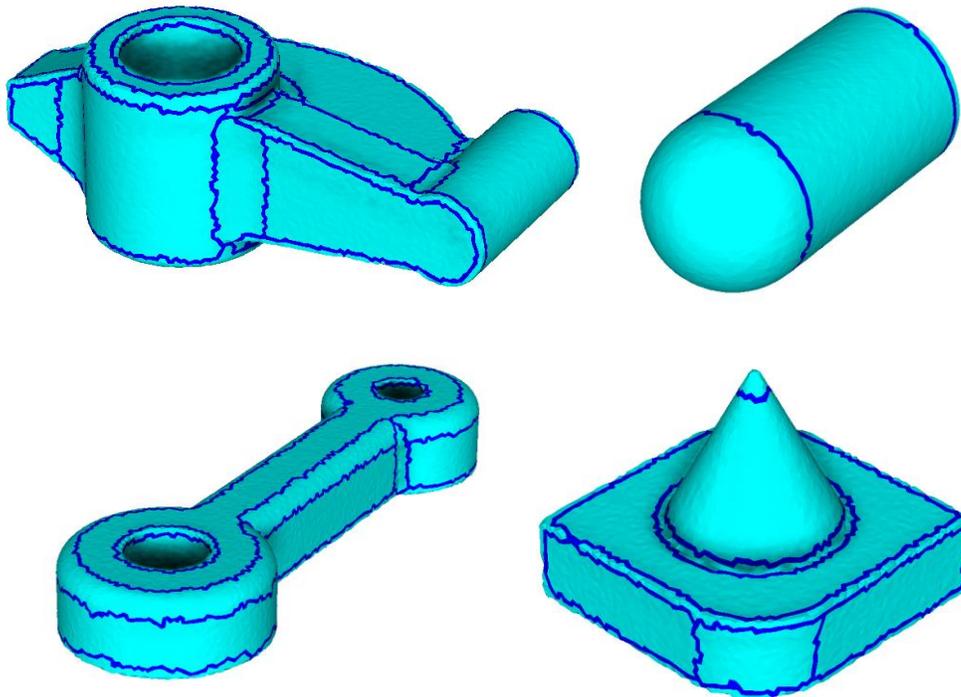


Figure 3.19 Segmentation results of additional synthesized scanning-derived meshes.

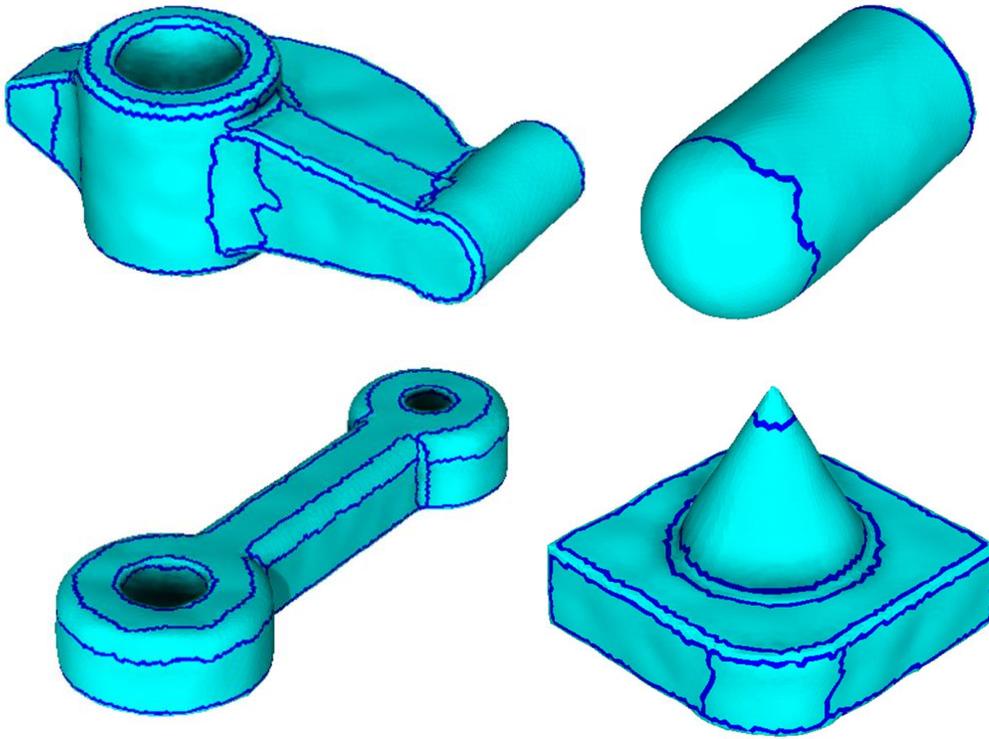


Figure 3.20 Segmentation results of additional synthesized user manipulated meshes.

3.3.3 Scanning-derived and user manipulated meshes

After testing the proposed algorithm using ideal and synthesized meshes to ensure it functions as expected, actual scanning-derived meshes and unconstrained user manipulated meshes are used in order to validate the proposed algorithm for the two worst triangle mesh variations. Comparison with the Local Geometric Filter method [57], seemingly the most capable algorithm to date, is also made.

The previous case studies were done to prove the underlying concept and to validate the performance of the proposed method on the better conditioned input triangle mesh models to the hybrid modeling scheme. It is shown that the developed method processes ideal and non-ideal software generated mesh model very well. The algorithm is able to robustly and automatically

segment the triangle mesh model into appropriate feature patches. However, for this work to be complete, meshes that are derived from real scanned point clouds and real user freehand manipulation need to be processed. It is then only proper to test the proposed algorithm on meshes that are derived from real scan data sets and real user manipulated mesh models. Thus, the following paragraphs showcases the segmentation algorithm, first for real scanning-derived mesh models then for real user manipulated mesh models.

Figure 3.21 shows the results of the proposed algorithm for a sample set of three scanning-derived mesh models. The first Rolling Stage model and the last Pulley model are scanning-derived meshes obtained from the Aim@Shape Shape Repository website while the middle Gear model is a scanning-derived mesh generated in our laboratory using an LDI laser scanner. All three models are noisy, have chamfered edges, and possess surface imperfections, an extra characteristic that synthesized scanning-derived meshes do not have. Since these mesh models are scanned from real objects, any surface imperfections caused by their respective manufacturing process are also translated into the digital domain. The proposed method not only manages to deal with noise and chamfered-edge effect, but it is also largely unaffected by the surface imperfections. As shown in Fig. 3.21, the Local Geometric Filter method does not reliably segment each mesh into its underlying features. Some portions of the models are under-segmented and others over-segmented. The proposed algorithm is able to automatically segment these scanning-derived meshes much more satisfactorily. To quantitatively illustrate the above observation, Table 3.1 summarizes the segmented results of the proposed method along with those of the Local Geometric Filter method for the Rolling Stage, Gear and Pulley models. The table lists the number of overall segmented patches and more importantly, the number of

correctly segmented patches. Note that a patch is considered correctly segmented only when it is neither over- nor under-segmented. For example, an individual plane feature is over-segmented if it is segmented into multiple plane features. The superiority of the proposed method is clearly demonstrated. As for the computational cost, the proposed two-part algorithm did generate the results with some computational time penalty; however, the increased computational load is relatively small as shown in Table 3.2.

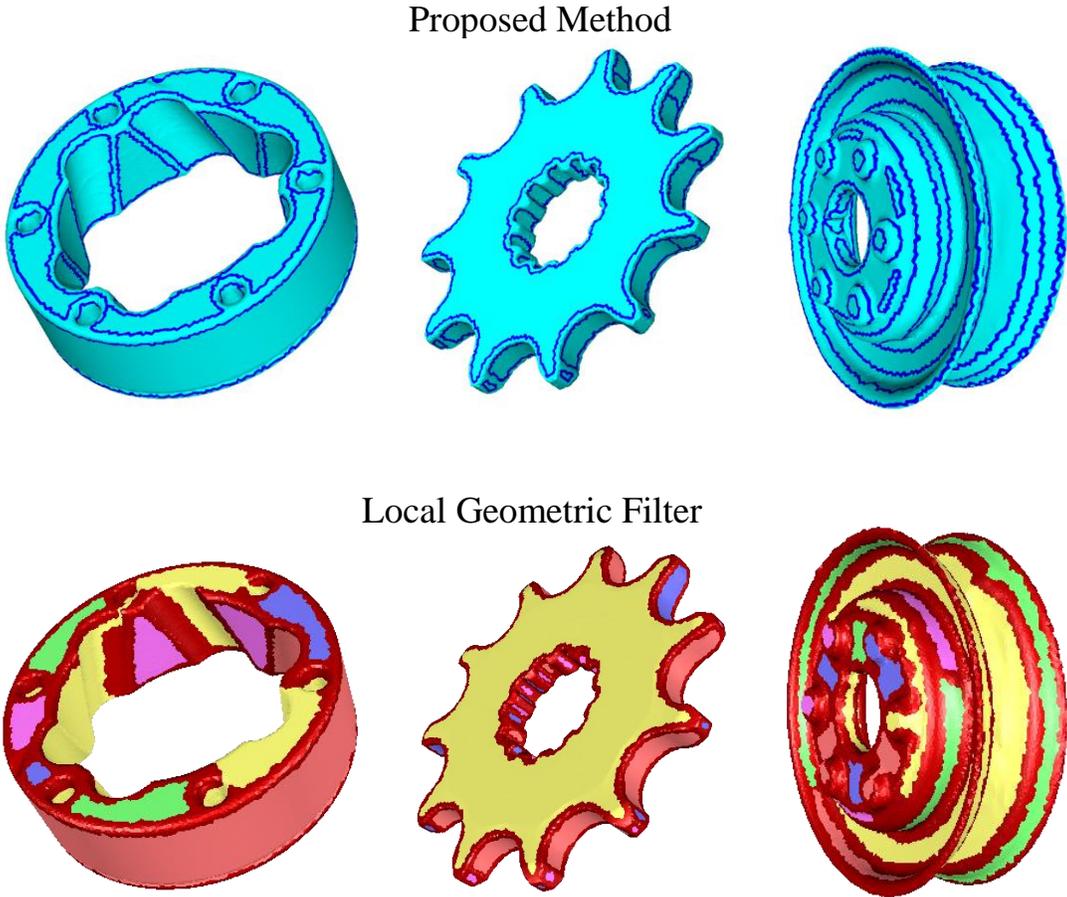


Figure 3.21 Segmentation results of real scanning-derived mesh models.

		Overall Segmented Patches	Correctly Segmented Patches
Rolling Stage (22 Patches)	Proposed Method	22	22
	Local Geometric Filter	16	6
Gear (72 Patches)	Proposed Method	49	47
	Local Geometric Filter	43	29
Pulley (50 Patches)	Proposed Method	55	48
	Local Geometric Filter	49	15

Table 3.1 Number of correctly segmented patches for real scanning-derived mesh models.

	Rolling Stage (61,342 Faces)	Gear (74,584 Faces)	Pulley (73,368 Faces)
Proposed Method (sec.)	6.1	6.7	6.4
Local Geometric Filter (sec.)	5.0	5.2	5.3

Table 3.2 Computational time comparison.

It is observed in Figure 3.21, however, that the blends are not perfectly segmented for the Gear and the Pulley model. This is largely due to mesh density (triangle size) at those locations. Given the noise level and the chamfer-edge effect, the numbers of triangles at those particular locations are inadequate to describe the underlying geometry. If the mesh density is increased at those regions, those features will be appropriately segmented. Overall, the proposed method is able to deal with chamfered edges, noise and model surface imperfections very well. Most, if not all, of the underlying features in real scanning-derived meshes are appropriately and automatically segmented. The feature path boundaries or transitions are well captured without any user input or intervention.

For real unconstrained feature-free user manipulated mesh models, three models (Fig. 3.22) were derived through Sculpttris, a mesh sculpting software [110]. Each model was sculpted by a user, with a given target geometry. The user would start with a basic mesh model which they would deform continuously until it becomes an adequate representation of the final model (deemed by the user). Once these adequate representations are finished, the proposed algorithm is imposed on the mesh model in order to segment these real user manipulated mesh models. Figure 3.22 shows the results of the proposed algorithm on three user sculpted meshes. As seen in the figure, all intended features on the user manipulated mesh are correctly segmented by the proposed approach. On the other hand, the Local Geometric Filter method is unable to segment the user manipulated mesh adequately. Over and under segmentations are seen in the results. The surface undulations serve to confuse the Local Geometric Filter algorithm. With the proposed two part segmentation algorithm, surface undulations do not cause over or under segmentation. The proposed method is robust in dealing with surface undulations; therefore, it produced much more consistent result for segmenting user-manipulated meshes.

The segmentation algorithm, however, does fail when surface undulations become large. Figure 3.23 shows a simple sculpted mesh cube with varying degree of user imposed surface undulations. It is very difficult to quantify surface undulations derived through unconstrained user manipulation since the user mesh manipulations are very complex and inconsistent in nature. Therefore, a qualitative approach is taken here to shown the capability/robustness of the proposed segmentation algorithm when dealing with surface undulations. Figure 3.23 shows the corresponding results of the application of the proposed segmentation algorithm on cubes with varying degree of surface undulations. It can be seen that for those cases where the proposed

algorithm fails, ambiguity in terms of the final intended shape arises, even from a user point of view. The final intended geometry might no longer be just a simple cube, but a juxtaposition of other shapes combined with the cube. Nevertheless, for the cases where the sculpted cube clearly represents a cube, the proposed algorithm segments the sculpted meshes without fail. Thus, this case study shows the developed segmentation algorithm is capable at segmenting unambiguous user manipulated meshes into its underlying elementary features. Its segmentation result are identical to those determined via users.

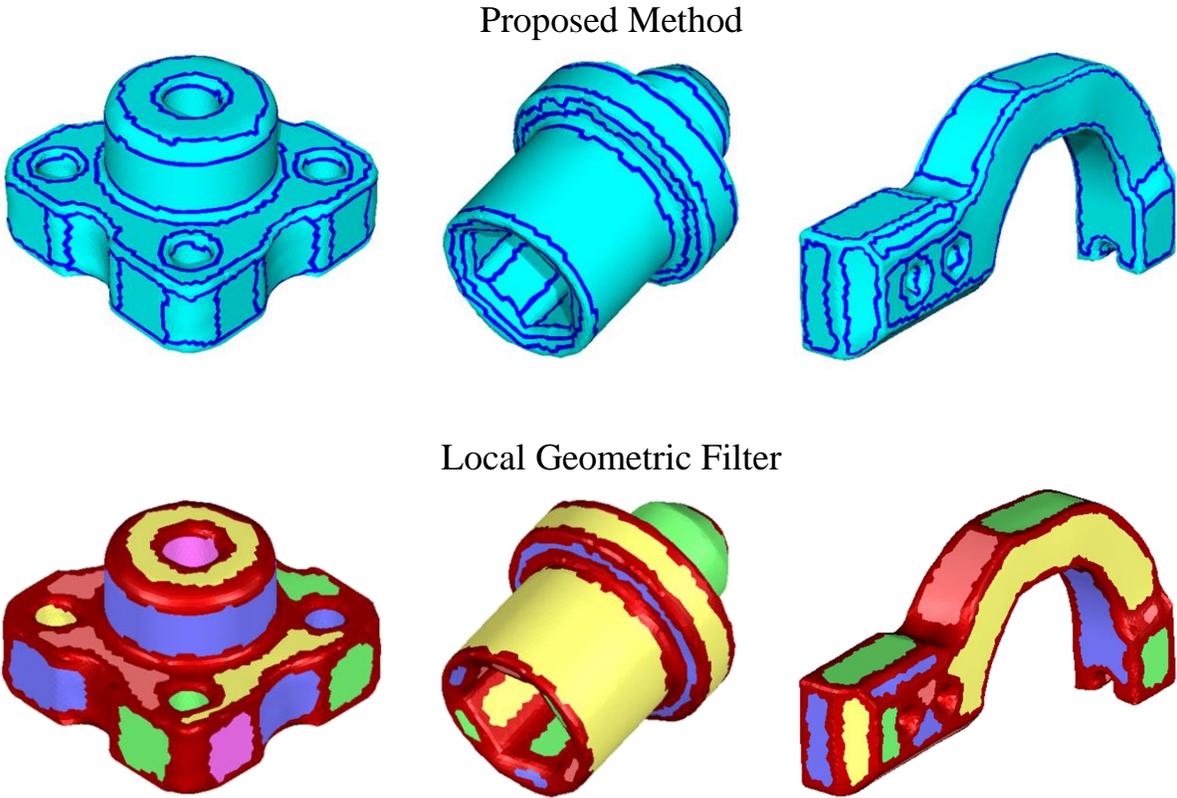


Figure 3.22 Segmentation results of real user-manipulated mesh models.

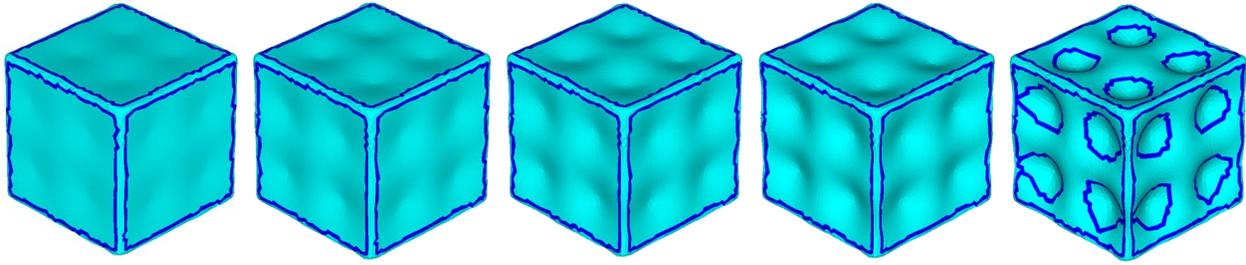


Figure 3.23 Segmentation robustness with regards to surface undulations.

3.4 Summary

The goal of this module is to enable robust feature detection/imposition on featureless triangle mesh engineering models in order to allow for the combination of the freedom and flexibility of mesh modeling techniques with that of the precision and accuracy of parametric CAD modeling. By developing an automated and robust segmentation scheme, features on featureless triangle mesh models can be extracted. Triangle mesh models can carry feature parameter information that is equivalent to that of current parametric CAD models. Parameter changes and feature associations seen in the parametric CAD modeling regime can be brought to the triangle mesh domain without the restrictive nature of individualized features. The segmentation method facilitates the combination of the flexibility in geometric modeling of the triangle mesh representation and the precision and accuracy of current parametric CAD modeling.

Furthermore, the automated nature of the segmentation method facilitates on-the-fly feature segmentation, which enables the ability for the proposed work to segment the unconstrained feature-free user manipulated regions into features as the user models. It gives rise to the ability for the proposed work to deduce the geometry which the user is attempting to model while they model. Users can freely manipulate the triangle mesh model into the desired design with the

hybrid modeler automatically on-the-fly detect the intended design features. Feature becomes an integral part of unconstrained feature-free user mesh manipulation and is implemented such that flexibility of geometric manipulation is not lost.

As shown in the case studies section, unconstrained feature-free user-manipulated meshes can be readily segmented into their underlying elementary features through the proposed segmentation method. Such segmentation capability is achieved by constructing and subsequently conditioning and analyzing a per-vertex mean curvature histogram that treats all geometric elements in the mesh as features. This enables the proposed segmentation algorithm not only to group similar curvature regions into features but also to divide the mesh around edges. The proposed method reliably segments scanning-derived engineering meshes and unconstrained feature-free user manipulated meshes/patches into their underlying prismatic features.

Nevertheless, there are limitations to the proposed method. One apparent limitation is the type of features the proposed method can segment reliably. Free-form features are difficult to extract using the proposed method. Such features are not represented by a single curvature value but rather a gradient of curvature values. As a result, they do not show up as a distinct peak in the curvature histogram but rather a smooth varying curve. If there are features on the mesh that are represented by distinct peaks, the free-form feature have the effect of adding a smooth function to the impulse-shaped histogram seen in this work. The feature represented by a peak can still be extracted by the current method. However, a separate dealing for the free-form features needs to be added, most likely as the third part to the proposed algorithm.

Overall, the proposed two-part algorithm functions well and serves to satisfy the realization of the proposed flexible hybrid CAD modeler. It is able to segment a given mesh model into feature patches automatically. The feature boundaries or transitions are also captured. For scanning-derived mesh models and unconstrained feature-free user manipulated mesh models/patches that do not contain free-form features, the method segments the mesh models into elementary features automatically without user interactions. With the developed segmentation method, the notion of feature is brought to the mesh domain, enabling the integration of flexible mesh modeling methods with that of the accuracy and precision of the parametric CAD regime.

Chapter 4: Mesh Idealization

To fully impose the precision and accuracy seen in the parametric CAD regime on the triangle mesh domain without the loss of modeling flexibility, Mesh Idealization is needed and proposed in this work. Precision and accuracy is only achieved when triangle mesh models ideally represents the features they are composed of. By only segmenting a triangle mesh into feature patches, idealized features are not yet realized; therefore, the ideal triangle mesh model is not yet derived after the preceding module. Precision and accuracy is facilitated but not achieved via segmentation. To fully attain the precision and accuracy seen in the parametric CAD regime and to fully integrate such precision and accuracy with feature-free modeling approach in order to enable flexible design, every segmented feature patch's vertices must reside at the exact location warranted by its identified feature, not only after initial input but at every modeling step.

Ultimately, 3D engineering design models are considered as the ideal models in engineering; therefore, imperfections do not exist in the models. Every model is accurate and precise to its design (design parameters). Any object that exist as a 3D engineering design model must reflect this strictly. And parametric CAD reflects this well. Any model created via parametric CAD software is considered as ideal. All features type and parameters and inter-feature relationships exactly reflect the design. This is achieved, as mentioned prior, through the imposition of the notion of features at a representation level at the expense of modeling flexibility and even speed. Thus, work to transform triangle mesh model into 3D engineering design model is completed in

this module of the thesis in order combine modeling flexibility with that of model precision and accuracy.

For a triangle mesh model to be a 3D engineering design model, the mesh needs to reflect the precision and accuracy required by design models. Any imperfections that deviate the mesh model vertices away from the identified geometry need to be fixed and any imperfections that cause the mesh features to deviate from their ideal relationships have to be repaired. Furthermore, feature intersections need to be processed to accurately represent the ideal feature intersections. The triangle mesh model cannot be only visually correct. It has to be numerically accurate.

To ensure that triangle mesh model properly represents 3D engineering design model, a robust Mesh Idealization algorithm is developed and outlined in this section. Any mesh input to the proposed hybrid modeling scheme and all user manipulated triangle meshes are idealized via the proposed method. Triangle meshes are automatically repaired to accurately represent its features and their relationships. This allows the realization of accurate and precise triangle mesh model at any modeling step for the hybrid modeler. With this idealization module, precision and accuracy is automatically combined with the flexibility seen in the triangle mesh domain, facilitating flexible geometric design. Such combination is not possible in the current rigid parametric CAD regime.

For the idealization scheme, the imperfections can be broken down into two types (treated equivalently): input imperfection and unconstrained feature-free modeling imperfections. Input imperfections are those that are caused primarily by geometry capturing devices such as 3D laser

scanners and unconstrained feature-free modeling imperfections are those that are caused by the user's free manipulation of the triangle mesh model. In essence, the work again tailors the algorithm to process scanning-derived triangle prismatic engineering meshes and unconstrained feature-free user-manipulated triangle meshes/patches, the two worst input triangle mesh varieties for the hybrid modeling scheme. The goal for this module is to bring forth precision and accuracy for all mesh input variations to the flexible hybrid modeler. Any triangle mesh inputs, regardless of their sources, are transformed into ideal design models and any freely user-manipulated regions, regardless of deformation methods, are transformed into ideal design features.

To idealize triangle mesh models to an accurate design model a simple movement, which translates non-ideal mesh vertices onto the surface of its identified feature, seemingly suffices; however, such action only repairs the mesh model on a per-feature basis. The interdependency of the features are not looked at nor used in the repair process. In order to fully idealize and to appropriately impose features on a mesh model, the feature interdependencies need to be considered. A model can only be considered ideal when it is geometrically similar and topologically identical to that of the original or, in the case of design, to that of the design idea. In other words, the model needs to both contain the correct features (types and numbers) with similar to original parameters or to design idea "sizing" (geometrically similar) and to have the correct inter-feature-relationships that are equal to the original model or to the design idea (topologically identical). For idealization performed on the input triangle meshes, it is not expected that the idealized model is an exact replica of the original which the input mesh is derived from; however, it is expected that the idealized model is a topological replica of the

original with geometric parameters similar to that of the original model. For idealization performed on the unconstrained feature-free user manipulated models/regions, it is expected that the idealized model is a close representation of the design idea. Close representation entails geometric similarity and topological identity, since in order to closely represent a geometric design idea; the feature setup (topology) has to be equivalent to that of the design idea. The parameters do not have to be equal, they only have to be similar to the user intended parameters.

In summary, this module of the thesis sets out to automatically achieve geometrically similar and topologically identical idealized mesh models in order to bring forth the model precision and accuracy needed to transform triangle mesh models into 3D engineering design models. With this module, triangle-mesh models are fully transformed into 3D engineering design models and the proposed flexible hybrid modeler is further realized.

4.1 Methodology

In order to ensure that geometry and topology take precedence and to ensure that feature information is present, this chapter proposes a direct non-solver-based idealization approach where the mesh models are idealized through exploiting CAD modeling workflow and inter-feature dependency. By leveraging common CAD modeling workflow knowledge along with utilizing inter-feature dependency within the model, the sequence of which the part is likely modeled is estimated. Having this sequence allows the proposed scheme to quickly and automatically idealize the non-ideal triangle mesh model in a manner very similar to that of a designer's modeling approach when using a CAD software tool. The scheme essentially mimics the user operations. An outline of the proposed method as a flow chart is depicted in Fig. 4.1.

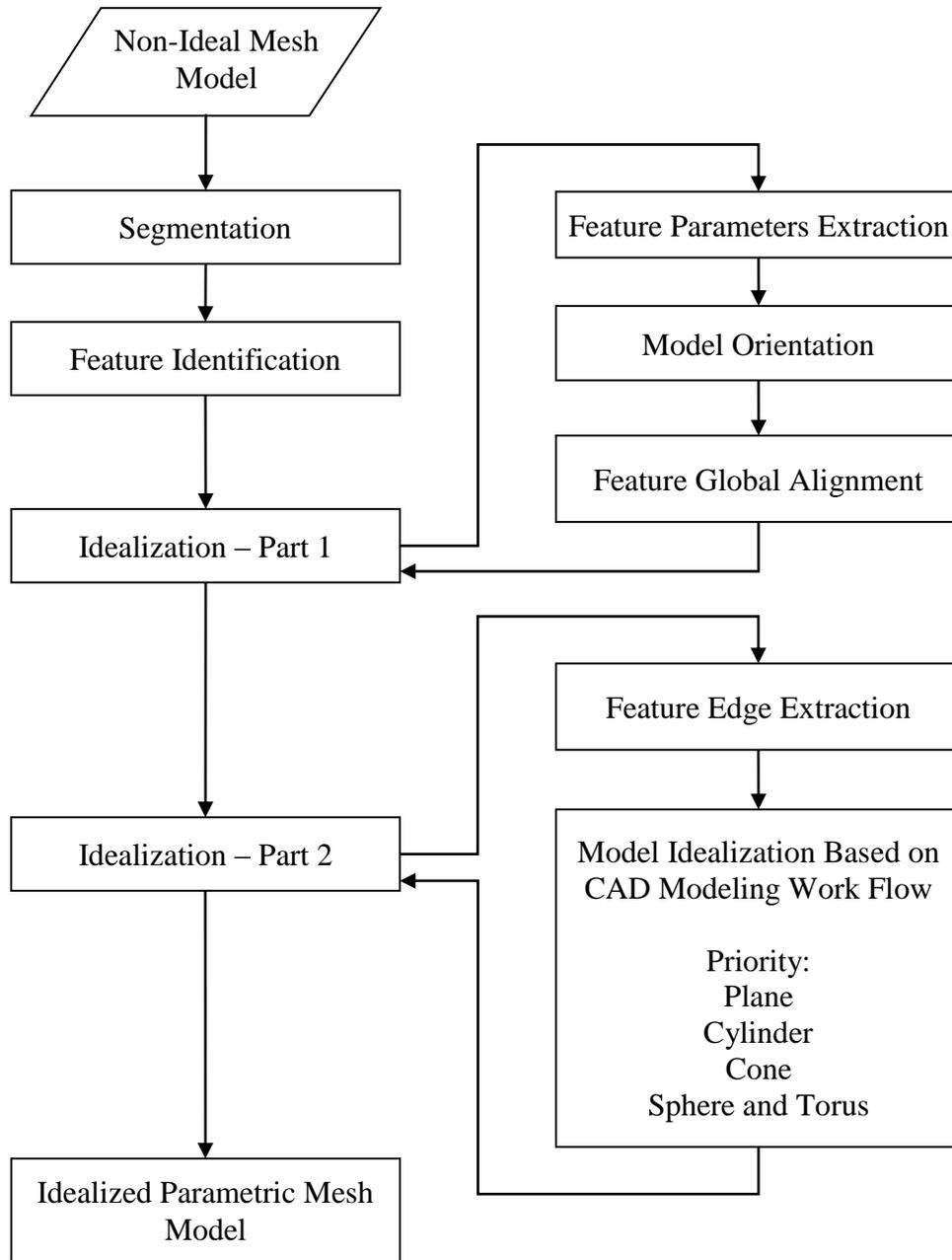


Figure 4.1 Proposed mesh-based idealization scheme.

The said CAD modeling work approach is derived through the most primitive method of digitizing a physical part into an ideal parametric CAD model, measuring and remodeling [111].

The designer studies the part and makes the best experienced decision regarding the features which the part is constructed from. Measurements relevant to define these features are taken and the geometric relationships among the features are estimated. The part is then reconstructed with the above information into a digital version in a parametric CAD software. The present work takes this one step further by automatically completing this task for the designer. To completely automate this process, the logical step is to replace the designer with a set of rules that best reflect the designer's insights. Such insights include the understanding of shape which allows them to extract features (types and parameters) from the model, the typical remodeling process (CAD modeling work flow) which the designer follows for remodeling and the geometric cues which the design uses in order to make the proper judgments.

Therefore, the first step to idealizing a mesh model of a prismatic engineering part is to extract and identify the features that are present on the featureless triangle mesh model and to also determine the relationships among these features. Much like the thinking of a designer, the features that are present on a physical object and their relationships are extracted through reasoning of the apparent geometric cues. These reasoning and geometric cues are summarized in the previous segmentation module. Effectively, the segmentation algorithm replaces the user who manually segments an object via their cognitive reasoning. The mean-curvatures functions as the geometric cues and the two-part per-vertex mean curvature-histogram algorithm functions as the reasoning. Therefore, with the segmentation algorithm, users are mimicked and the input meshes are reliably segmented into feature patches.

With the input triangle mesh segmented into patches, the exact feature type and parameters are determined next. Each feature patch is identified into the feature it best represents with its feature parameters subsequently extracted. Note, up to this point in the thesis, the segmented features have not been identified through any means. The triangle mesh is segmented into different patches due to the difference in geometric cues but these patches are not yet identified. The identification is done in this module. After identification and feature parameters extraction, the model is oriented to the system's coordinate system and the feature parameters are globally aligned to the ideal axes. The mesh patches are idealized based their corresponding determined feature types and extracted feature parameters. With the first part of the idealization algorithm complete, the mesh patches are further idealized via CAD Modeling Workflow, the second part of the idealization algorithm. The mesh is idealized as if a user is remodeling the object.

Essentially, the two part idealization algorithm breaks the mesh model feature parameters down into two categories, independent parameters and dependent parameters. Independent parameters are those that are not related to other feature parameters in the model and dependent parameters are those that are derived either from independent parameters or from previously idealized dependent parameters. The proposed two-part algorithm, therefore, idealizes the independent aspects of the features first then the dependent portions of the features following the CAD modeling workflow. This ensures that a dependent parameter is always idealized before it is used by another feature's dependent parameters. In other words, the idealization algorithm idealizes the features with independent parameters first and sets these features as the seeds for further idealization. The idealization then grows based on the CAD modeling workflow to idealize dependent parameters until all parameters are idealized.

4.2 CAD modeling workflow

Since the proposed method hinges heavily on CAD modeling work flow, the typical CAD modeling workflow [111-114] is studied in order to determine the path on which the idealization grows. In terms of CAD modeling workflow for modeling prismatic engineering parts within a CAD environment, the designer typically starts with the very basic element of planes. Planes are the basic building blocks of most engineering parts and they often serve as datum for other features in the model. If a plane is present, the location of the plane and its orientation are independent parameters. Cylinders are usually added next and depending on the related geometric cues, they can be dependent or independent on the planes. If dependent, the dependency between the cylindrical and plane features is established. Modeling of dependent-parameter features (cylinders) based on independent-parameter features (planes) is performed. At this point, a check for conflicting constraint is executed. If a conflicting constraint exists, actions are taken to reduce them to redundant constraints. Once planes and cylinders are added and their relationships are imposed, cones are inserted next. This is because spheres and tori are typically present in prismatic engineering models as part of blends, which are normally added to the model last. Relationships between cones and planes/cylinders are established. If a dependency exists, the dependent cone parameters are modeled based on either features with independent parameters or features with idealized dependent parameters. Again, the designer checks for conflicting constraints. Spheres and tori are added last using the same procedure.

Therefore, according to the typical CAD modeling workflow, the idealization priority is set to planes, cylinders, cones, and spheres and tori, with planes possessing the highest priority and

spheres and tori the lowest. Dependency conditions between each of these feature types are also established based on the same priority. For example, if a plane exists for a given model, its location and orientation are set as the starting independent parameters. If a cylinder is tangentially continuous to the plane, then the radius and orientation of the cylinder is idealized base on the independent parameters of the plane. If a model does not contain any planes but cylinders and spheres, the cylinder parameters are set as the independent parameters. A sphere, if tangentially continuous to the cylinder, is idealized based on the cylinder's independent parameters. This CAD modeling work flow manifests itself in most CAD software when models are constructed from scratch [113]. Fig. 4.2 illustrates this typical flow. With the priority presented above, the proposed algorithm automatically idealizes triangle meshes through mimicking the typical manual part digitization remodeling steps of a designer.

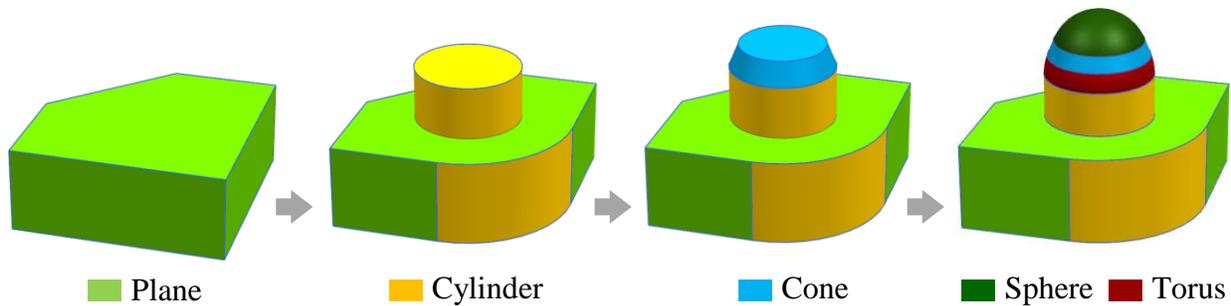


Figure 4.2 Typical CAD modeling workflow.

One thing to note is that a model can have multiple independent features and thus, multiple seeds and paths which the idealization grows on. Also, one seed can feed multiple paths. The relationship is not bijective. Therefore, depending on the mesh model, the algorithm is completely sequential or is highly parallel. In some cases, conflicting or redundant constraints

can occur. Fortunately, since the proposed method does not utilize constraint solvers, the solvability is not affected and a solution can always be found. Furthermore, the CAD modeling workflow checks for conflicting constraints and reduces them down to redundant constraints. With the non-solver approach, redundant constraints do not pose any issues and therefore, properly idealized mesh model is straightforwardly found. Since typical CAD modeling workflow is always followed on every idealization path, the solution is not compromised and is still an accurate representation of the designer's approach. Further details are given in the next section.

4.3 Detailed procedure

Before digging into the details of the proposed method, the type of inputs and their properties need to be clearly stated or rather, clearly reminded. The type of inputs the proposed flexible hybrid CAD modeler is designed to process ranges from ideal prismatic engineering triangle meshes to scanning-derived prismatic engineering triangle meshes and unconstrained user manipulated meshes. As mentioned previously, scanning-derived meshes and unconstrained user manipulated meshes are the two worst variations which the flexible hybrid modeler has to process. The scanning-derived meshes are plagued with noise and chamfered-edge effects while the unconstrained user manipulated meshes are plagued with chamfered-edge effects and surface undulations. Noise and surface undulations displace the vertices of the mesh such that they are no longer at the ideal locations and chamfered-edge effect corruptions the edges of the part. For scanning-derived meshes, the chamfered-edge effect is a result of the discrete 3D capture devices while for user manipulated meshes the chamfered-edge effect is a result of shape manipulations.

With the combined effect of chamfered edges and noise for scanning-derived meshes and chamfered edges and surface undulations for user manipulated meshes, care is needed when processing these meshes. Since they are the most challenging to process mesh variations, the idealization algorithm is tailored to tackle scanning-derived meshes and unconstrained user manipulated meshes.

4.3.1 Feature identification

With the feature segmentation results readily available from the previous module, the next logical step is feature identification. At this point the mesh model is segmented into patches through difference in geometric measure but the features which they represent are not yet identified. To identify the features, the method based on mean and Gaussian curvatures by Besl and Jain [115] has been extended. By studying the signs of the average mean and Gaussian curvature of each segmented patch, the shape of each segmented patch can be quickly identified. However, since scanning-derived mesh models and unconstrained user manipulated mesh models/regions are the inputs, noise and surface undulations are present to corrupt the curvature values. In order to determine the signs for the average mean and Gaussian curvature for each segmented patch, the curvature value range which effectively represents zero (planes) for the given triangle mesh model is required. The histogram-based approach developed for feature segmentation is used in order to automatically determine the range of curvature values that represents zero (Fig. 4.3).

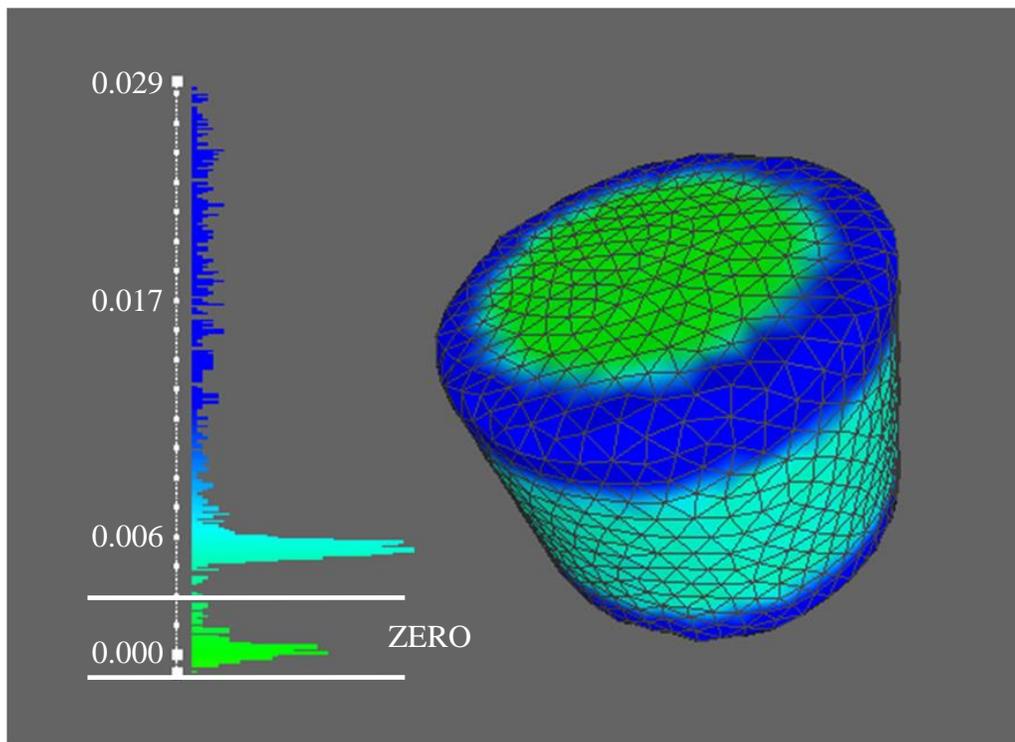


Figure 4.3 Feature identification: effective zero determination.

To determine the range, the per-vertex Gaussian curvature of the mesh model is calculated and the associated histogram is constructed via the same method proposed for the mean-curvature histogram construction for segmentation in Section 3.2.1. Equal amount of Laplacian smoothing is applied to the per-vertex Gaussian curvature values as to the per-vertex mean-curvature values during histogram conditioning. The mean-curvature histogram determined by the proposed segmentation method is also used. Once the histograms of the two types of curvatures are constructed, monotonic smoothing (Section 3.2.1.2) is applied to the per-vertex curvature values which leads to two smoothed and well-behaved histograms. Unwanted noise effect is removed. If a plane exists in the model, the valleys adjacent to the peak on the histogram which represent

the plane (considered to have zero curvature) are extracted from both histograms. These valleys mark the effective zeros of the mean and Gaussian curvature histograms. If planes do not exist in the model, the width of the valley of the two closest peaks to zero is used. With the effective zero ranges determined, the segmented feature patches are categorized into planes, cylindrical features and spherical features. Cylindrical and spherical features are listed because cones and tori cannot be straightforwardly separated from cylinders and spheres, respectively, by using the standard mean and Gaussian curvature sign method. Thus, the average mean and Gaussian curvature sign method is extended.

Since mean curvature is the average value of the principal curvatures (maximum and minimum) and Gaussian curvature is the product of the principal curvatures, per-vertex principal curvatures can be calculated from the mean and Gaussian curvatures. Two more histograms are then constructed from the maximum and minimum principal curvatures. The effective zero for each principal curvature is again determined and the per-feature-patch per-vertex principal curvature distributions are used to distinguish cones from cylinders and tori from spheres using the curvature rules illustrated in Fig. 4.4.

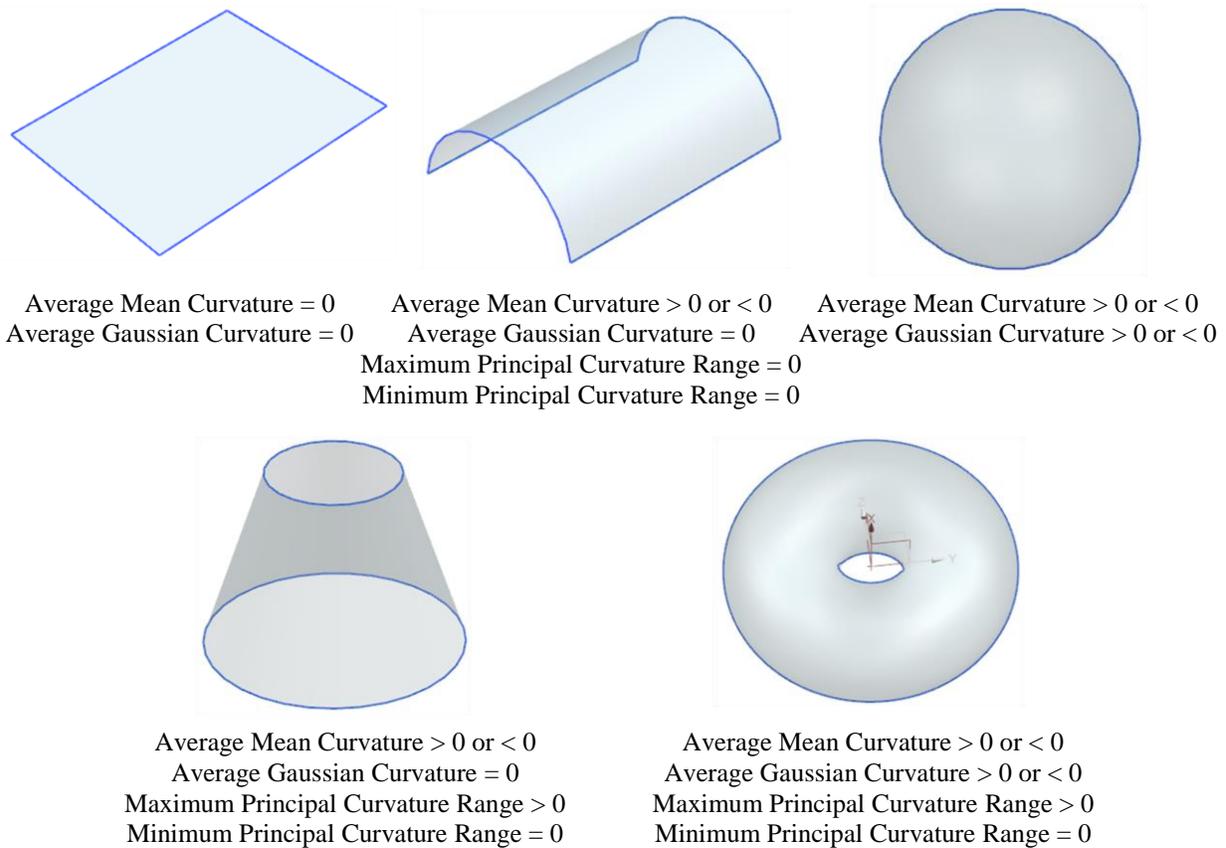


Figure 4.4 Feature identification: detailed curvature rules.

4.3.2 Idealization: part 1

Once the feature types are identified, the feature parameters are determined. Instead of using a fitting method that relies on time-consuming nonlinear search, direct calculation through discrete differential geometry is used. The feature parameter extraction hinges on three geometric measures: per-vertex mean curvature, per-vertex Gaussian curvature and per-vertex normal [107]. Per-vertex maximum and minimum principal curvature are used. One thing to note is that these geometric measures used are recalculated individually for each feature patch in order to

ensure the most accurate feature parameter extraction results. With this individualization, each feature's geometric measures do not affect another feature's parameter extraction. The previously calculated per-vertex curvature values are thus not utilized here in order to ensure better accuracy.

4.3.2.1 Parameter extraction

Before utilizing the three curvature geometric measures, they are smoothed by Laplacian smoothing for each feature patch. This is to ensure that the effect of any noise on the mesh does not much affect the geometric measures. An iteration value of six is found to work well for all cases tested which have varying feature patch sizes and mesh densities. The smoothing is applied to each patch individually in order to minimize cross contamination among the feature patches. Fundamentally, the smoothing is done per feature to allow vertices with similar geometric measures to blend in order to provide better values. With the smoothing completed, the feature parameters are extracted. Table 4.1 outlines the parameters used to describe each type of features and how these parameters are determined from the discrete geometric measures.

Plane	Origin	$O_{plane} = \frac{\sum_{i=1}^m P(i)}{m}$	P : vertex location m : number of points per patch
	Axis	$A_{plane} = \frac{\sum_{i=1}^m \hat{n}(i)}{m}$	n : per-vertex normal vector
Cylinder	Radius	$R_{cyl} = \frac{\sum_{i=1}^m R_c(i)}{m}$ $R_c(i) = \frac{1}{2 * \kappa_m(i)}$	κ_m : per-vertex mean curvature
	Origin	$O_{cyl} = \frac{\sum_{i=1}^m O_c(i)}{m}$ $O_c(i) = P(i) - \hat{n}(i) * (R_{cyl})$	
	Axis	$A_{cyl} = \frac{\sum_{i=1}^{m-1} A_c(i)}{m}$ $A_c(i) =$ $\text{Norm} \left(\hat{n}(i) \times \left((P(i) - O_{cyl}) \times \hat{n}(i) \right) \right)$	$if \text{Angle}(A_c(i), A_c(i + 1)) < \pi/3$ where $\text{Angle}(A, B) = \cos^{-1} \frac{A * B}{\ A\ * \ B\ }$ and $\text{Norm}(C) = C / \ C\ $
Torus	Minor Radius	$R_{torus_minor} = \frac{\sum_{i=1}^m \frac{1}{\kappa_2(i)}}{m}$	κ_2 : per-vertex minimum principal curvature
	Major Radius	Convex $R_{torus_major} = \max \left(\frac{1}{\kappa_1(i)} \right) - R_{torus_minor}$ Concave $R_{torus_major} = \max \left(\frac{1}{\kappa_1(i)} \right)$	κ_1 : per-vertex maximum principal curvature
	Axis	$A_{torus} = \text{Norm} \left(\frac{\sum_{i=2}^m O_{minor}(1) \times O_{minor}(i)}{m - 1} \right)$ $O_{minor}(i) = P(i) - \hat{n}(i) * R_{torus_minor}$	
	Origin	$O_{torus} = \frac{\sum_{i=1}^m O_{minor}(i) - \hat{n}_p(i) * R_{torus_major}}{m}$ $\hat{n}_p(i) = \text{Norm}(\hat{n}(i) - (\hat{n}(i) * A_{torus}) * A_{torus})$	

Sphere	Radius	$R_{sphere} = \frac{\sum_{i=1}^m \frac{1}{\kappa_m(i)}}{m}$	
	Origin	$O_{sphere} = \frac{\sum_{i=1}^m P(i) - \hat{n}(i) * R_{sphere}}{m}$	
Cone	Radius (max)	$R_{cone_max} = \frac{\sum_{i=1}^m \frac{1}{\kappa_1(i)}}{m}$	
	Axis	Set $\hat{n}_{original}(i) = \hat{n}(i)$ Iterate (100x) { $O_{cone} = \frac{\sum_{i=1}^m P(i) - \hat{n}(i) * \frac{1}{\kappa_1(i)}}{m}$ $A_{cone} = \text{Norm} \left(\frac{\sum_{i=1}^m P(i) - \hat{n}(i) * \frac{1}{\kappa_1(i)} - O_{cone}}{m} \right)$ $\hat{n}(i) = \hat{n}(i) - (\hat{n}(i) * A_{cone}) * A_{cone}$ }	<i>if</i> Angle($A_{cone}(i), A_{cone}(i + 1)$) < $\pi/3$
	Cone Height	$H_{cone} = h_1 + h_2$ $h_1 = \max \left(\left(P(i) - \hat{n}(i) * \frac{1}{\kappa_1(i)} \right) - O_{cone} \right)$ $h_2 = \max \left(\left(P(i) - \hat{n}(i) * \frac{1}{\kappa_1(i)} \right) - O_{cone} \right)$	<i>if</i> Angle(h_1, h_2) > $\pi/3$
	Cone Angle	$\alpha_{cone} = \sum_{i=1}^m \text{Angle}(\hat{n}_{original}(i), \hat{n}(i))$	
	Origin	$O_{cone} = O_{cone} - A_{cone} * h_1$ $O_{cone} = O_{cone} - A_{cone} * h_2$	<i>if</i> $R_{cone}[h_1] > R_{cone}[h_2]$ <i>if</i> $R_{cone}[h_1] \leq R_{cone}[h_2]$

Table 4.1 Feature parameter extraction.

4.3.2.2 Model orientation and global feature alignment

Once the feature parameters are extracted, the mesh model is oriented and the features are globally aligned. To establish one of the many possible proper orientations of the model with respect to the coordinate system, the feature with the highest priority in terms of CAD modeling workflow is used. If planes exist, the largest plane is to be aligned to the XY plane. Another feature on the model with an axis component that is orthogonal to the plane axis is used to complete the orientation. If such feature does not exist within the model, the model's final orientation step is not critical and can be arbitrarily set. In cases parts do not contain planes, the next feature type with the highest priority in terms of CAD modeling workflow is used. If the designer believes at this point that a more favorable orientation is preferred, the orientation can be set manually.

With the model oriented, tolerance-based global alignment is performed on all features axes assuming they are all independent parameters. As stated previously in the CAD modeling workflow section, any feature in the model can have completely independent parameters. Furthermore, alignment of features for an idealized model can always be done with respect to the reference coordinate system, independent with other features, as long as the appropriate idealization axes are specified. In other words, the relationship of a feature's axis with the reference coordinate system determined directly or indirectly through other feature's axes will lead to identical alignment results.

In this work, the angular tolerance control and the feature axis tolerance control are used for global alignment. The angular tolerance defines the angular deviation allowed for a feature axis

from the defined ideal axis. The feature axis tolerance defines the linear deviation allowed for a feature axis from the ideal axis (normalized). It has been found that axes of 10°, 30°, 45° and 60° from the reference coordinate system axes are the most frequently encountered ideal axes for most mechanical parts. It has also been found that an angular tolerance of 5° and an axis tolerance of 0.1 mm work very well for most scanned-derived mesh models and user freely manipulated mesh models. With the feature parameters extracted, model oriented and features aligned, the individual feature patches are idealized. This is done by shifting each non-ideal feature point on the mesh model to a corresponding ideal feature point. Such an incremental point moving scheme is beneficial especially when idealizing of the model edges as detailed in the next subsection.

4.3.3 Idealization: part 2

Before part two of the algorithm takes place, some specific segmented mesh patches, referred to as “edges” previously, need to be lumped with its adjacent feature patches. These edge patches include the sharpest feature patches and the one-triangle strip transitions between features (colored blue), both of which serve as boundaries of features. These patches have been kept separate from the feature patches to allow inter-feature dependency extraction and to ensure reliable feature identification and feature parameter extraction. However, these edge patches do belong with the feature patches and do need to be reintegrated. To accomplish this, the edge which lies within each edge patch is first extracted. With the edge extracted, the edge patch is divided by the edge. Each divided sub-patch is integrated with its adjacent feature patch and subsequently takes on the adjacent feature’s type and parameters. Please note, the sharpest

feature patch is now colored blue and is thus integrated with the one-triangle strip transitions after this point of the thesis. The edge patches consist of two patch types but are colored equivalently since they are processed equally.

To extract the edge, a progressive erosive approach is employed where the edge is extracted by integrating the triangle which least likely contains an edge into its adjacent feature patch. The likelihood of a triangle is evaluated based on the average mean curvatures of its three vertices. For a given transition patch, the likelihood values are calculated for all mesh triangles within the transition patch and ordered into a priority queue with increasing likelihood value. The algorithm then removes triangles one by one from the transition patch based on the queue and integrates these triangles with their adjacent feature patches. The triangle removal is allowed as long as it does not break the edge. When no more triangles can be removed, a one triangle wide strip is left. The edge is then extracted by further eroding the triangle strip by successively removing the triangle edge with the lowest dihedral angle. Again, the removal is allowed as long as it does not result in a broken edge. Figure 4.5 illustrates the edge extraction process.

With the edge extracted and the mesh triangles integrated with their adjacent feature patches, the transition patch triangles are to be idealized. Since CAD modeling workflow has not been imposed yet, the edges do not correspond to the final idealized edges. However, the edges do reflect the intersection between features and provide good initial conditions for feature parameter recalculation when performing idealization based on CAD modeling workflow. Essentially, after this step, an idealized parametric mesh model with sharp edges is present. Any required tangential continuity is to be achieved by altering the involved features and edges. In order to

ensure that intersections can always be found, an inflation technique is used for tangentially continuous edges. As depicted in Fig. 4.6, extracting feature parameters separately for tangentially continuous features often result in features that do not intersect. In order to guarantee intersection and facilitate quick edge calculation, the cylindrical (or spherical) feature's radius is inflated 20 times of its original radius. It should be noted that the edges are to be made as close to the conceivable originals as possible.

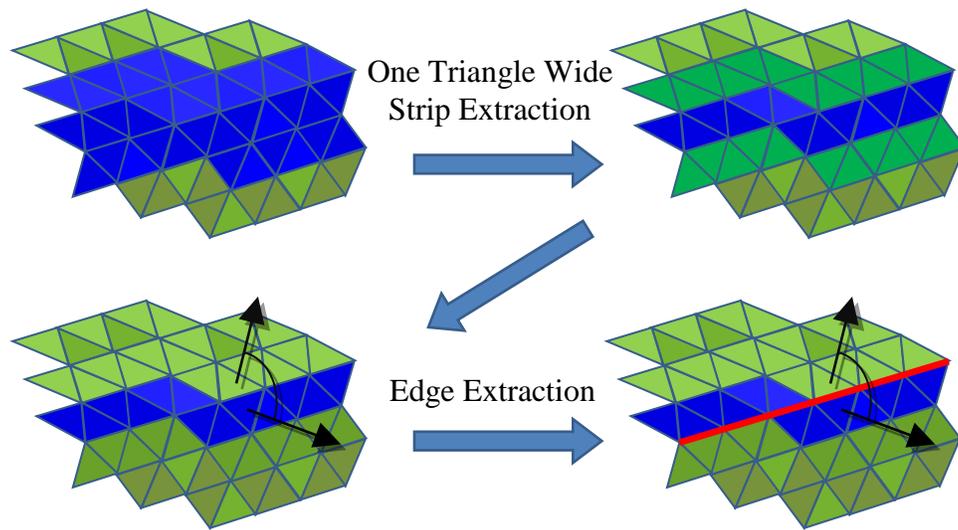


Figure 4.5 Edge extraction in a transition patch.

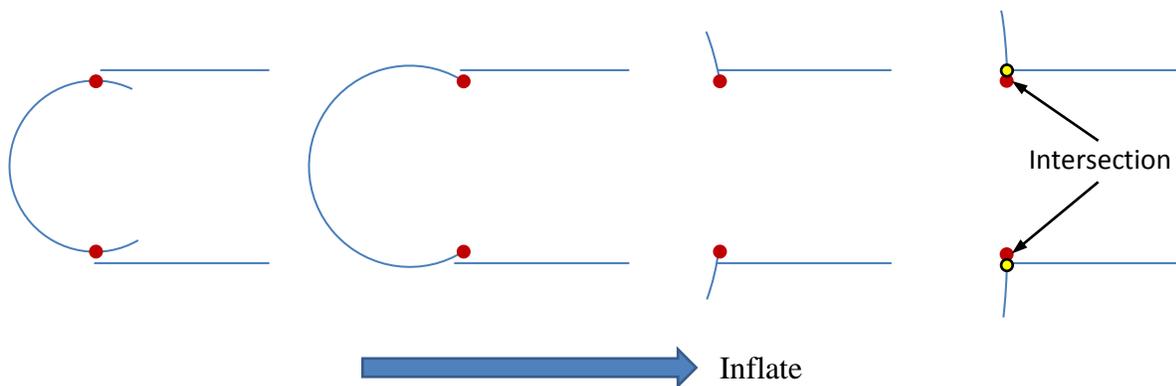


Figure 4.6 Inflation technique to determine tangentially continuous edges.

CAD modeling workflow is now ready to be applied to completing the idealization of the scanning-derived mesh model. As outlined previously, features with independent parameters are set as seeds and the idealization grows from these seeds according to the CAD modeling workflow. The following subsections expand on this basic procedure and illustrate the method in detail for each feature type assuming planes are the initial seeds.

4.3.3.1 Cylinders

Having assumed planes as the initial seeds, the next in line in the workflow in terms of priority are cylinders. There are three variations of how a cylinder relates to its adjacent planes, as illustrated on a 2D projection plane in Fig. 4.7: self-defined, semi-defined, and completely-defined. The self-defined variation involves completely independent parameters. The semi-defined variation involves the adjacent planes contributing to determine some but not all of the parameters. Feature parameters of the completely-defined variation are completely determined by the adjacent planes. For a self-defined cylinder, its parameters are completely independent and therefore, there is no need to recalculate them. These self-defined cylinders then become seeds themselves for features in the CAD modeling workflow with lower priorities. For a semi-defined cylinder, the radius is to be recalculated using the adjacent plane feature that is tangentially continuous to the cylinder. The center is not recalculated. The new radius is

$$R_{cyl} = |(O_{plane} - O_{cylinder}) * A_{plane}| \quad (1)$$

where A_{plane} is the orientation vector of the adjacent tangentially continuous plane and O_{plane} and $O_{cylinder}$ are the centers of the plane and cylinder, respectively. For a completely-defined cylinder, both the radius and the center are recalculated from the adjacent planes.

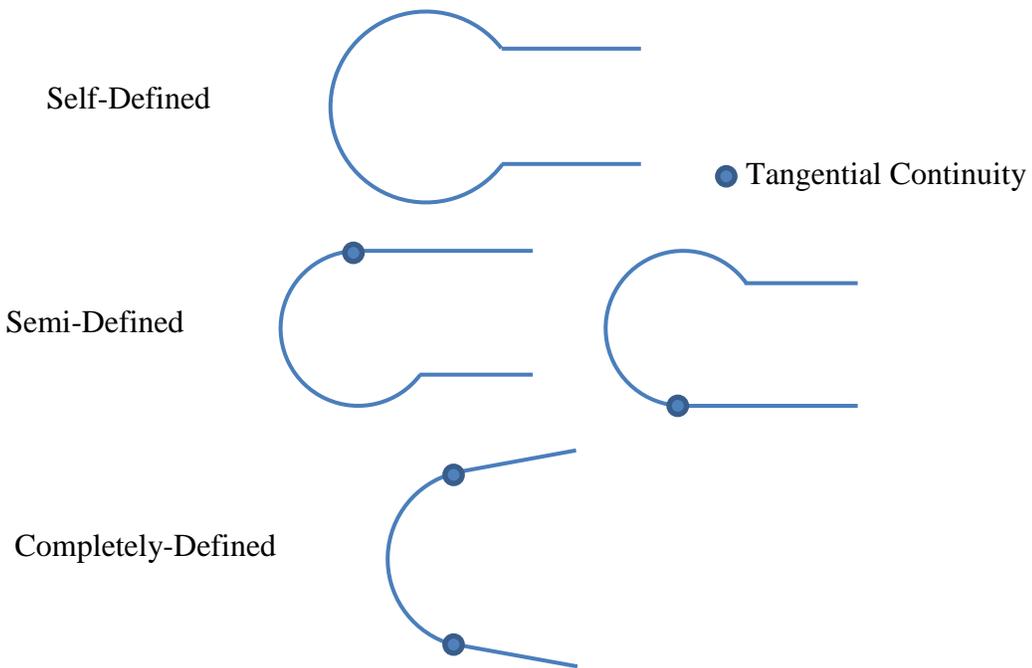


Figure 4.7 Example of how a cylinder relates to its adjacent planes.

Once every cylinder in the mesh model is idealized, the proposed method checks the cylinder radii against the maximum principal curvature histogram. Cylinders with 180° span (half cylinders) are not checked since these cylinders cannot have their radii altered without violating the continuity conditions with their adjacent features. For non-half cylinders, the cylinder radii are set according to the corresponding peaks in the curvature histogram. In order to ensure that

the change in the cylinder radii does not affect the tangential continuity conditions, the centers of the affected cylinders are updated using the following equation:

$$O_{cylinder} = O_{cylinder} + \frac{A_{plane1} + A_{plane2}}{2} * (R_{cyl} - R_{cylnew}) / \cos(\text{Angle}(A_{plane1}, A_{plane2}) / 2) \quad (2)$$

The orientation and location of every cylinder are also checked. If it is found that some cylinder has the same orientation as another cylinder and that their location falls along the same axis (within a set tolerance), the cylinders are considered coaxial. They are aligned on to the same axis. The tolerance for the axis is determined during global alignment. During the global alignment step, the difference between the ideal and non-idealized axis is projected on to the plane described by the ideal axis and its corresponding ideal feature center point. The projected length is the said tolerance for a given cylinder. The tangential edges are updated for completely defined cylinders through

$$P_{edge_1}(j) = O_{cylinder} + \left((P_{edge_1}(j) - O_{cylinder}) * A_{cylinder} \right) * A_{cylinder} + \text{Norm}(\left((O_{plane_1} - O_{cylinder}) * A_{plane_1} \right) * A_{plane_1}) * R_{cyl} \quad (3)$$

$$P_{edge_2}(k) = O_{cylinder} + \left((P_{edge_2}(k) - O_{cylinder}) * A_{cylinder} \right) * A_{cylinder} + \text{Norm}(\left((O_{plane_2} - O_{cylinder}) * A_{plane_2} \right) * A_{plane_2}) * R_{cyl} \quad (4)$$

where $P_{edge_1}(j)$ and $P_{edge_2}(k)$ are points on the tangential edges associated with plane 1 and 2, respectively. For cases where cylinders reside adjacent to the cylinder-of-interest the same method outlined above is used. The adjacent planes' directional normals (axes) are replaced with the average edge normals at the intersecting edge of the two cylinders.

4.3.3.2 Cones

With planes and cylinders idealized, the idealization path continues to cones. Some sample variations for cones are shown in Fig. 4.8. Again, parameters do not need to be recalculated for self-defined cones and they become seeds for features in the CAD modeling workflow with lower priorities. For a semi-defined cone, an adjacent feature typically defines the orientation, location and maximum radius of the cone. Thus, these parameters are directly inherited from the adjacent feature. For a completely-defined cone, adjacent plane features fully constraint the cone through defining the cone angle as

$$\alpha_{cone} = \frac{\pi}{2} - \text{Angle}(A_{plane}, A_{cone}) \quad (5)$$

where A_{cone} is the current cone axis and A_{plane} is the orientation vector of one of the adjacent planes. This is then followed by updating the cone center and the tangential edges of the cones.

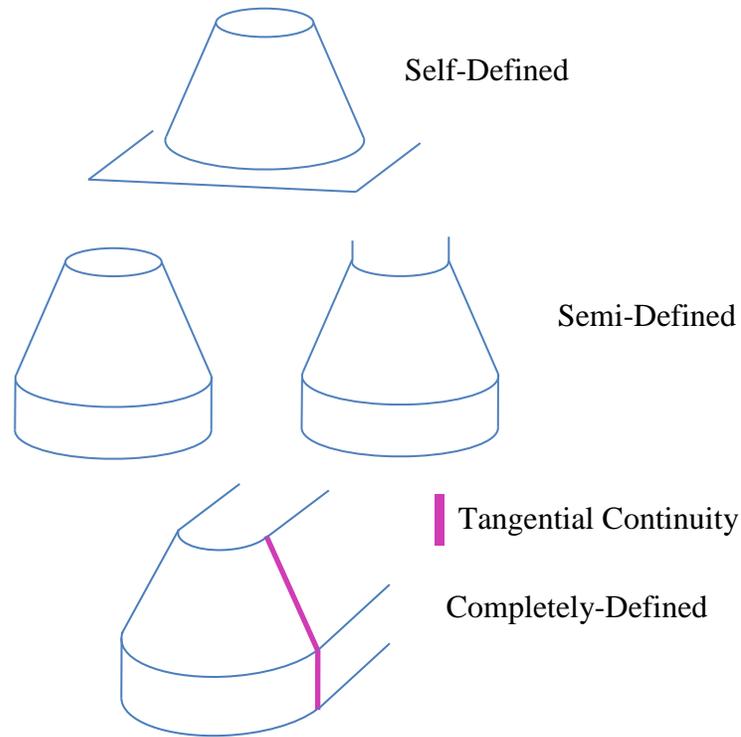


Figure 4.8 Sample cone variations.

4.3.3.3 Spheres

With planes, cylinders and cones idealized, the next feature type to be idealized in terms of CAD modeling workflow is sphere. For spheres, samples of the three variations are shown in Fig. 4.9. Again, no recalculation of parameters is necessary for self-defined spheres. For a semi-defined sphere, its center is located on the axis of its adjacent feature. For a completely-defined sphere, if tangential continuity is present between the sphere and its adjacent feature, the adjacent feature's radius replaces the sphere's original radius. As shown in Fig. 4.9, a sphere can have two adjacent cylinder features and the sphere's radius is to be inherited from both of the cylinders. Conflicting constraint can occur in this situation if the cylinders have different radii.

This is why, during the cylinder idealization step, the radii are to be checked and set equal according to the maximum principal curvature histogram. Conflicting constraints are reduced to redundant constraints. Direct radius inheritance and sphere alignment are possible because the relevant cylindrical features, cylinders and cones, have all been idealized already. Once the sphere parameters are recalculated and the sphere patches idealized, the edges bounding the sphere patches are readily determined.

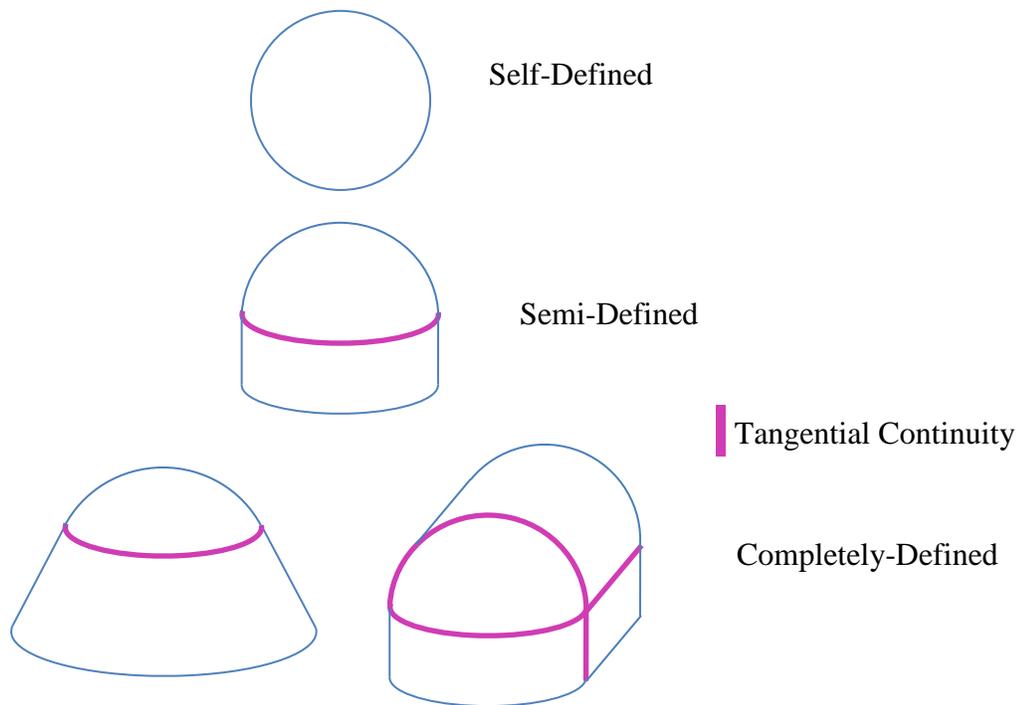


Figure 4.9 Sample sphere variations.

4.3.3.4 Tori

The last feature type to be idealized in the CAD modeling workflow is torus. Examples of the three variations are shown in Fig. 4.10 and again, self-defined tori do not need parameter

recalculation. For a semi-defined torus, the orientation and location are inherited from its adjacent feature. However, the radius parameters are less clear. For the concave semi-defined torus on the left in Fig. 4.10, there is no continuity with its adjacent feature. This suggests that either the minor torus radius or the major torus radius should be inherited from the adjacent feature. In the case shown in the figure, if the difference between the calculated major torus radius and the adjacent feature's radius is within the allowable tolerance, the torus inherits the adjacent feature's radius as its major radius. The allowable tolerance is set as the inverse of the average spread between the maximum principal curvature histogram peaks. For the convex semi-defined torus on the right in Fig. 4.10, tangential continuity exists with the adjacent (lower) cylinder feature such that some of the radius parameters are readily defined. For the completely-defined tori, the adjacent tangentially continuous cylinders completely define the torus parameters, including the minor and major torus radii. Once all the parameters are recalculated, the edges of the torus are then idealized.

It is worth noting here the advantage and importance of global alignment of features in part 1 of the idealization scheme (Section 4.3.2). By globally aligning the cylinder features and thus their axes, torus idealization can take place appropriately. If the cylinders were not aligned and oriented appropriately, torus idealization cannot be completed correctly. Although a solution is guaranteed, it will be incorrect.

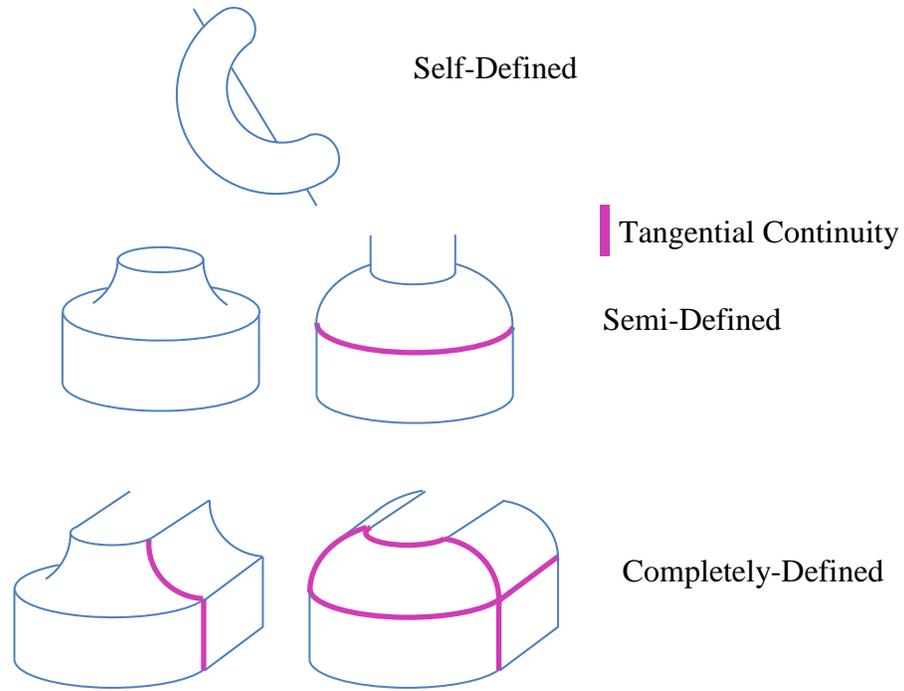


Figure 4.10 Sample torus variations.

Once the idealization reaches the end of the CAD modeling workflow, all the non-idealized edges are in effect idealized numerically. Each edge is moved to its idealized location based on its adjacent features. Over a number of iterations, the adjacent features sequentially and repeatedly impose their type and parameters on the edges. The edge points incrementally approach their ideal locations. It should be noted that the edge points are found within a numerical accuracy specified by the user. The mesh model is thus idealized and no longer a parameter-less model but an idealized parametric mesh model.

Triangle mesh model processed via the proposed idealization algorithm are now topologically equivalent and geometrically similar to the original model or the user's design idea. Accuracy

and precision are imposed and the triangle mesh models can now be considered as 3D engineering design models.

4.4 Case studies

To demonstrate the proposed idealization algorithm, five case studies were carried out utilizing synthesized non-ideal triangle mesh models. The five cases are then followed by examples of real world scan-derived mesh models and unconstrained user manipulated mesh models of varying geometries.

4.4.1 Synthesized non-ideal mesh models: case 1

Fig. 4.11 shows the first synthesized mesh model, which was specifically designed to illustrate the idealization process according to the CAD modeling workflow. This mesh model included three types of prismatic features (planes, cylinders and spheres) connected via either positional (G0) or tangential (G1) continuity within a bounding box of 60×60×120 mm. There are three variations of this model used in this case study. The first model only possessed the chamfered edges. The second variation possessed chamfer edges and noise (synthesized scanning-derived mesh model) and the third variation possessed chamfer edges and surface undulations (synthesized user manipulated mesh model). Essentially, the case study tests the idealization algorithm first on the better quality triangle mesh model variation then on the two worst quality triangle mesh model variations that the flexible hybrid modeler is designed to process.

The first step to process this mesh model, regardless of its variation, was to segment the mesh into patches and identify the feature type for each segmented patch. As shown in Figs. 4.11b,

4.11g and 4.11l, the segmentation proposed in the previous module was capable of automatically segmenting the mesh into appropriate feature patches, with or without noise or surface undulations. Then, the extended mean and Gaussian curvature sign method was used to identify and categorize the segmented feature patches into correct feature types. The planes, cylinders and spheres, respectively colored as light green, yellow and dark green, were all seen to be segmented and identified correctly. Once the features were segmented and identified, the model was oriented to the reference coordinate system. In this case, there were three planes available for the orientation task. The largest plane at the bottom was used to orient the model first and the larger plane on the side was then used to finish the orientation. With the model orientation completed, global alignment of all feature axes was enforced and parameters for each feature were calculated. Part one of the algorithm was thus completed.

With the global model alignment completed, part two of the algorithm, where the idealization was to be done according to the CAD modeling workflow, took place. For this specific model, the cylinders were detected to have tangential continuity with the two side planes. As the cylinders still had one degree of freedom to move their locations parallel to the two side planes, the cylinders were considered as semi-defined. The cylinders' common radius was, however, determined directly from the two planes and the cylinders' locations perpendicular to the planes were also constrained. The spheres, being tangentially continuous to the three adjacent cylinders, inherited their common radius and their centers. In other words, the sphere was considered as completely-defined. The idealized mesh models from the synthesized non-ideal mesh models with their feature types and edges (red) are shown in Figs. 4.11c, 4.11h and 4.11m.

Figures 4.11d, 4.11i and 4.11n show the idealized models without feature edges to visually illustrate the quality of the idealized models.

Figures 4.11e, 4.11j and 4.11o quantitatively compare the idealized models to the original. For the chamfer-edges only case, the resulting deviation is negligible and the radii of the cylinders and spheres are all 30 mm, the exact value used to create the model. This demonstrates the effectiveness of the presented method in accurately extracting the feature parameters when a mesh model only has imperfect edges. As for the scanning-derived case, the maximum deviation is found to be 0.229 mm. The radii are all 29.990 mm, very close to those of the original even though substantial amount of noise is present on the mesh model (the imposed noise was about 20 times higher than the actual scanning noise). This can be attributed to the per-patch Laplacian smoothing applied to the feature patches before extracting the feature parameters. For the synthesized unconstrained user-manipulated mesh model case, the maximum deviation is 0.2192 mm. The radii for this case are 29.94 mm, again, very close to those of the original. The algorithm is able to deal with the larger surface flaw (surface undulation) well. The automatically idealized model is thus geometrically similar and topologically identical to the original.

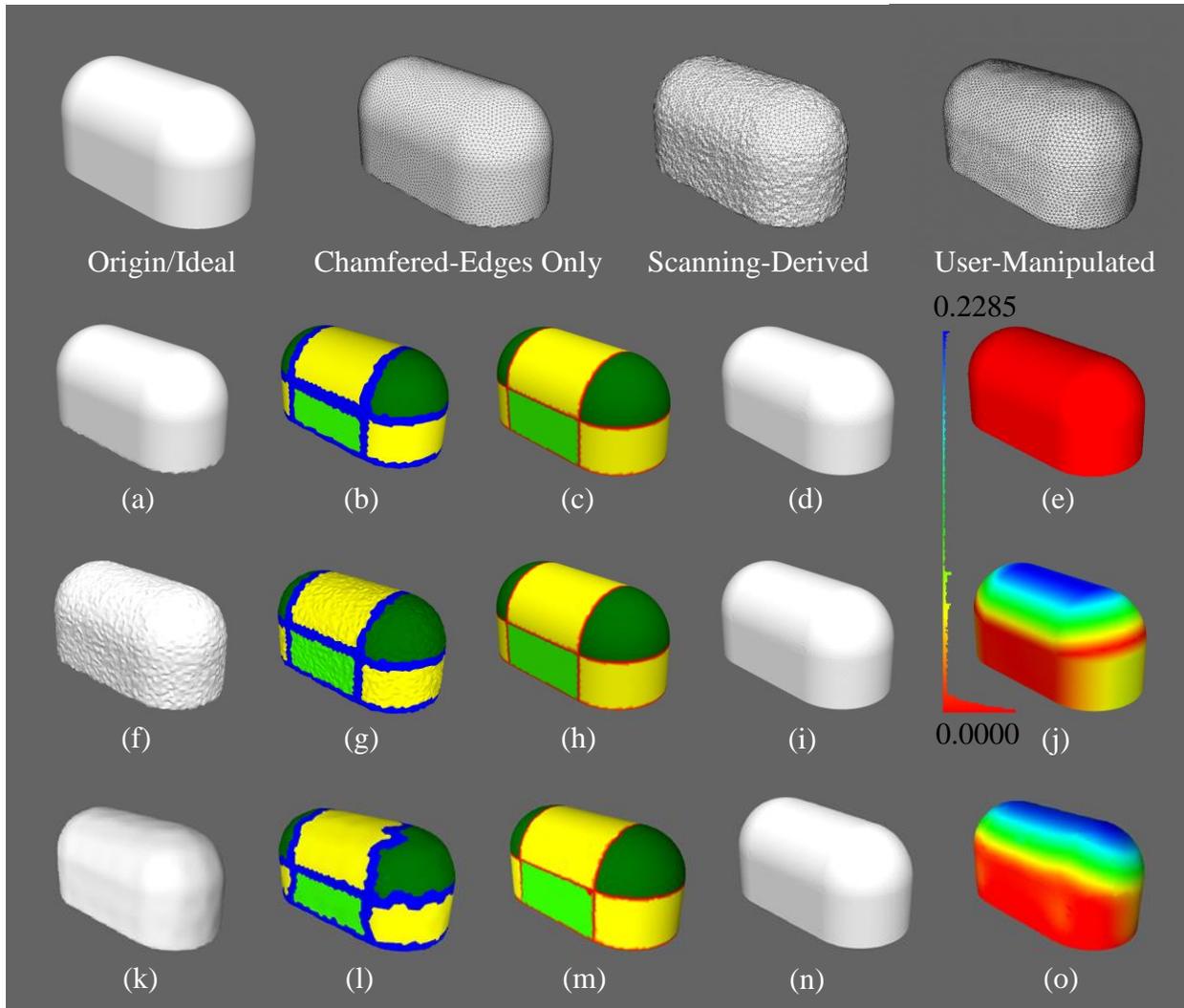


Figure 4.11 Case study 1: synthesized non-ideal mesh models with normal geometric complexity: (a), (f), (k) synthesized non-ideal meshes; (b), (g), (l) segmented non-ideal meshes; (c), (h), (m) idealized meshes; (e), (j), (o) deviation map of idealized meshes.

4.4.2 Synthesized non-ideal mesh models: case 2

The second case study, as shown in Fig. 4.12, involves a 100x100x100 mm model with increased geometric complexity. Even though the model geometry is seemingly simple, complex continuity conditions exist for the corner blend. The cylinders in this case are no longer bounded by the same set of parallel planes, which means that their extracted radii and cylinder centers as

projected on the sphere corner can be different. If the radii of the three cylinders or the cylinder centers projected onto the sphere are not exactly the same, idealization of the model is possible but will be incorrect. The end result is that the sphere corner will be geometrically and topologically idealized with respect to only one of the cylinders but not all of the three cylinders as originally intended. Such complex constraints are always difficult to deal with; however, the proposed idealization scheme based on the CAD modeling workflow is able to perform such complex idealization correctly and automatically.

The algorithm, again, first segmented and identified the features on the featureless mesh (Figs. 4.12b, 4.12g and 4.12l). The features were then oriented and globally aligned. As the idealization grew from planes to cylinders, a check against the maximum principal curvature histogram was done for all the cylinders in order to ensure that cylinders that should be considered of the same size have equal radii. The centers of these cylinders were then shifted via Eq. (2). By doing so, the cylinders all had the same radii and projected-to-sphere centers. The mesh model could then be idealized appropriately and the completely-defined sphere's parameters be automatically determined. Conflicting constraints were avoided and reduced to redundant constraints. The sphere was idealized by inheriting its adjacent cylinders' common radius and projected centers. As shown by Figs. 4.12c, 4.12h and 4.12m, the features and their edges are all properly identified and idealized. Figs. 4.12d, 4.12i and 4.12n show the rendered images of the idealized models. Again, for chamfer-edges only case, the original model was recovered identically. For scanning-derived case, small deviations existed for the cylinders and propagated to the sphere. A maximum deviation of 0.583 mm was found on the sphere. For the unconstrained user manipulated mesh model case, comparable deviations occur for the cylinders

and sphere. A maximum deviation of 0.578 mm was found on the sphere. This case study demonstrates that even with much increased continuity complexity, the idealized models for the two worst input variations are still geometrically similar and topologically identical to the original. Furthermore, the goal to translate both scanning-derived and unconstrained user manipulated meshes into 3D engineering design models is achieved on a synthesized level.

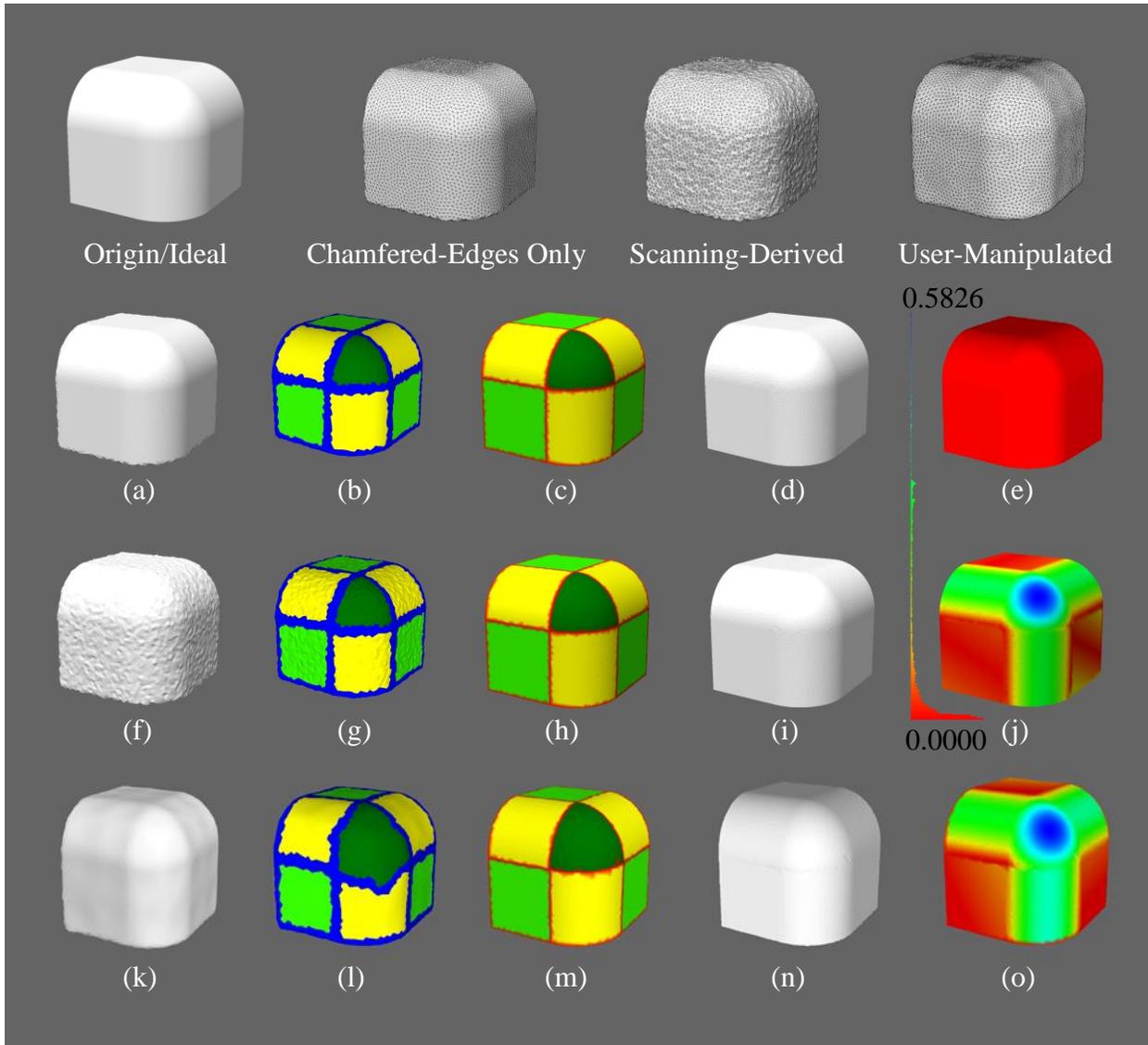


Figure 4.12 Case study 2: synthesized non-ideal mesh models with increased geometric complexity: (a), (f), (k) synthesized non-ideal meshes; (b), (g), (l) segmented non-ideal meshes; (c), (h), (m) idealized meshes; (e), (j), (o) deviation map of idealized meshes.

4.4.3 Synthesized non-ideal mesh models: case 3 – 5

Three more synthesized models were processed with the proposed method following the CAD modeling workflow with consistently good results. The mesh models were idealized and the resulting idealized models were all similar and topologically equal to their respective original models. Fig. 4.13 shows the three synthesized models as well as their corresponding idealized models. For the first model, a maximum error of 0.130 mm was found for the scanning-derived case and a maximum error of 0.1288 mm was found for the user manipulated case. For the second model, a maximum error of 0.5046 mm was found for the scanning-derived case and a maximum error of 0.5121 mm was found for the user manipulated case. And for the third model a maximum error of 0.7461 mm was found for the scanning-derived case and a maximum error of 0.7319 mm is found for the user manipulated case. Considering the chamfered-edge effects, the high level of noise and the large surface undulations, good idealization results were obtained. The mesh models are automatically idealized such that the geometry is similar and the topology is equivalent to that of the original models for all cases.

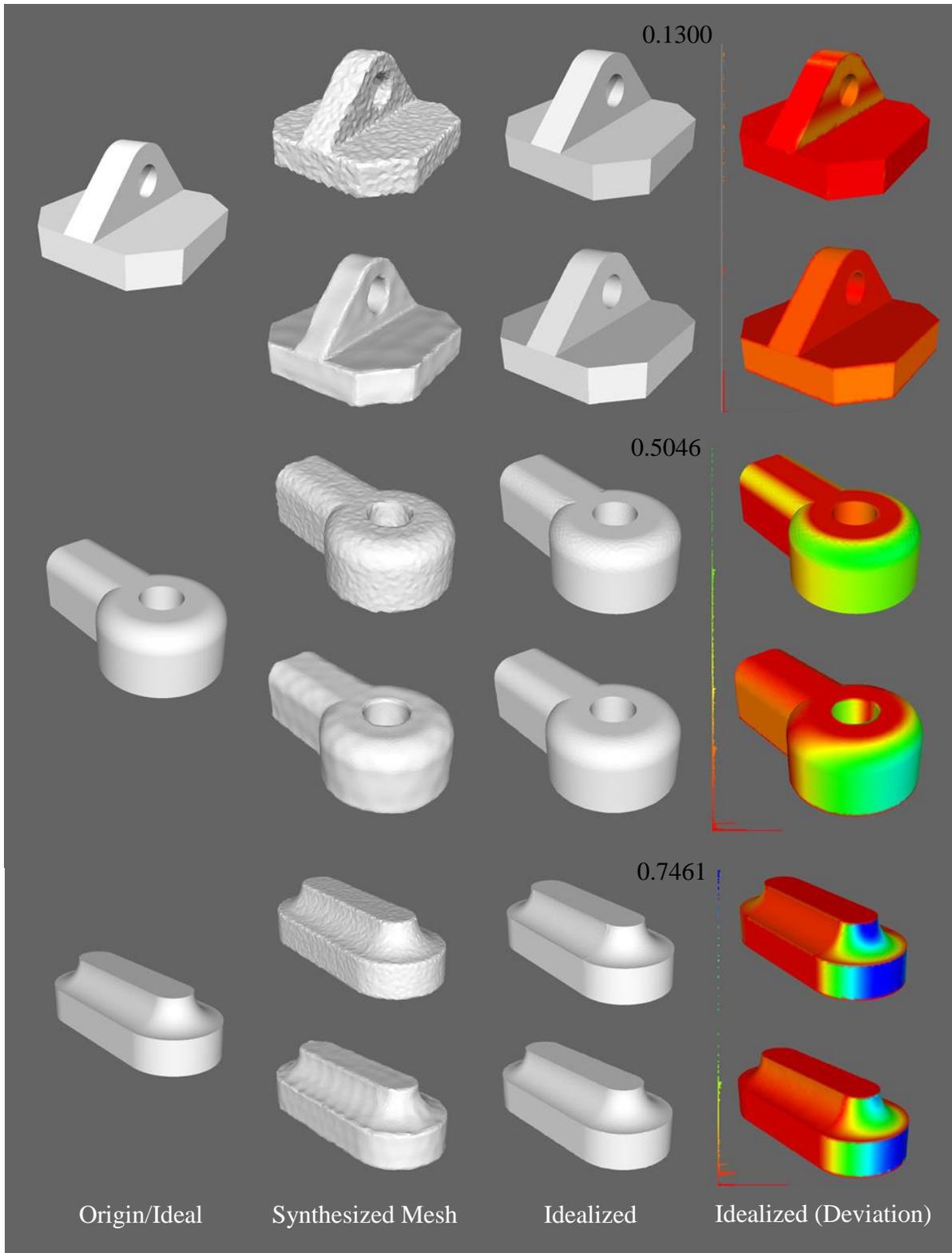


Figure 4.13 Additional synthesized non-ideal mesh models.

4.4.4 Scanning-derived and user manipulated mesh models

With the proposed algorithm tested and proven on synthesized mesh models, the next logical step is to validate it on actual scanning-derived mesh models and real user manipulated mesh models. To ease readability, this section is broken down into the above two parts, respectively. Real life scanning-derived mesh model cases are outlined first with real user manipulated cases outlined second.

In order to have an ideal reference model to compare with, each object to be scanned was first constructed by a commercial CAD software package and then 3D printed by a Stratasys uPrint SE Plus 3D printer. The 3D printed object was scanned by a high-precision LDI Surveyor WS-3040 3D laser scanning system. The scanned point cloud data was used to construct a watertight manifold mesh model which was then processed by the proposed idealization scheme. Fig. 4.14 shows these scanning-derived mesh models along with their idealized models and deviation maps. As can be seen in the figure, the idealized mesh models are all geometrically close and topologically equal to the original reference CAD models. The feature types and parameters are all correctly identified. It is thus evident that with the proposed algorithm, real-life scanning-derived mesh models can be reliably and automatically idealized into parameterized mesh models.

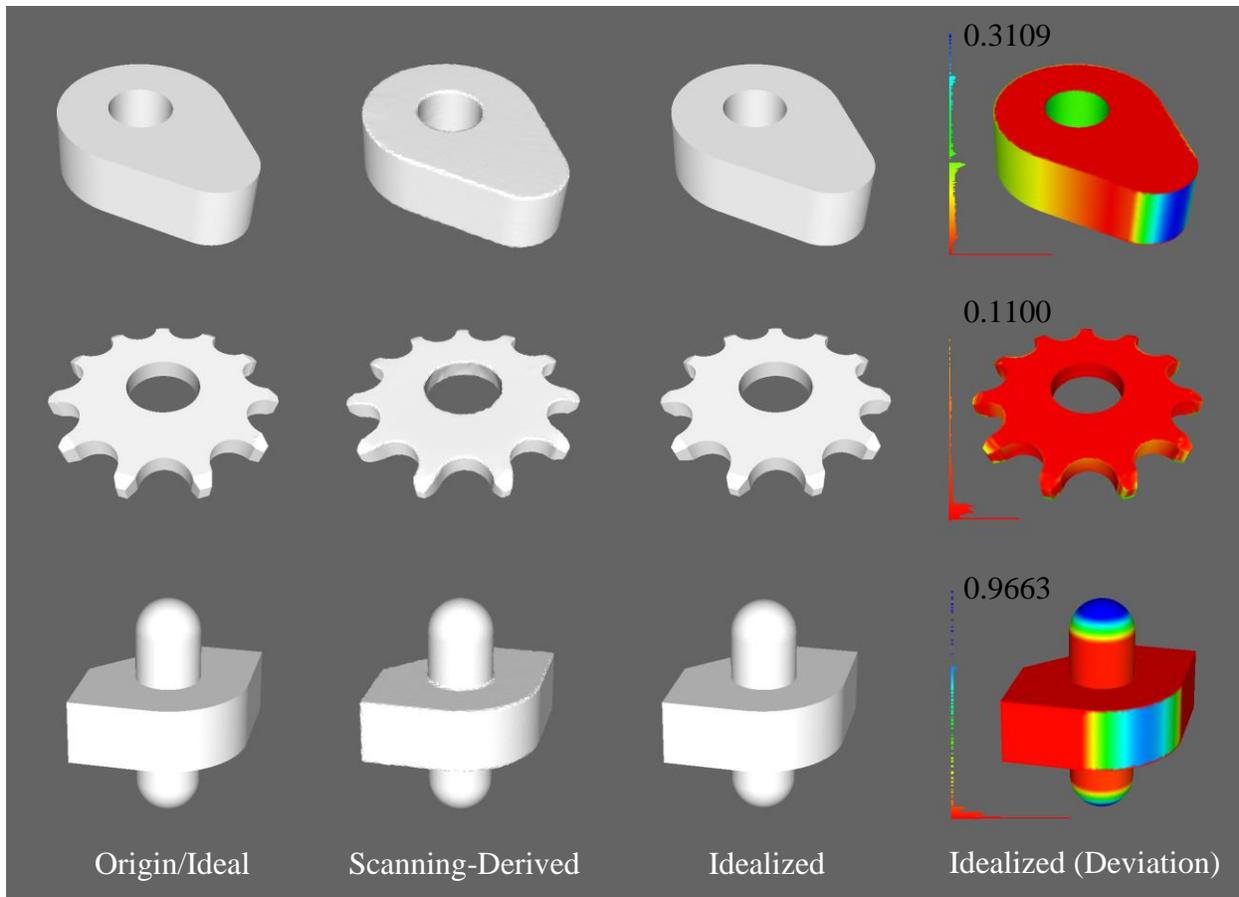


Figure 4.14 Scanning-derived mesh models.

For real unconstrained user manipulated mesh model cases, three models (Fig. 4.15) were derived through real user mesh manipulation utilizing Sculpttris, the chosen mesh sculpting software. Each of the three models were sculpted on a base geometry with the ideal target geometry shown to the user. The user freely manipulated the base geometry until the design reflected that of the target. The manipulated mesh model is then processed directly by the proposed algorithms. The right portion of Figure 4.15 shows the results of the proposed algorithm on these three user sculpted meshes. As seen in the figure, all intended features on the user manipulated mesh are correctly idealized by the proposed approach. The relationships

(continuity conditions) between the features are also correctly idealized. With the proposed work, the original intended design is extracted and idealized from the user manipulated mesh model. This enable the hybrid modeling scheme to transform freely user manipulated meshes into 3D engineering design models. The precision and accuracy of 3D design models can thus be readily integrated with the flexibility of mesh modeling.

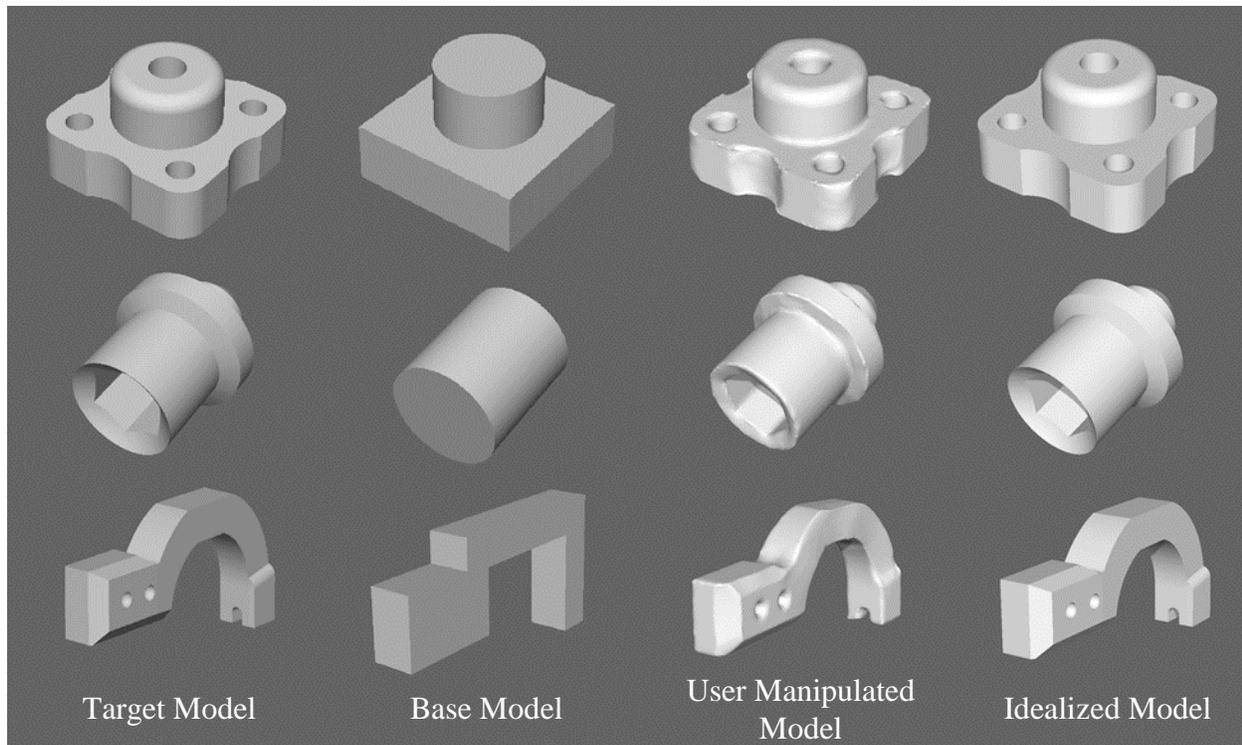


Figure 4.15 User manipulated mesh model.

4.5 Summary

This module aims to idealize all inputs, including user manipulated triangle mesh models, in order to best realize the flexible hybrid modeler. For triangle mesh models to become 3D engineering design models that serves equal function as the current day parametric CAD models, the triangle mesh model has to not only encompass feature information but also represent the

ideal intended design geometry. 3D design models are considered as the ideal models in engineering; therefore, the resulting triangle mesh model derived through the proposed flexible hybrid modeling scheme needs to geometrically reflect this “perfection”. Consequently, a scheme that directly idealizes mesh models based on CAD modeling workflow is proposed in this module in order impose this perfection (within numerical accuracy). Mesh inputs from various sources such as scanning-derived mesh models and user manipulated mesh models/regions are idealized to the ideal design geometry. Triangles meshes (of prismatic geometry) become ideal engineering design models that carries critical feature information.

The case studies outlined in this section validates that the above goal is achieved through the proposed work. The proposed idealization method automatically idealize the two most difficult to process mesh variations into idealized mesh models. The parts are processed such that the end meshes are geometrically similar and topologically equivalent to the original parametric design. One important advantage of the proposed approach that should be outlined is that even if idealization does not result in the intended geometry due to overly large surface undulations or mesh noise, a partially correct geometry will result. This partly correct geometry can be manipulated and reprocessed to reach the correct result. With a solver-based approach, a partially correct geometry usually does not result. Nevertheless, there are still aspects of the proposed idealization scheme that can be further improved such as the detection of feature symmetry and feature patterns. Even though some symmetric feature cases are already implicitly satisfied via the proposed scheme, explicit detection and imposition of the geometric constraints due to symmetry and pattern in the feature configuration of a model will further improve the idealization result.

Overall the idealization method reliably and automatically idealizes mesh models with prismatic features into ideal parameterized engineering design (mesh) models. The precision and accuracy of parametric CAD models are achieved in the flexible triangle mesh model domain. With this idealization module along with the segmentation technique, flexible modeling for prismatic engineering parts can be achieved without the loss of accuracy and precision. The flexibility in terms of geometric manipulation resulting from the use of triangle mesh representation can be fully exploited in an engineering sense. Higher level of flexibility in modeling can finally be realized in the engineering design world.

Chapter 5: Unconstrained Feature-Free Model Modification

With the prismatic triangle mesh model segmented, identified and idealized, feature-based mesh model is realized. The mesh model now possesses feature information that is comparable to parametric CAD models. The mesh is no longer featureless, but contains parametric information that facilitates downstream design tasks. Moreover, the feature information is imposed as an associated information layer independent of the model representation. This enables the utilization of unconstrained feature-free modeling methods without the loss of the notion of features. Users can now alter the geometry of the prismatic model through direct unrestricted interaction with the model, which is not attainable via current parametric CAD systems. An equivalence can be drawn with clay modeling where the modeling clay is interactively deformed to achieve the intended design. Such deformation capabilities relax the rigidity of CAD modeling.

5.1 Prime candidate

As mentioned in the literature review chapter of the thesis, due to its capabilities and unambiguity in shape alteration [97], mesh sculpting is the prime candidate for unconstrained feature-free model modification. With its potential, mesh sculpting is a popular research topic with many powerful algorithms proposed tackling mesh sculpting [37,94,95]. Many robust implementations are already commercially available in well-developed software packages. Thus, to robustly enable mesh sculpting for this work, it is suitable to employ these already well developed mesh sculpting software packages. These packages are more than sufficient in terms

of facilitating the needed level of shape alteration required by the proposed flexible hybrid modeler.

Software packages such as ZBrush [116], MudBox [117], Cinema 4D [118], Modo [119], 3D Coat [120], Blender [30], and Geomagic Sculpt [121], are just some of these said mesh sculpting based/capable packages. These packages offer the users with very extensive featureless modeling environments that go beyond mesh sculpting itself. Modeling approaches, such as those with featureless “pseudo features” are all part of these modelers but, as previously mentioned (Section 2.3), are of no benefit to the hybrid modeler. Given the goal and scope of this work, a light weight sculpting only software package is all that is required for prototyping purposes.

One such light weight software is Sculptris, a mesh sculpting tool by Pixologic [110]. Sculptris is a simple triangle mesh shape alteration tool that not only freely deforms mesh models but also adaptively mesh the newly deformed areas in order to preserve the triangle quality. Sculptris ensures that improper triangles (skinny, inverted and self-intersecting) do not form during large model deformation operations. Triangles are automatically divided, merged and flipped as the modeler see fit. Large scale shape manipulation can take place without detrimental effect on the mesh model. With the use of Sculptris, users can directly and robustly deform any input mesh models. Various operations such as flattening or smoothing of the mesh and creasing or pinching of the mesh are available in Sculptris. These operators in Sculptris allow the mimicking of physical sculpting in the digital realm. With Sculptris, the author can efficiently perform the case studies on the proposed hybrid modeler.

5.2 Summary

Overall, Sculptris seemingly contains the inner workings to enable flexible unconstrained mesh manipulation. It is simple, adequately powerful and does not contain unnecessary functions (with respect to this work). This makes it the prime shape manipulator candidate for this work. Furthermore, Sculptris is well-developed and implemented and freely available. Hence, Sculptris is chosen as part of the proposed flexible hybrid modeler.

With Sculptris, the user is able to straightforwardly interact with the mesh models in order to bring designs to life. The shape of geometric models is quickly changed without the user dealing complicated menus or software constraints. Rough mockups can be effectively modeled as shown and illustrated in Fig. 4.15 and repeated here in Fig. 5.1 (with sculpted models only). With the segmentation module and the idealization module, these rough design mockups are autonomously transformed into ideal prismatic engineering mesh models (Fig. 4.15). However, the exact workings behind Sculptris is proprietary and thus, not known to the author of this thesis. Through comparing functionalities, it is found that the publication, Freestyle [37], is very similar to Sculptris. They both perform powerful shape alteration with build in adaptive meshing technique. For a more in-depth understand of digital sculpting, please refer to [37]. For hands on experience, please refer to Sculptris.

With feature segmentation, mesh idealization and unconstrained feature-free Model modification, the proposed work isolates some of laborious feature-by-feature modeling work from the users and enables an flexible direct geometry manipulation scheme that derives geometric models with similar geometric information and functionality as those laboriously

modeled through the current parametric CAD software. Engineering design models are realized without the rigidity caused by features as seen in the current CAD systems. The user can simply deform the mesh model close to the desired shape. The hybrid modeler completes the other tedious but necessary tasks to fully derive the intended engineering design model.

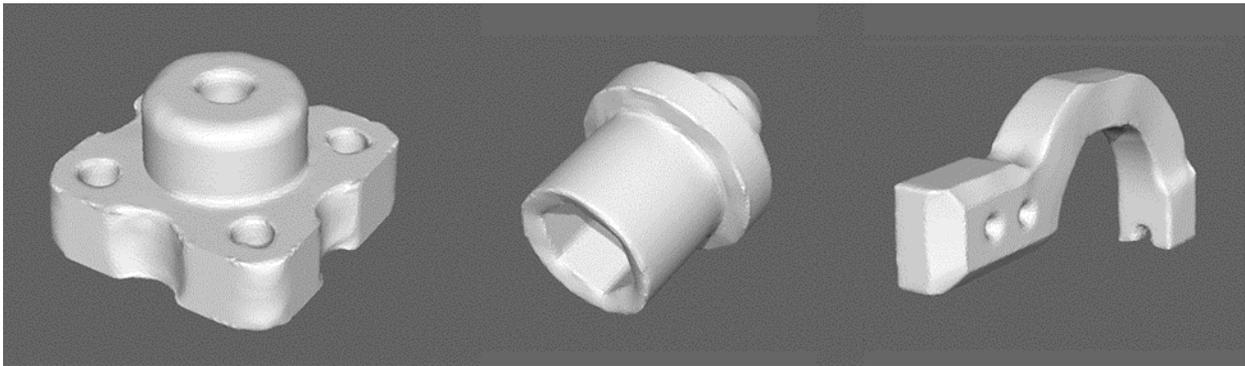


Figure 5.1 Unconstrained mesh manipulation for engineering design applications.

Chapter 6: Feature-Based Model Editing

With the first three components of the proposed work, users are able to realize their designs in a 3D digital environment less restrictively. The flexible hybrid modeler provides users with the ability to freely manipulate the geometry of models without losing the features and therefore, with losing the model precision and accuracy. A set of background algorithms transparently idealizes the users' freehand manipulations into prismatic mechanical features and models. Designs along with their (design) intent are realized in a straightforward manner with the geometric precision and accuracy retained. Users do not have to decompose designs into sets of operations which the CAD modeler can recognize. Users have the choice to "pick up" an object in a 3D environment and deform it as they see fit, which is not available in current CAD modelers. This capability gives users the ability to "play" with the model geometry without the need to wonder "how do I create this design/impose this change in this particular CAD software". Objects can be deformed into the desired designs without involved operations with cumbersome menus and dialogs. Models can be altered without the deletion and remodeling of features. The flexible hybrid modeler provides such increase in modeling freedom to the users.

Nevertheless, feature-based model editing is still very useful and necessary in 3D engineering design modeling. In cases where parameter changes are required after the initial mockup, feature-based model editing approach (much like direct modeling) is easier and faster. A simple "push-and-pull" operation with parameter feedback allows the users to very quickly change the size and/or location parameters of any features. For example, a hole or a boss can be very

straightforwardly moved to a different location on the model through simple “grabbing and pulling” operators. The size of the hole or the boss can be quickly changed by just “stretching the hole larger.” In these type of cases, it is simpler to utilize feature-based modeling with direct model editing operators instead of unconstrained feature-free model manipulation operators.

Having the ability to directly manipulate the geometry of the design model has its advantages in term of increasing modeling flexibility; however, as outlined, there are instances where feature-based model modifications are simpler. Unconstrained model modification allows for unrestrictive feature-free geometric alterations while feature-based model editing give rise to effective parameter adjustments/changes. Incorporating feature-based model editing to the unconstrained model manipulation portion of this work combines the advantages of both types of modeling techniques, arriving at a truly flexible and therefore, hybrid modeling scheme. Thus, feature-based model editing for mesh models is proposed in this thesis to further increase the hybrid modeler’s modeling flexibility and capabilities.

To fully realize robust feature-based model editing with structure-altering capability, a feature-mapping mesh editing approach is proposed in this module. The said technique treats the previous extracted model features (type and parameters) and model feature dependencies (continuity conditions) information as a feature information layer, with the triangle mesh taking on the shape conveyed by the feature information layer in order for the mesh to geometrically represent the desired model. As initially mentioned in this thesis and reminded here, the features are treated as information that is not rigidly attached to the triangle mesh in order to promote increased modeling flexibility. For feature-based model editing, this information layer

effectively acts as the constraint layer. This constraint layer governs the behavior of the associated triangle mesh in order to give rise to feature-based model editing. With the existence of this feature modeling information layer and its association with the triangle mesh model, feature-based modeling is readily achievable. Furthermore, with the notion of feature imposed as a separate information layer, large movement alterations and flexible structural changes are trivial. Structure-altering edits simply implies a change in the feature modeling information layer which propagates to the associated triangle mesh. With this approach, cutting and filling is avoided for structure-altering edits (only required for genus changes/“number of through holes” changes) since the mesh at any location can readily adapt to its new feature layer information. Feature-based editing, structural or non-structural, required to finalize common engineering design models is thus readily achievable. Figure 6.1 shows the flow chart of the proposed method.

To best describe the proposed feature-based mesh editing method, this thesis breaks the method down into two major sections. The first portion outlines feature editing that does not result in structural alteration and the second portion outlines feature editing that involves structural alterations. Please note that the two sections are not completely separate entities. The structure-altering portion of the algorithm builds on top of the non-structure-altering portion. The breakdown is only meant to increase the readability of the work.

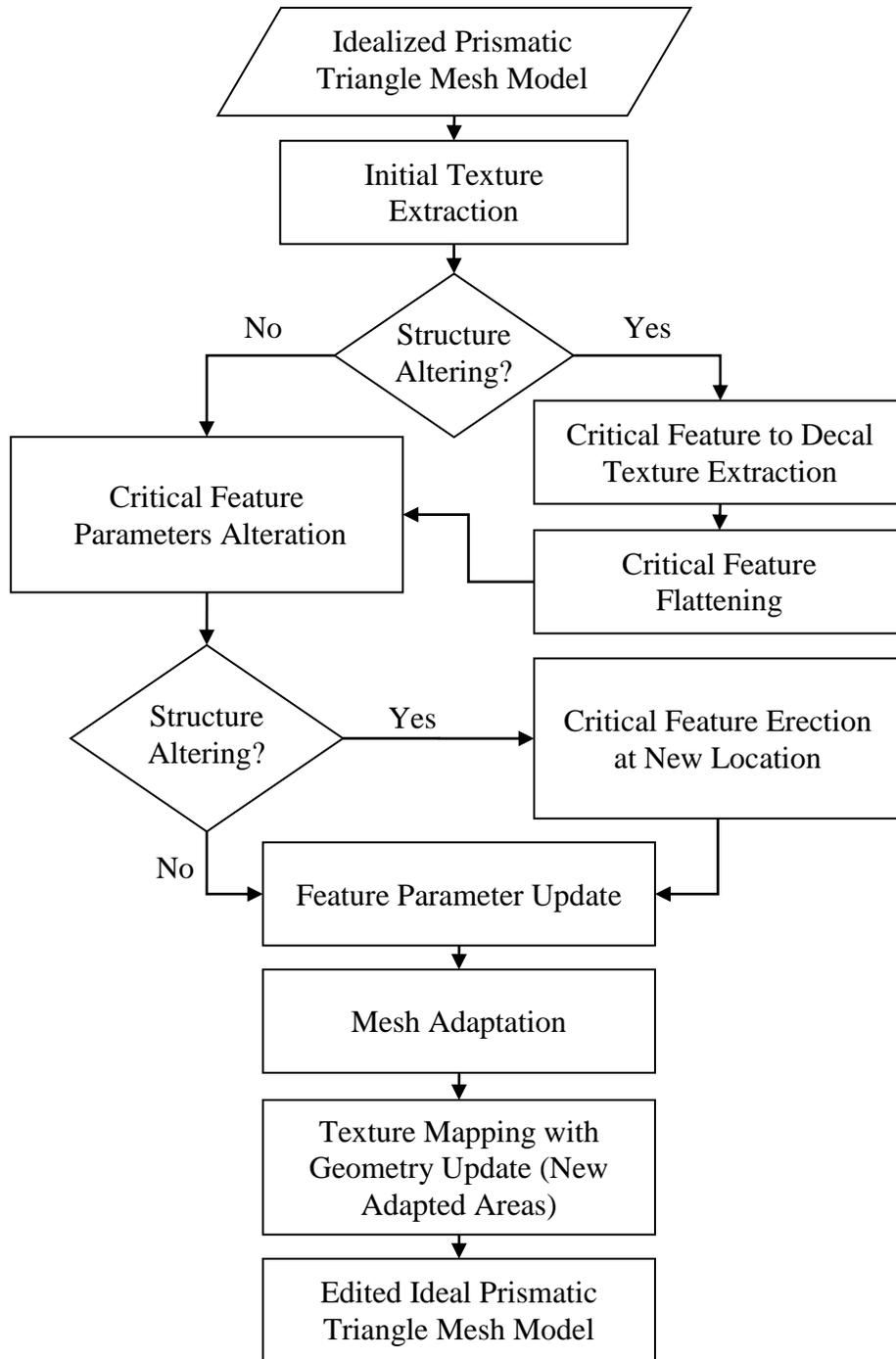


Figure 6.1 Proposed feature-based mesh editing method flow chart.

6.1 Non-structure-altering edits

Previously, it was mentioned that the goal of this module is to enable feature-based model editing for engineering triangle mesh models. Feature-based editing refers to constrained model editing where the deformation and movement of any portion of the model is controlled by geometric constraints. In other words, a plane stays as a plane, a cylinder stays as a cylinder and a tangentially continuous edge stay tangentially continuous. The typical approach to solve any constrained geometric problem is the utilization of a 3D constraint solvers. However, as mentioned in the idealization module, the usage of constraint solvers requires careful selection of geometric constraints. The existence of even redundant constraints could render these solvers unable to arrive at an appropriate solution. Intensive user interactions are required at the initial stage as well as any stage where the geometric constraints need to be specified or changed, making the model editing process very tedious. Automatic selection of geometric constraints which guarantee solvability of 3D constraint problems is still an open problem due to the non-unique nature of the solution. Any one constraint is correct locally on its own, but as a group globally, it can cause the formulated problem to be unsolvable.

6.1.1 Feature parameter update

For this work, the developed idealization algorithm (Section 4) is adapted to facilitate feature-based model edits. The idealization method is used to solve for the geometric parameters of the affected features as the result of the edits made on certain features on the model. These features are referred to either as critical features or feature-of-interest in this thesis. For a given model, when a critical feature is pushed or pulled, its immediate neighboring features (1st-ring) are

usually affected. The immediate neighbors' feature parameters need to be updated in order to maintain proper inter-feature relationship. Some non-immediate neighbors (2nd -ring and above) are also affected in the editing process, depending on the continuity conditions. These feature's feature parameters also need to be updated. Thus, the idealization algorithm is utilized in this module with the starting seed set as the critical feature. CAD Modeling Workflow is no longer followed for feature-based model editing. Instead the idealization algorithm propagates through the affected neighboring features from the immediate affected neighbor (1st-ring) to the "last" affected neighbor (nth-ring). For each ring, positional continuity constraints are propagated first with tangential continuity constraints propagated second.

Positional continuity constraints are propagated first for each ring because it has been found that for prismatic engineering models with orthogonal features (features aligned with the axial planes of the global Cartesian coordinate system) at the location of tangential continuity, tangential continuity constraints are implicitly satisfied if positional continuity constraints are satisfied with feature types preserved during editing. Figure 6.2a shows an example of the above point. The plane on the right of the model is moved outwards in this example. The adjacent cylinder features, which are immediately affected by the movement of the plane, need to increase their feature parameters in order to maintain positional continuity with the plane. The sphere, being affected by the cylinders, is enlarged to maintain positional continuity as well. As illustrated, the resulting edited model automatically satisfies the tangential continuity constraints between features without the need to explicitly solve for the tangential continuity conditions. Figure 6.2b, shows another example where the features present are orthogonal. Again, tangential continuity constraints are implicitly satisfied through satisfying positional continuity constraints.

Thus, by propagating positional continuity locally for orthogonal features, tangential continuity conditions are implicitly solved.

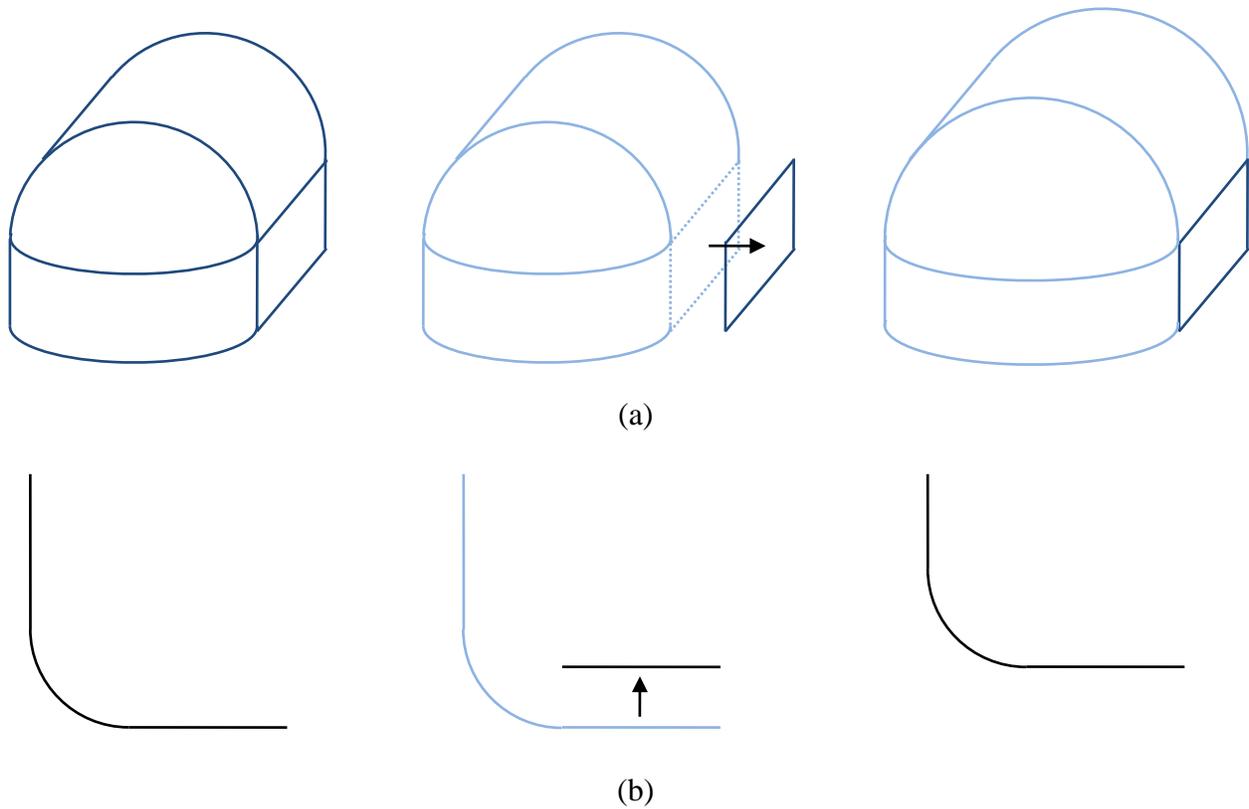


Figure 6.2 Tangential continuity fulfillment via positional continuity satisfaction given feature orthogonality.

However when features are not orthogonal, tangential continuity constraints are not implicitly satisfied. Even so, for practical prismatic engineering models, the parameters of features necessary to ensure tangential continuity are solved in a straightforward way. For example, given the geometry shown in Fig. 6.3, an edit which moves the cylindrical feature upwards, results in the elongation and change of angles of the two adjacent planes. Since these features are not orthogonal, the tangential continuity conditions are not implicitly satisfied through positional continuity. Nevertheless, the size and location of the cylinder can be calculated from

the planes such that the tangential continuity conditions are satisfied (as illustrated in the figure). The opposite is also true if the cylinder is the critical feature. The plane parameters can be calculated given the cylinder parameters. If there are other tangentially continuous non-orthogonally positioned features in the 2nd-ring, their parameters are calculated based on the 1st-ring features such that all continuities conditions in all rings, linking to the critical feature, are satisfied. As stated before, the continuity conditions are propagated from the critical feature to its affected neighboring features. If the user does chose to break any of the tangential continuity conditions on the model, only the positional continuities are propagated. After the non-structure-altering edit solution is found, the triangle mesh model is updated (feature patches and feature intersections outlined in Section 4.3.2 and Section 4.3.3) accordingly to reflect the edited model. The geometry is correctly updated by the direct local usage of continuity conditions.

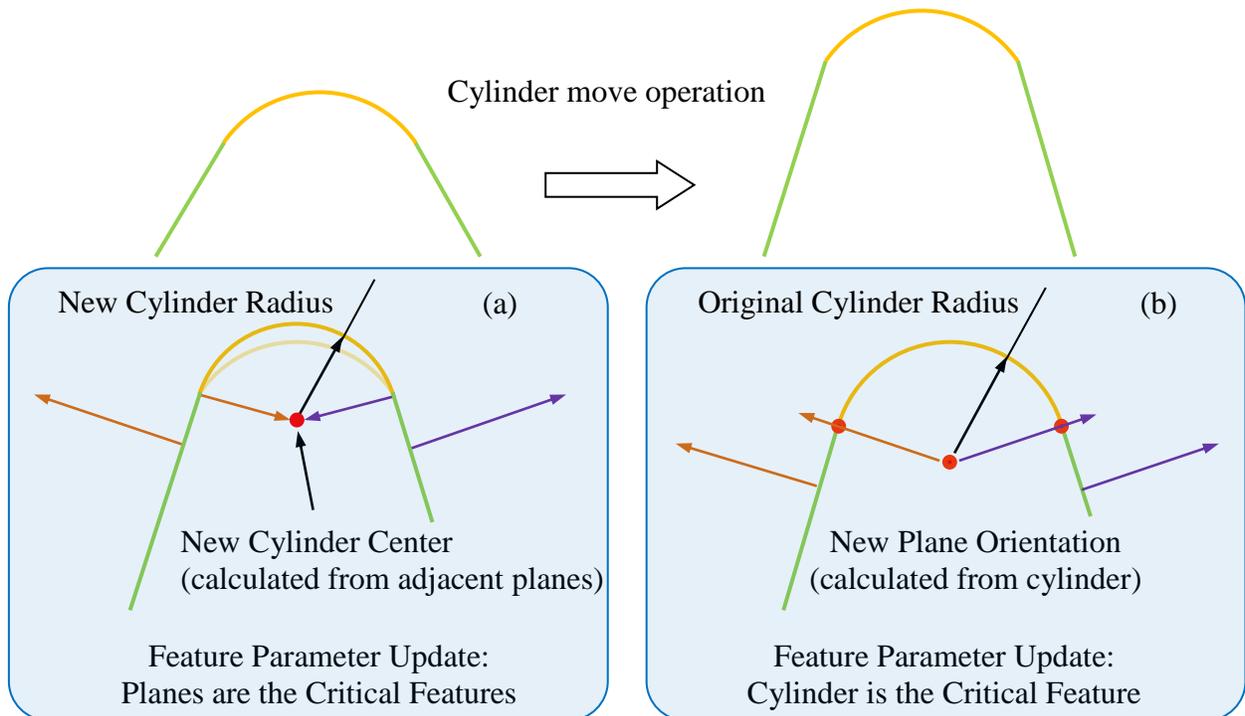


Figure 6.3 Direct parameter calculations for maintaining tangential continuities (example): (a) planes are the critical feature; (b) cylinder is the critical feature.

6.1.2 Adaptive meshing

However, non-structure-altering edits can potentially involve very large feature movements. These large feature movements often result in the formation of bad mesh triangles (skinny, inverted, self-intersecting) which is detrimental to the quality of the triangle mesh model. Even though the mesh points are accurate representations of its feature information layer after an edit update, the mesh quality can become very poor. This is an inherent issue for mesh editing since the input model is defined by a preset number of triangles. Without splitting or merging, the number of triangles does not change over the course of the edit. With large movements, mesh triangles are overly strained even with the redistribution of mesh triangle points with respect to features. In order to prevent this issue, adaptive meshing technique [37] is used in this work. Adaptive meshing allows mesh to evolve as deformation occurs to the model. If triangles are becoming too thin, splitting or merging or flipping operations are used to avoid the formation of bad triangles (Fig. 6.4). The quality, often measured as the size and aspect ratio of the triangles are kept relatively constant. As the model is edited, the quality of the mesh is minimally affected.

Due to the proposed work treating geometric information as an additional information layer, the adaptive meshing technique is integrated into the method seamlessly. The mesh is able to evolve and take on feature information as required. As the mesh adapts, the feature information is imposed on the newly adapted sections. The adapted triangles take on the feature type and parameters which bounds the new triangles. Once the adapted mesh takes on the edited feature information, the mesh model becomes an accurate representation of the intended edited model

with good quality mesh triangles. By imposing features as an information layer and incorporating the proposed geometric re-calculation method with adaptive meshing, non-structure-altering edits of large scale is achieved. The edited mesh model quality is minimally changed compared to the original input mesh model quality. If required the mesh can be accurately adapted to increase or decrease mesh quality (triangle size). The quality of the model is easily changed while retaining an accurate representation of the intended model.

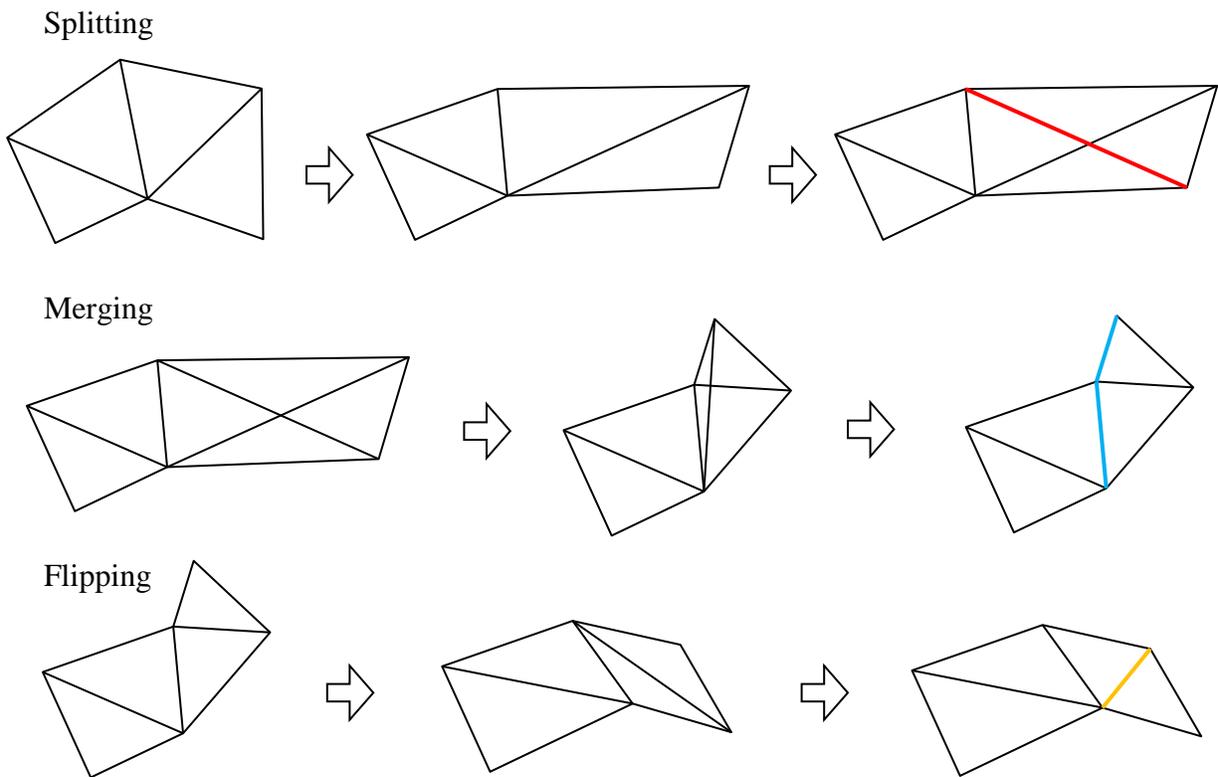


Figure 6.4 Adaptive meshing

6.2 Structure-altering edits

With large-scale capable non-structure-altering feature-based mesh editing realized, structure-altering editing is tackled next in order to fully realize feature-based model editing for prismatic

triangle mesh models. Only with the feature-based modeling editing fully realized, can the flexible hybrid modeling scheme be brought to life. The unconstrained feature-free modeling portion is already developed to increase flexibility in CAD modeling. The incorporation of feature-based model editing is needed to complete the modeling scheme.

6.2.1 Information layer mapping

To outline the proposed scheme for structure-altering editing, the mapping concept is introduced and elaborated in this section. Mapping, more specifically UV mapping is a topic that has been rigorously worked on in the computer-graphics community, largely for application of texture to meshes. UV mapping is a process where 2D texture map (or maps) is projected on to a 3D object or surface. The three typical steps that are involved in UV mapping is unwrapping, texture generation and texture application. Unwrapping, also known as parameterization [122] is the breakdown of a 3D surface into planar surface or surfaces in order to enable mapping of 2D texture maps. Please note the difference in definition for parameterization between the computer-graphics community and the CAD community. Texture generation, the second step, is where the desired texture is created with respect to the parameterization. And texture application, the third and final step, is the utilization of the created texture on the parameterized model.

The three steps of UV mapping describe the proposed feature-based mesh model editing method precisely if the preliminary feature segmentation and identification steps are together treated as a form of model texturing. The initial segmentation which prepared the triangle mesh model for the proposed editing algorithm essentially breaks the mesh down into patches that are ready for

texturing. The texture, in this case as simple colors, is applied to the mesh vertices based on the identified geometric types (plane: light green, cylinder: yellow, cone: blue, sphere: dark green, torus: dark red, edge: red). Therefore, the feature type information can be viewed as a colored texture map imposed on the mesh. Conversely, the mesh vertices can be viewed as pixels of the colored texture map. This relationship creates an association of feature type via colored texture map to the mesh vertices. The feature parameters are also associated with the mesh model via texture map in order to complete the feature information layer. When the model is edited, the feature information layer is updated in accordance with the deformation. The texture map is then used to impose the appropriate feature information on the altered mesh vertices.

For structure-altering edits, it becomes similar to texture editing exercises. To move a feature to a new location, the texture which represents the critical feature is selected and extracted from the rest of the texture of the model. The associated feature parameters are also extracted with the texture. To increase the interactive-ness of the edit, the extracted texture becomes a decal texture. The decal texture is mapped to any location on the base/original texture. The movement of the extracted decal texture effectively represents the movement of the critical feature.

6.2.2 Mesh-based feature editing

Thus, one of the first steps for structure-altering edits is the extraction and transformation of the critical feature into a decal texture. When a specific feature is selected for editing, the complete feature type defined by its color is selected through region growing within the feature borders. The region growing algorithm selects vertices within the feature borders until every vertex of the area is selected. The selection then extends to include the feature border points. Since these

border points defines the current edges, which results from the interaction of the critical feature and the rest of the model, they will not exist in its current state once the critical feature is moved elsewhere. Therefore, they need to be selected and processed.

With the feature selection completed, parameterization (computer-graphics definition) is necessary in order to find a planar (decal) projection of the critical feature. Since the goal is to acquire one single decal texture from the given selection, [123] is used to parameterize the selection. The cited work is a spherical parameterization method that parameterizes close meshes through dividing the meshes in half and projecting each halves onto hemispheres. The hemispheres are combined to form the spherical parameterization. For the proposed work's purpose, only one hemisphere is necessary. The edges of the critical feature act as the division boundary for the cited spherical parameterization method. The selected mesh region corresponding to the critical feature is projected on to a hemisphere which robustly enables the extraction and transformation of the critical feature into a planar decal texture. One issue that arises with the hemisphere planar projection is that there is a center concentration effect where the triangle at the middle of the circular disc is much smaller in size than the edge triangles. This causes the texture to concentrate about the center of the disc, which is not ideal. To correct for this, a circular distortion is imposed on the extracted planar decal texture in order to enlarge the center region. Figure 6.5 illustrates the decal texture extraction method with the distortion correction.

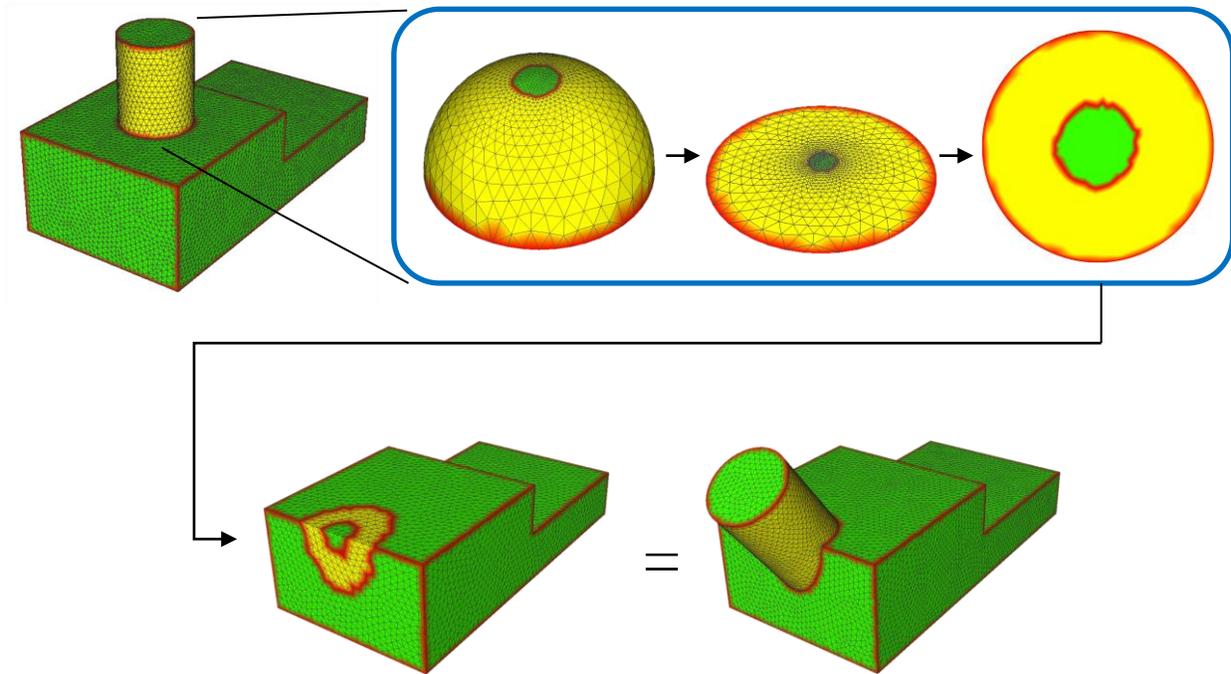


Figure 6.5 Decal texture extraction and projection.

Once the projection is completed, a planar decal texture representing the critical feature is created. It is important to note that all feature parameters for the critical feature is still associated with the decal texture. The decal texture not only represents the feature type of the critical feature, it also carries the feature parameters of the critical feature. Therefore, the movement of the decal texture implies changes in its associated feature parameters. In essence, a projected decal texture represents a phantom feature.

With the decal texture extracted, the user can now move this decal texture to any local on the mesh. As mentioned previously, the decal texture is treated as a separate texture layer to that of the remainder of the model texture. Local parameterization [124] is used to allow the decal texture to flow across the surface of the mesh, enabling interactive feature movement and

placement. The user can not only move but also resize the decal texture as they desire. This allows the user to place the critical feature at the intended location, and scale the feature to the desired size. The movement and the resizing of the decal texture propagates into the feature parameters that are associated with the decal texture. The feature parameters are updated in accordance to the user's modifications. Once the placement of the decal texture is finalized, the mesh encapsulated by the decal texture adapts to represent the feature (or features) which the decal texture represents. The texture is then reintegrated with the rest of the model texture.

Nonetheless, before the decal texture can be reintegrated, the model texture where the decal texture is extracted from needs to be repaired. The triangles in this area also need to adapt to the new feature setup. The mesh adaptation occurs at the decal texture extraction step; however, to stay within the scope of texture editing, mesh adaptation is outlined separately in Section 6.2.3. Essentially, decal texture extraction is equivalent to a feature removal operation. With the texture at the original location moved, the texture is left "blank" in the old feature location. To repair the blank texture, an erosive technique is used (seen in Section 4.3.3). The erosive technique first classifies the blank area as an edge area and then subsequently allows the adjacent feature to erode into the blank area. The erosion occurs such that the triangles within the blank edge area are reclassified into its adjacent features' feature definitions. They take on both the adjacent feature types (color) and feature parameters. The erosion occurs until the blank edge area is reduced down to one-triangle-wide connected strips. To extract the edges, the one-triangle-wide connected strips are further eroded down to edges through reclassifying the triangles with the lowest dihedral angle into its adjacent features such that the reclassification does not result in any broken edges.

With the reclassification, three possible edge cases result: a single edge point, a new edge attached to original edges and a new edge attached to new edges. Figure 6.6 shows the corresponding locations of blank areas that result in the three different cases. For the first case (Fig 6.6a), a single edge point is derived from a blank texture area situated on a single feature with no contact with edges. This single edge point is simply reclassified into its surrounding feature definitions. For the second case (Fig 6.6b), a new edge attached to original edges is derived from a blank texture area situated on multiple features and edge or edges without the blank texture area bridging opposing feature edges. In this case, the new edges accurately represent the missing edges at the blank texture area. For the third case (Fig 6.6c), a new edge attached to new edges results from a blank texture area situated on multiple features and edges with the blank texture area bridging opposing feature edges. For this case, the edge is an accurate representation of the missing edges or it is an extra edge that does not actually represent the underlying feature edge setup within the blank texture area. To determine if this edge is an actual edge or not, the feature definition of the patches adjacent to this edge is compared. If the feature definitions are identical then this edge is a false edge and is removed. If the feature definitions are not identical then this edge is a real edge and is kept. Once these three cases are resolved (if they were present) the model texture is repaired and completed. If the extracted decal texture is not applied at a new location, the edited mesh model is represented by the most up-to-date model texture with the critical feature texture removed. The adapted mesh, with its vertex locations updated to reflect the edited model texture and its associated feature parameters (feature information layer), becomes an accurate representation of a structurally altered prismatic engineering triangle mesh model.

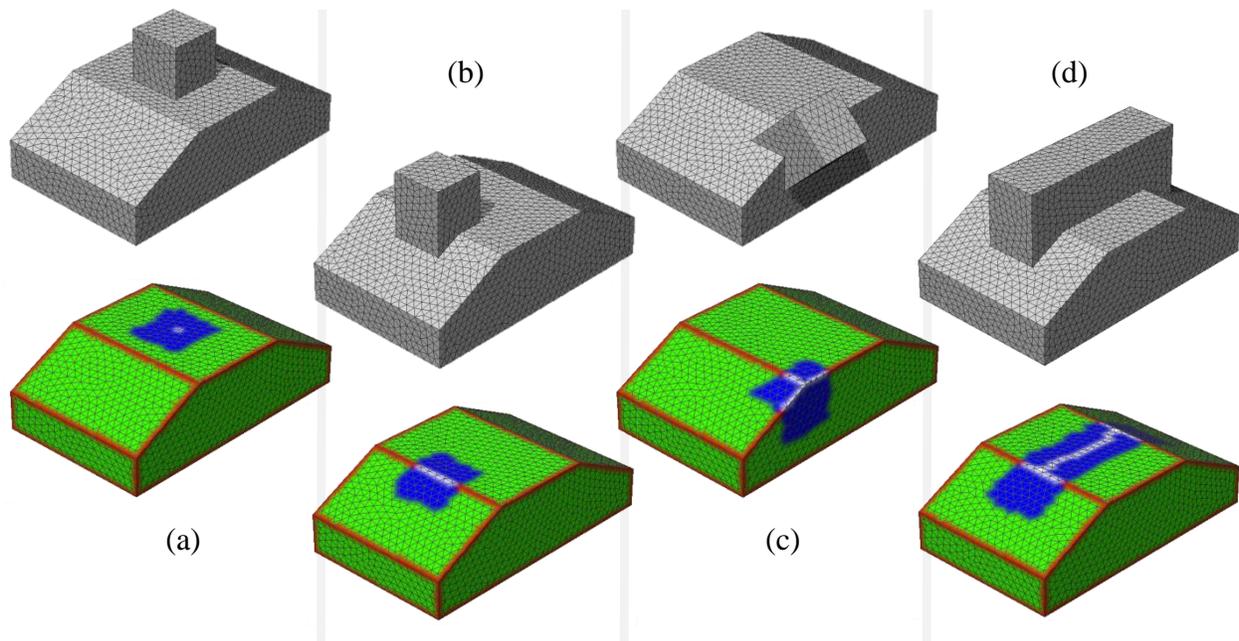


Figure 6.6 Edge extraction through erosive technique.

6.2.3 Mesh adaptation

Along with the movements and changes to the texture of the model, the mesh needs to adapt in order to accurately represent the edited prismatic engineering triangle mesh model with good mesh quality. Structure-altering edits can be viewed similar to that of the large scale edits seen in the non-structure-altering edits section since the deformation necessary to create the new features and to remove the old features are relatively large. The mesh needs to adapt in order to retain the current mesh quality and to accurately represent the new features. Therefore, the adaptive meshing technique mentioned and cited in Section 6.1.2 is used here in order to maintain consistent mesh quality. Figure 6.5 illustrates the resulting triangle mesh quality achieved through adaptive meshing.

6.2.4 Genus alteration

On occasion, genus changing operation is required. Such structural changes usually result from through-hole operations. The proposed method deals with this type of special structural changes through self-intersection detection, hole filling operation and local idealization. If the user specifies a hole, or to be general, a concave feature with large enough depth to cause self-intersection, the self-intersecting triangles are detected and removed. This usually leaves three bodies of triangle patches: the main model, the now islanded-end of the concave feature and another islanded patch where the concave feature intersected the main model. The latter two are removed. The gap which surrounds the self-intersected area is bridged by first connecting two of the closest boundary triangle edges of the gap and then closed by performing a hole-filling operation [125]. The connection of the two of the closest boundary triangle edges is to ensure that the hole filling algorithm does not treat the gap as two separate holes. These filled triangles are classified as edge triangles and the erosive technique is used to extract the edges. The adjacent feature information (and color) is imposed on the filled triangles in order for the processed area to accurately represent the desired edited model. Genus of the object is increased with the color mapping updated to reflect the through hole feature. Figure 6.7 illustrates the genus changing process.

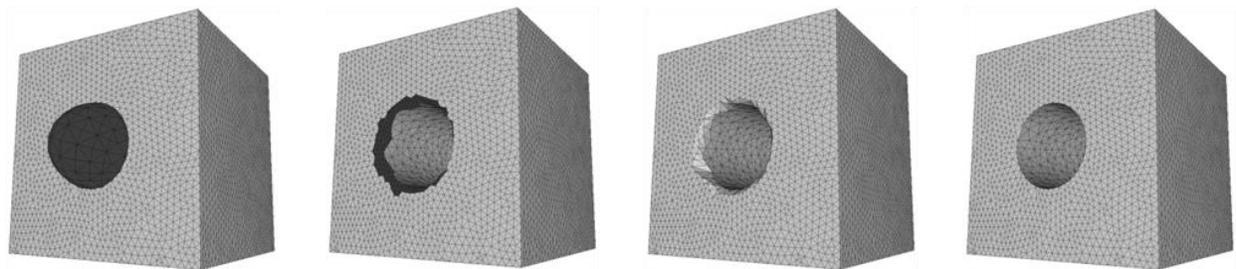


Figure 6.7 Genus increase operation.

To decrease the genus number of the model, through-hole removal operation is usually involved. In order to completely remove through holes or alter through holes into blind holes, one of the edges of the through-hole is collapsed into a single point. This single point is then broken up into two in order to separate the concaved through hole feature from the surface which it intersects with. Once the separation occurs the stretched triangles are processed with the adaptive meshing technique. With the number of triangle at the previously intersected surface increased, the erosive technique outlined previously for structure-altering edit is used on this surface in order to reclassify the adapted region to that of its neighbors (with color mapping update). For the adapted concave feature, the feature is either completely removed through the structure-altering editing scheme or is processed into a blind hole by projecting a plane texture to the adapted triangles of the concave feature. The choice of operation depends on if the user is removing the hole or changing it to a blind hole. The above process is illustrated in Figure 6.8.

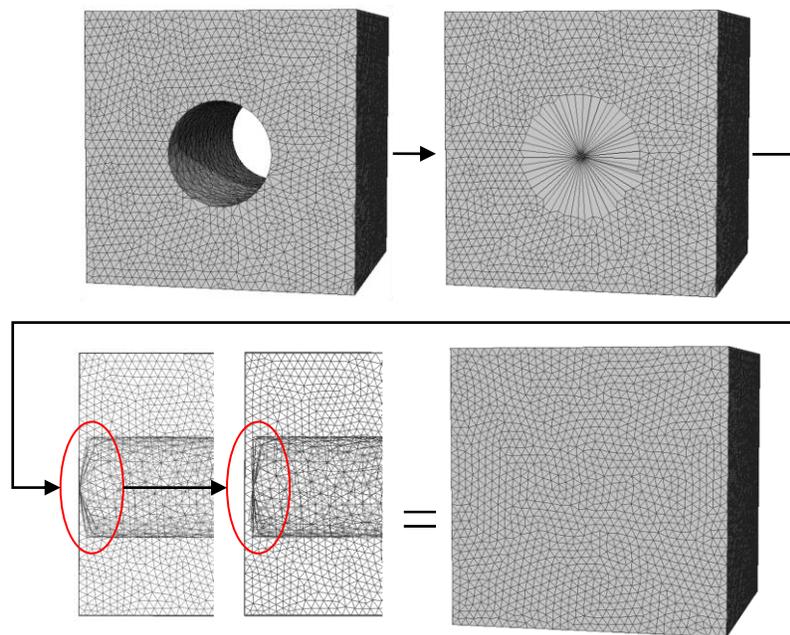


Figure 6.8 Genus reduction operation.

If the user is moving the through-hole instead, the modeling scheme performs the above technique for genus reduction and transform the through-hole into a blind hole. With the genus reduction technique, the through hole can be parameterized and projected into a decal texture. The decal texture can be moved to any location on the model as the user requires. Once moved to the desired location, the hole is reconstructed based on its previous parameters utilizing the method outlined in Section 6.2.2 and 6.2.3. If the hole feature causes self-intersection, the intersected triangle are removed and the above genus increase technique is used to create the proper geometry.

This technique is also used when non-zero genus features are moved. Any through-hole feature on the critical feature is reduced to genus zero through the above reduction technique. With the genus reduction, the features are decreased to genus zero and is thus readily projected into a planar decal texture. With the planar decal texture realizable, non-zero genus features can be moved as a decal texture. Once the reduced-genus projected decal texture is moved, the features are reconstructed with the through hole features reconstructed lastly. The through-hole feature start out as blind holes which intersects the opposite feature (or features) signaling a genus increase operation. Genus increase is thus performed and the through hole is recovered. The mesh adapts to the new geometry. The non-zero genus feature is moved to the desired new location. With genus changing technique readily available for prismatic engineering mesh modeling, non-zero genus features can be moved as desired and genus-changing operation can be performed as per user's requirements.

6.3 Case studies

The following sections illustrate the proposed mapping-based flexible feature-based model editing method with structure-altering capability. The first subsection shows the capability of non-structure-altering model edits with the subsequent subsection outlining the capability of structure-altering model edits. Further cases are included at the end of the sections.

6.3.1 Non-structure-altering edits

The first case is an object which consists of features with heavy interdependencies. As shown in Fig. 6.9, the object is made up of planes, cylinder and spheres with tangential continuity existing between all types. If these continuities need to be kept during design modification, the proposed method solves the dependent features' parameters based on the edited feature. In the case where one of the sphere is moved, the location and orientation of the cylinders are solved such that the tangential continuities are not violated (Fig. 6.9a). Furthermore, if one of the planes is moved as shown in Fig. 6.9b, the tangential continuities are automatically satisfied through satisfying the positional continuities. If large non-structure-altering edits are required, the triangle mesh is adapted in order to maintain mesh triangle quality (Fig. 6.10). The feature types and parameters are mapped on to the adapted areas in order to ensure that the new triangles geometrically represent the desired edited model. With the mapping technique, the mesh can adapt as required while accurately representing the desired model. Large scale edits do not have detrimental effects on the mesh quality. Figure 6.11, shows two more models with non-structure-altering edits. The features location and size parameters are freely changed within the structural confinement of the models.

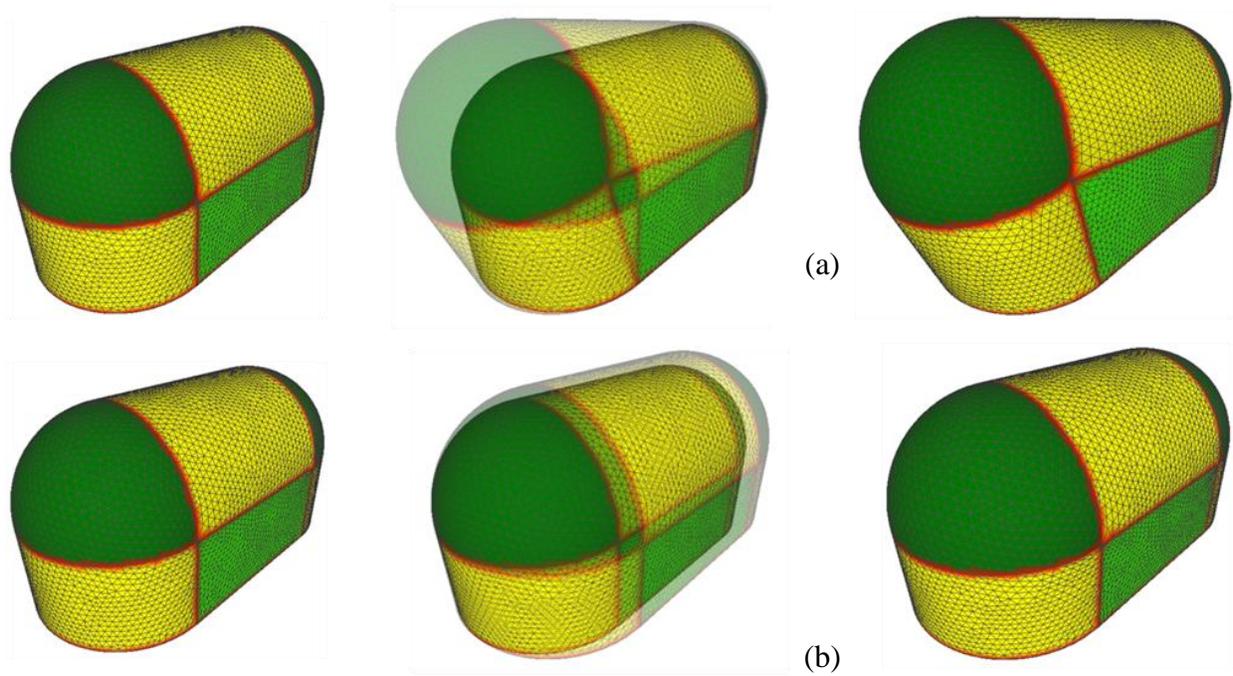


Figure 6.9 Non-structure-altering edits with tangential continuity conditions.

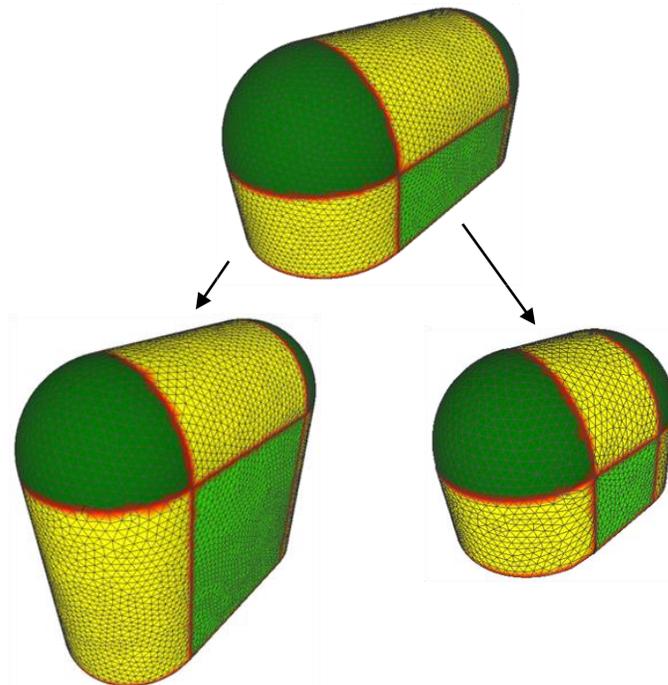


Figure 6.10 Non-structure-altering edits with large movements.

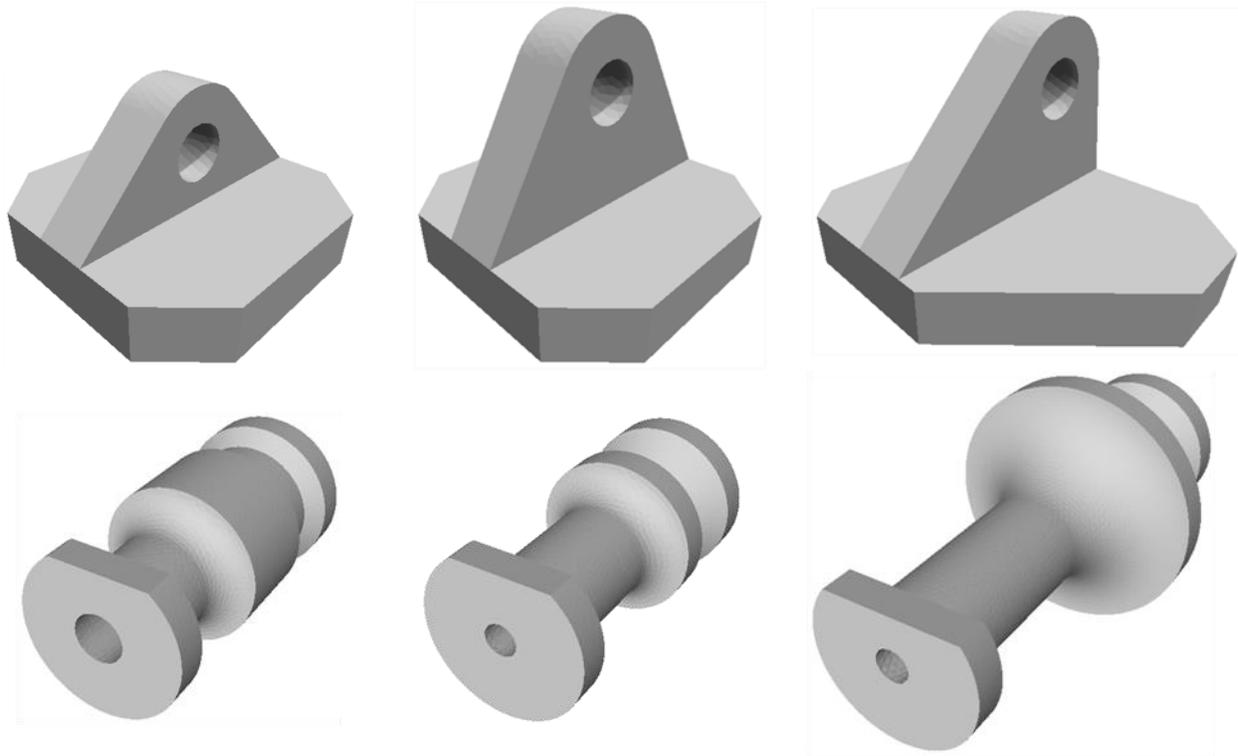


Figure 6.11 Non-structure-altering edits – supplementary examples.

6.3.2 Structure-altering edits

For structure-altering edits, the texture mapping technique is more prominent. A one step rectangular block with a boss attached, previously shown in Fig. 6.5, is used as the first example to show case the current method. In this instance, the boss is moved from the edge back to the top plane. To move the boss, the feature is first converted to a decal texture (Fig. 6.12a). The area where the decal texture is extracted from is adapted and eroded. The triangles where the boss is located are now part of the top plane. Figure 6.12b shows the result of the mesh adaptation and erosion. The boss, represented by the decal texture is moved to the top plane. Once placed, the triangles vertices encapsulated by the decal texture of the boss are deformed (per feature based on distance from decal texture edge) to take on the new geometric information

defined by the decal texture. Adaptive meshing is used to split the stretched triangles into good quality triangles (Fig. 6.12c). If the decal needs to be moved again, it is converted back into a decal texture. The same process repeats. Figure 6.13 shows various positions of the boss moved by the feature-based model editing method. The boss can be placed at any location on the mesh. Furthermore, extra bosses can be added to the model simply by selecting the boss and copying the extracted decal texture. The bosses can even intersect each other as shown in Fig. 6.13. If desired, the bosses can be removed, arriving at just the one step rectangular base. By using a texture mapping technique with associated feature information, models reflecting the desired structure-altering edits are realized. Features can be imposed on any portion of a mesh.

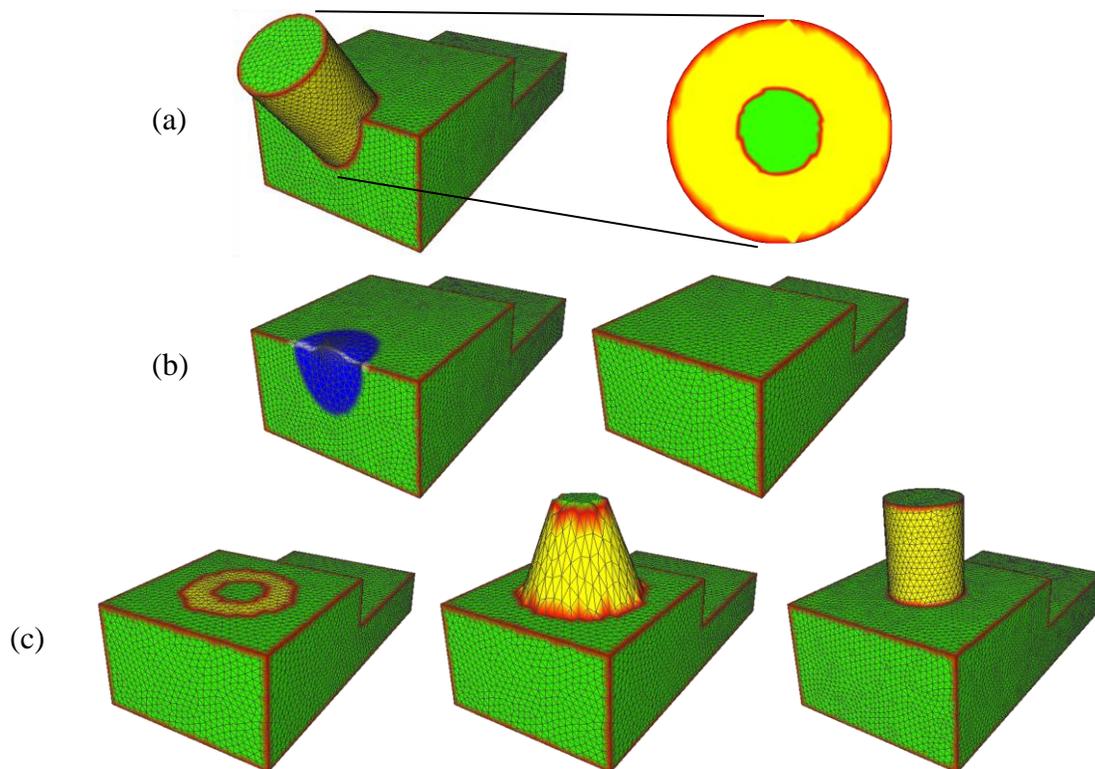


Figure 6.12 Structure-altering edits – step-by-step: (a) decal texture extraction; (b) original location edge extraction and mesh adaptation; (c) new feature location decal projection and mesh adaptation.

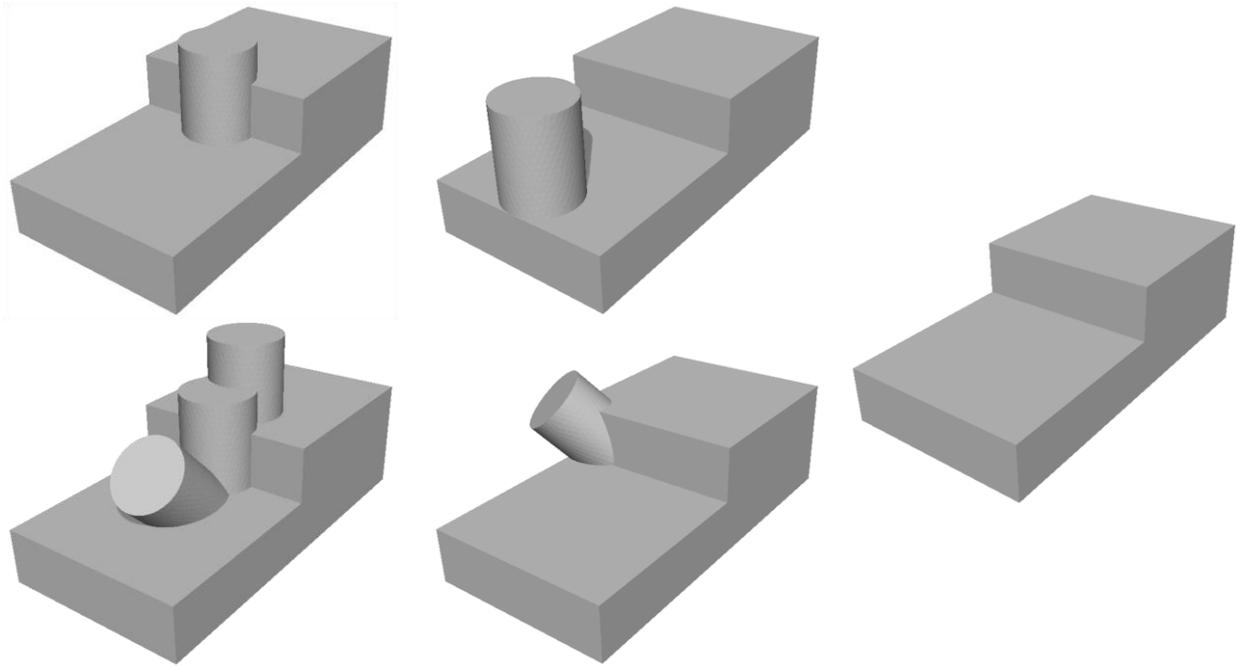


Figure 6.13 Structure-altering edits.

To further illustrate the capability of the proposed structure-altering edits technique, genus changing examples are shown. Figure 6.14 shows the elongation of a blind hole which causes the hole to penetrate the opposite surface of the model. The intersected area is processed and the through hole is constructed. It is shown that with the genus increase method, the through hole is constructed appropriately even though the intersecting surface possesses multiple geometries. The correct edges are determined and the proper geometry is realized.

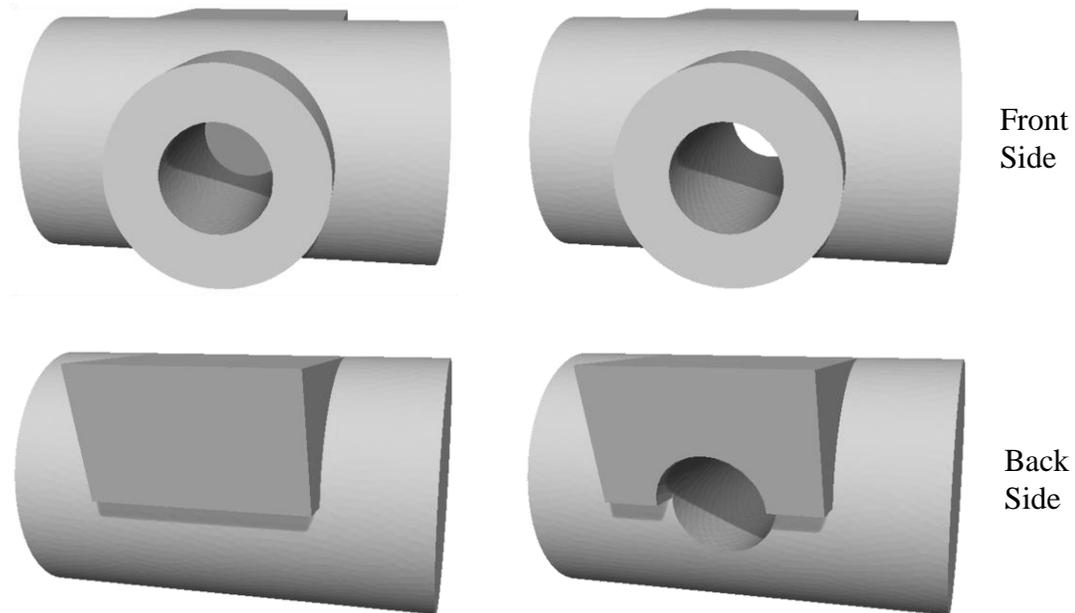


Figure 6.14 Structure-altering edit with genus increase.

6.3.3 General edits

With the proposed scheme for feature-based triangle mesh model editing with structure-altering capability, prismatic mesh model can be edited in a manner similar to that of direct modeling seen in parametric CAD modeling. Mesh models can be changed in accordance to their current feature structure or they can be changed freely as the user desires. Features can be moved, scaled, rotated, deleted and added, facilitating straightforward feature-based modeling in the triangle mesh domain. Figure 6.15 shows three more examples showcasing the capabilities of the proposed method. The left most model represents the unedited version of the model. It is shown in the figure that non-zero genus features are moved utilizing genus reduction, tangential continuity of non-structural edits are automatically satisfied, new geometries are straightforwardly added and complex boss geometries are moved and copied as the user desired.

With these capabilities at the disposal of the user, feature-based model editing for prismatic triangle mesh models is realized.

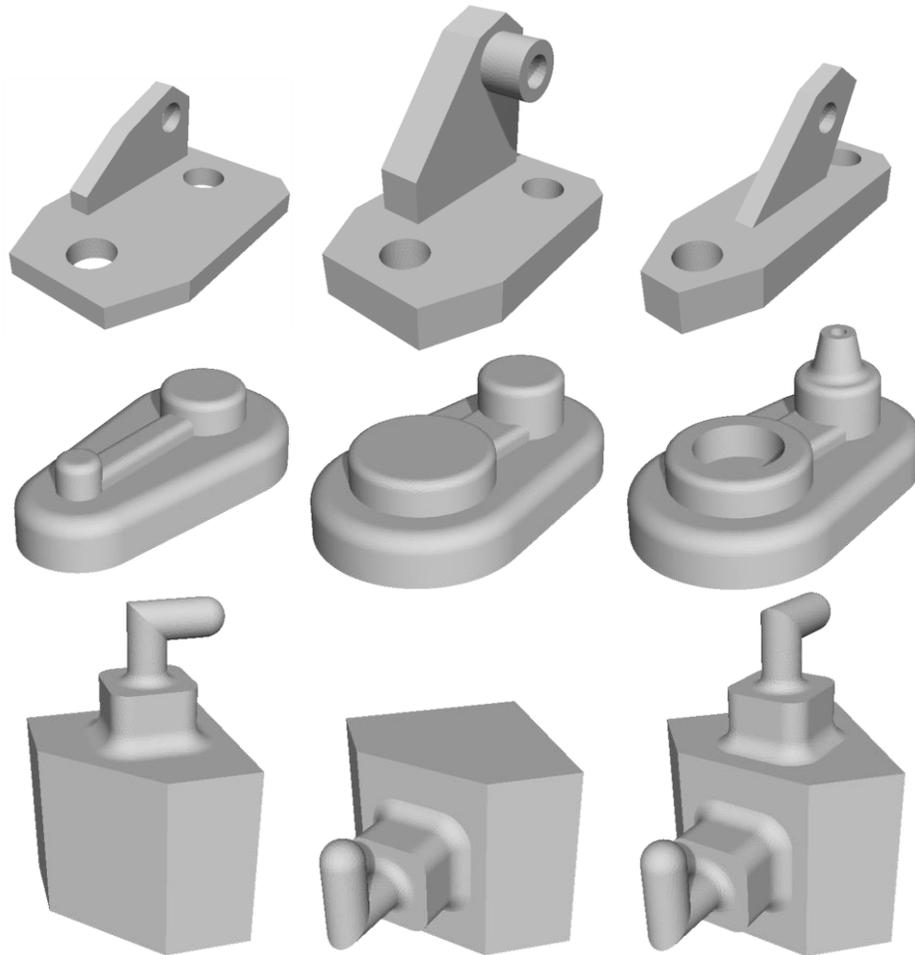


Figure 6.15 Structure-altering edits – supplementary examples.

6.4 Summary

This module proposed a feature-based prismatic triangle mesh editing scheme that enables both non-structure and structure-altering edits in the mesh regime. The method permits feature alteration through straightforward push-and-pull techniques. Feature parameters and locations

are easily changed through these interactions. The method is able to handle not only simple prismatic features but also complex non-zero genus prismatic features. Non-zero genus prismatic features can be moved, scaled, added and deleted as the user pleases, making this a capable feature-based engineering triangle mesh editing scheme. The features and their relationships can be altered with the triangle mesh automatically updated to suit. With this feature-based mesh model editing scheme realized, the proposed hybrid modeler can achieve the desired modeling flexibility.

Nevertheless, more elaborate genus reduction technique would further increase the capability of this work. The proposed method can effectively perform structure changing and non-structure changing edits to prismatic engineering mesh models, but further work is required for acquiring planar decal texture projection of complex through-hole features. At the moment the proposed method is capable of moving non-genus-zero features, with the requirement that the through-hole feature at the intersection is single-featured. A through-hole feature can be more than a simple circular or square hole. Through-hole features can possess highly complex cross sections and/or geometry, requiring more sophisticated genus reduction technique. Therefore, instead of simply collapsing one end of the through-hole feature to reduce the genus number, a more complex collapsing technique is required. When multiple features are present at the intersection, a technique based on the extraction of the spine of the cross section can be explored [126]. By collapsing the intersecting edge to the calculated spine, more appropriate hole closure should result, allowing the genus reduction of complex through holes. Furthermore, if the features being moved contain many features and is very large compared to the attachment area, the proposed algorithm (without iteration) cannot reliably reconstruct the features at its new location.

An iterative approach is necessary where the vertices are incrementally moved towards the final location with mesh adaptation imposed at every step in order to ensure dependable results.

However, as the proposed method stands, a large variety of existing prismatic engineering models can be processed. Non-structural and structural edits can be performed by the users. Feature-based triangle mesh editing is enabled in this module, allowing the full realization of the proposed flexible hybrid modeling scheme. Users have the choice of utilizing the most convenient modeling methods for deriving their designs. Unconstrained feature-free modeling is available along with feature-based model editing.

Chapter 7: Hybrid CAD Modeler

With the proposed hybrid modeler realized through the combination of the four modules, flexibility in engineering design modeling for prismatic mechanical parts is increased. Unconstrained model modifications that were previously not attainable via current parametric CAD modelers are enabled while the direct modeling capabilities seen in the current parametric CAD modeling regime are mimicked. By combining the advantages of the two modeling methods through the revamping of the inner workings of CAD modeling, a flexible geometric modeling scheme for prismatic mechanical parts is realized. The flexibility of triangle mesh modeling is combined with the accuracy and precision of feature-based parametric modeling. Features no longer restrict the CAD modeling flexibility.

7.1 Methodology recap

To offer the said flexibility, the model representation of the proposed CAD modeler no longer hinges on individualized feature definitions. Homogeneous triangle mesh representation, which can discretely represent any geometry, is utilized boosting the geometric manipulation capability of the proposed hybrid modeling scheme. Such triangle mesh representation not only allows for exceptionally flexible geometric manipulation capabilities but it also allows for a format translation free representation across different engineering design tools. Technologies such as 3D printing and 3D scanning, which are all mesh based, are all straightforwardly integrated with the proposed hybrid CAD modeler. Objects can be scanned, modified and 3D printed in a

shorter period of time. Design updates can be more streamlined, leading to faster design iterations.

In order to effectively utilize the triangle mesh representation for CAD purposes, or rather for creating 3D engineering design models, four modules were proposed and outlined prior to this section: feature segmentation, mesh idealization, unconstrained feature-free model modification and feature-based model editing. As a reminder, the feature segmentation and mesh idealization modules impose the notion of feature and realize the precision and accuracy seen in parametric CAD modelers on the featureless triangle mesh representation. While the unconstrained feature-free model modification and feature-based model editing modules enable the flexible model manipulation component of the hybrid CAD modeler. With the four modules, users can freely manipulate geometries as they see fit or simply push-and-pull existing prismatic features on a mesh model as required by the design. All inner workings of the algorithms are transparent to the user, leaving the user with a less cumbersome CAD modeler.

7.2 Implementation details

To realize the proposed flexible hybrid modeler, Sculptris is used in conjunction with MeshLab [127], where MeshLab is the developmental platform. MeshLab is an extensible and open source system created for the processing of 3D triangular meshes. The system is based on the Visualization and Computer Graphics Library (VCG) [128], which is an open source C++ library for the processing, display and manipulation of triangle meshes.

With the extendable characteristic of MeshLab, three of the four modules (Feature Segmentation, Mesh Idealization and Feature-Based Model Editing) are implemented as plugins to the main MeshLab platform. Each module is coded in C++ based on the MeshLab structure. Any function pre-existing in MeshLab is used directly in the development of the four modules. Furthermore, MeshLab permits the creation of custom mesh attributes. This allows the straightforward realization of the proposed feature information layer and the proper association of the feature information layer to the triangle mesh model in the MeshLab platform. The feature information layer is composed of feature type, coded in face and vertex color attributes and feature parameters and feature continuity conditions, both coded in custom face and vertex attributes.

MeshLab, together with the three implemented modules in plugin form, interacts with Sculptris and Geomagic Studio 12 [129] through import and export operations to complete the flexible hybrid modeler prototype. Sculptris is responsible for performing all the unconstrained feature-free model modifications seen in this thesis and Geomagic Studio 12 is responsible for pre-processing and triangulating the scanned point-cloud data to create scanning-derived mesh models. Full integration of all four modules within one platform is not yet completed. Nevertheless, without full system integration, the interaction of multiple platforms adequately allows for the demonstration of the hybrid modeler's capabilities. Figure 7.1 is an abstract illustration of the current setup of the hybrid modeler prototype.

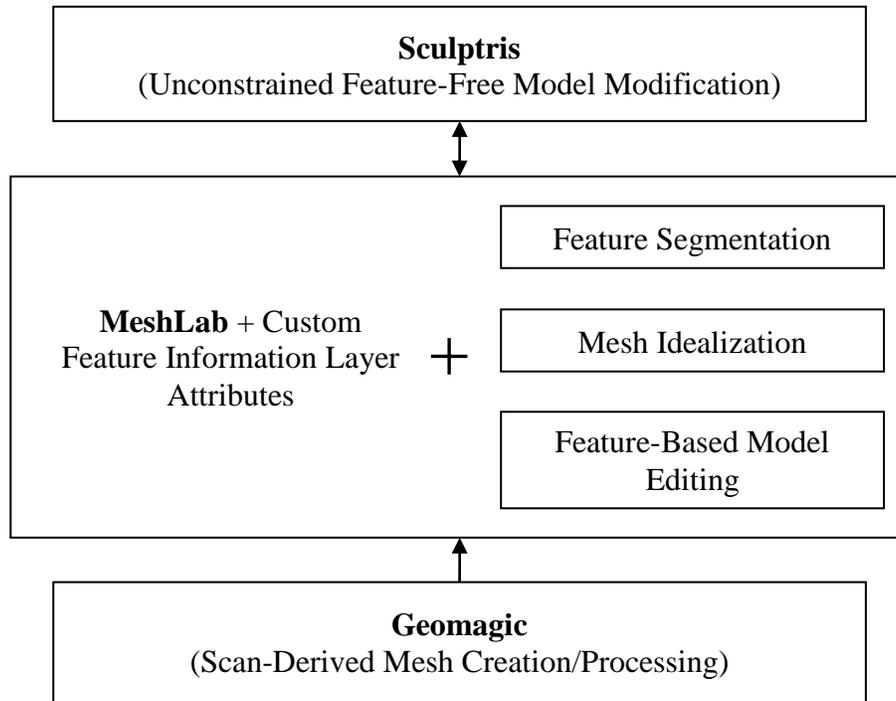


Figure 7.1 Implementation setup of proposed flexible hybrid CAD modeler.

7.3 Case studies

To demonstrate the hybrid modeler’s capabilities and the complete operation of the proposed hybrid modeling scheme, work-through case studies are shown in the subsections. Note, each feature is specifically color coded: plane - light green, cylinder - yellow, sphere - dark green, cone - light blue and torus - dark red.

7.3.1 Case 1: basic modeling via proposed hybrid CAD modeler

Case 1 (shown in Figure 7.2) brings one of the very initial examples shown in Fig. 1.6, which graphically illustrates the proposed method, to life. Through the modules outlined in this thesis, the initial model with two spheres tangentially continuous to a torus is freely modified to the

final model, which consists of two spheres, a torus and a cone. Instead of deleting existing features and creating new ones, unconstrained feature-free model manipulation is utilized to drastically change the shape of the object as shown in Figure 7.2. The user's involvement in the modeling process is only to manipulate the model geometry close to the desired shape via Sculptris. The manipulated shape is automatically identified and idealized into prismatic engineering geometries with no further user interactions. The proposed method transparently aided in the creation of the ideal geometry. Cumbersome modeling menus are avoided via the proposed hybrid modeling scheme. An ideal 3D engineering design mesh model with accurate features is derived. Additionally, final model adjustments performed at the end this case study is simply completed via the feature-based model editing method with the push-and-pull technique. The rest of the feature updated automatically in order to not violate the previously established/detected continuity conditions. In this case study, the initial geometry is straightforwardly transformed into the final design with the actions such as feature type changes and feature number increases all being completely transparent to the user.

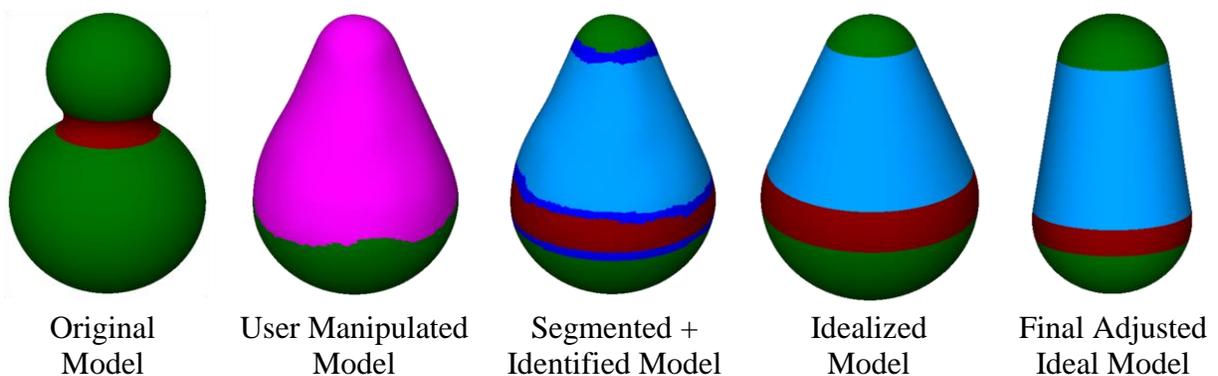


Figure 7.2 Case 1: basic modeling via proposed hybrid CAD modeler.

7.3.2 Case 2: complex modeling via proposed hybrid CAD modeler

Case 2, shown in Fig 7.3, further validates the capability of the hybrid modeler. The geometry of the model is heavily manipulated in this case. The initial model consists of six features while the final model consists of 17. Planes are counted as separate features. As mentioned early on in the thesis, features are the elementary elements in a model. For this case study, a genus change is also involved in the model modification process.

Given the heavy amount of modification imposed on this model, the user utilized only unconstrained feature-free model modification operators. The geometry is manipulated by the user in a clay like manner in order to achieve the shown geometry modifications. The hybrid modeler automatically identifies and idealizes the modified areas at every geometry alteration step, creating the ideal 3D engineering design model for the user. The lack of need to deal with menus and dialogs allows the user to quickly realize the bulk shape of the design and to also quickly update the design. There is no need to setup datum plans, draw feature defining curves and to perform feature operations. The user's idea is transformed directly into geometric manipulations which resulted in the necessary changes.

As shown in the Case 2 figure, the bulk shape is straightforwardly modeled. The final step in the figure involves the usage of the feature-based model adjustments method, much like Case 1. Again, the user simply adjusts the features through simple push-or-pull operators. This allows quick and precise placement of features with accurate feature parameters. As shown by Case 1 and Case 2, through the combined efforts of feature segmentation, mesh idealization, unconstrained feature-free model modification and feature-based model editing, increase in

modeling flexibility and guarantee in model precision and accuracy are simultaneously achieved via the proposed hybrid modeling scheme.

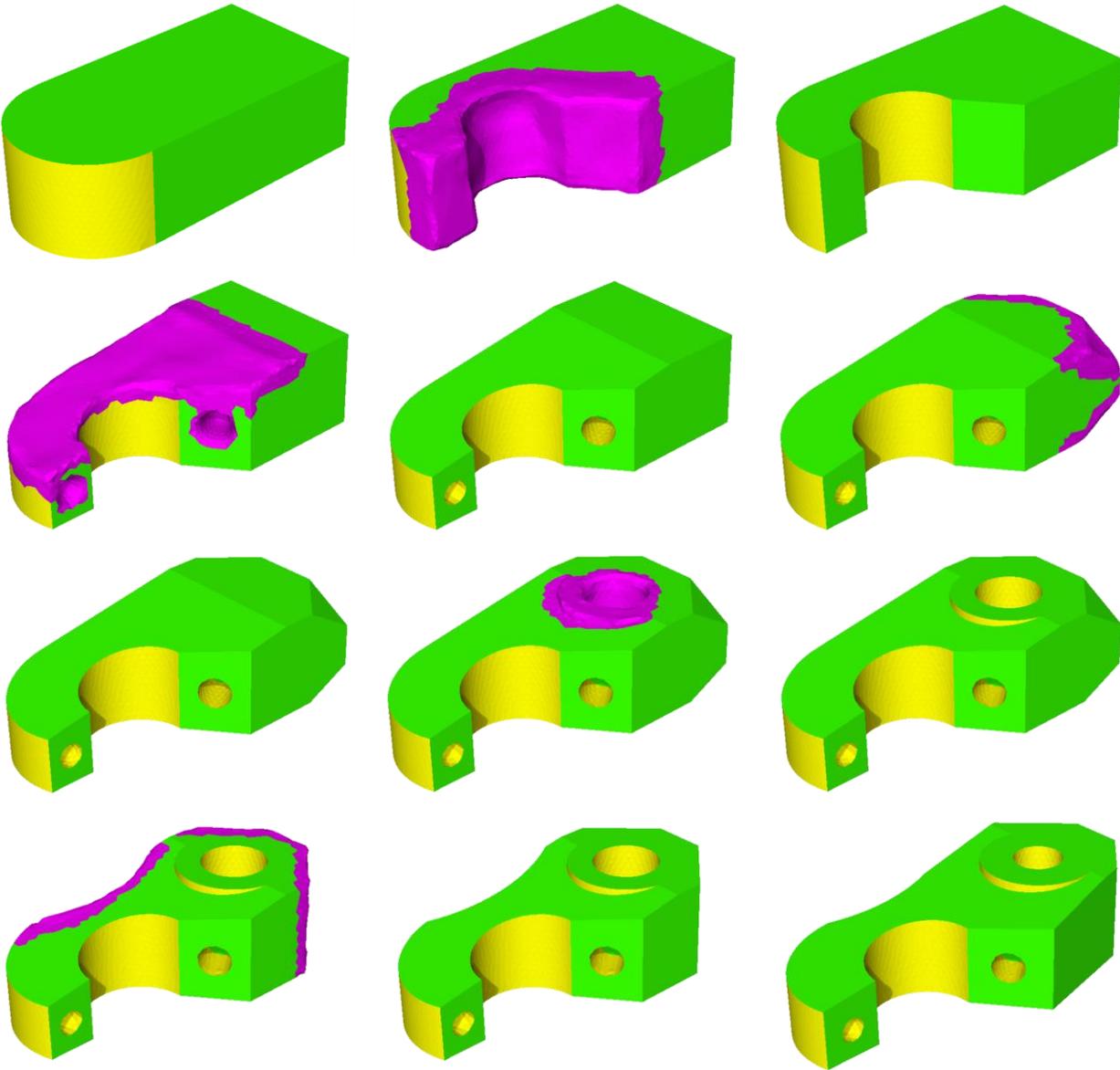


Figure 7.3 Case 2: complex modeling via proposed hybrid CAD modeler.

7.3.3 Case 3: complex modeling via proposed hybrid CAD modeler – reverse order

Note though, feature-based model editing does not always have to be performed at the end of the modeling steps for the sake of fine tuning the model. It can be used at any time at any step. As shown in Fig. 7.4, Case 3, feature-based model editing is used to initially alter the features to the desired size; since for the required model changes at this stage, feature-based modeling operations are seemingly more straightforward. The user push-or-pull on the feature-of-interest until the desired size is achieved. Once the desired size is achieved, further feature-based model editing is utilized to adjust the locations of the features. Again, the user grabs the features and moves them to the desired locations. With the desired size and location achieved, the top and bottom blocks along with the middle square pillar with its chamfers are transformed into cylinders via unconstrained feature-free model modification. The cylinders are then further manipulated into cams in the final stages through unconstrained feature-free manipulations. Mesh idealization is transparently applied to transform the triangle mesh model into an accurate 3D engineering design triangle mesh model.

Thus, the proposed hybrid method not only enables flexible geometric modeling through unconstrained feature-free modeling with feature segmentation and mesh idealization, but it also capitalizes on this flexibility through incorporating feature-based model editing. At any point during modeling, either approach can be utilized (hybrid). The user can push-and-pull existing features to the desired locations or can create or alter features through unconstrained feature-free modeling. In essence, the proposed hybrid method gives the user increased flexibility in geometric manipulation. The user is less constrained by the rigidity of the CAD modeler.

Unconstrained geometric manipulation, which is not possible and not available for the current parametric CAD modelers due to their model representation and feature management approach, is now available to the user in conjunction with feature-based model editing.

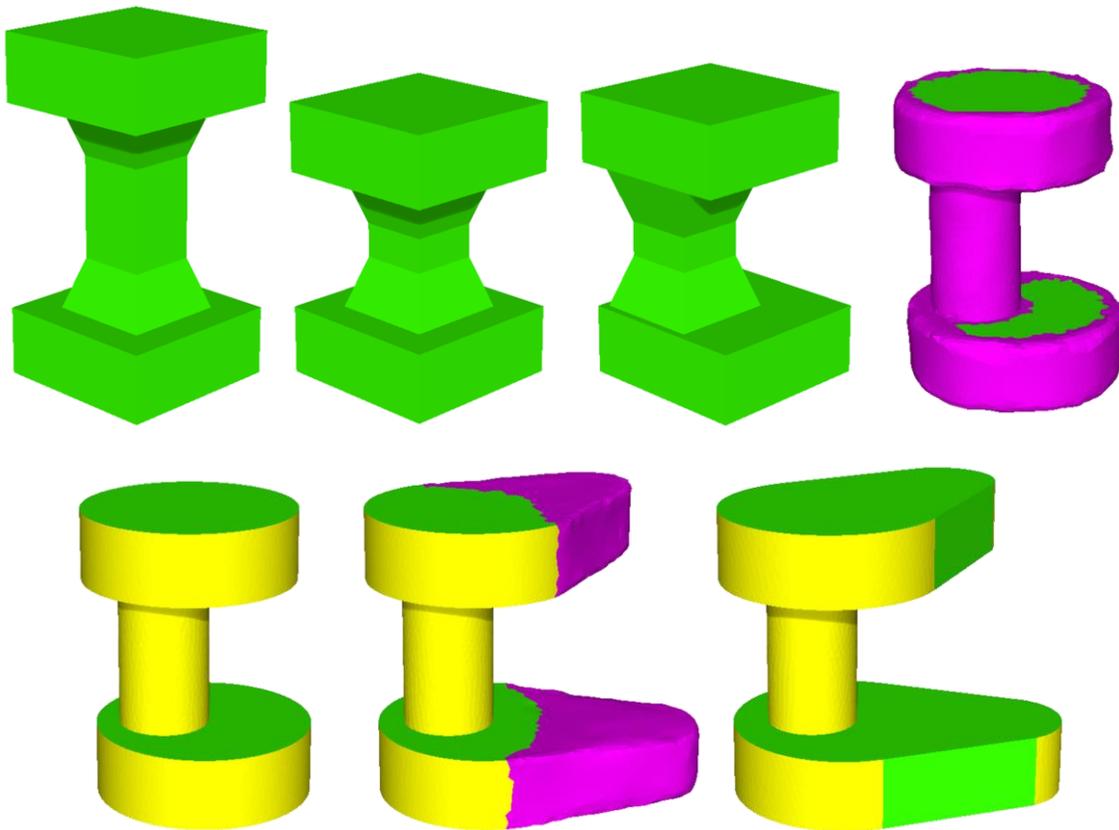


Figure 7.4 Case 3: complex modeling via proposed hybrid CAD modeler – feature-based model editing operations first.

7.3.4 Case 4: complex modeling via proposed hybrid CAD modeler – mixed usage

Case 4 additionally illustrates the flexibility of the proposed hybrid modeler. As shown in Fig. 7.5, the final model is derived through the combined efforts of unconstrained feature-free model modifications (with feature segmentation and mesh idealization) and feature-based model editing. To create the initial bulk geometry, unconstrained feature-free model modifications is

used (Fig. 7.5a) followed by feature-based model adjustments (Fig. 7.5b). To add some of the desired details into the bulk geometry, unconstrained feature-free model modifications is, again, used (Fig. 7.5c). Feature-based model editing is then utilized to make further alterations and unconstrained feature-free model modification is used to create some of the final detailed features (Fig. 7.5d). With the combined efforts of unconstrained feature-free model modifications and feature-based model editing, the final desired model (Fig. 7.5e) is achieved. This combination increases the flexibility of engineering design modeling. Users has the option of utilizing unconstrained manipulations and/or feature-based model modifications where they see fit. Figure 7.6 and 7.7 shows two more examples of the proposed method.

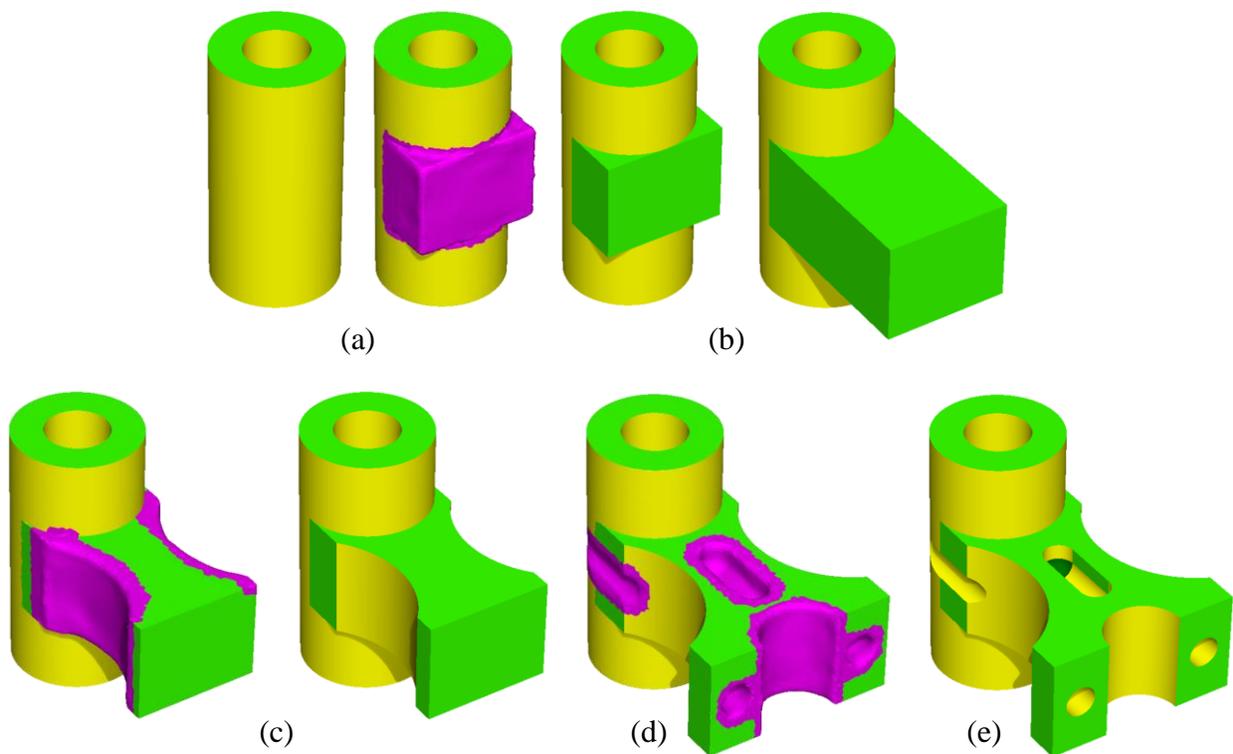


Figure 7.5 Case 4: complex modeling via proposed hybrid CAD modeler – mixed usage.

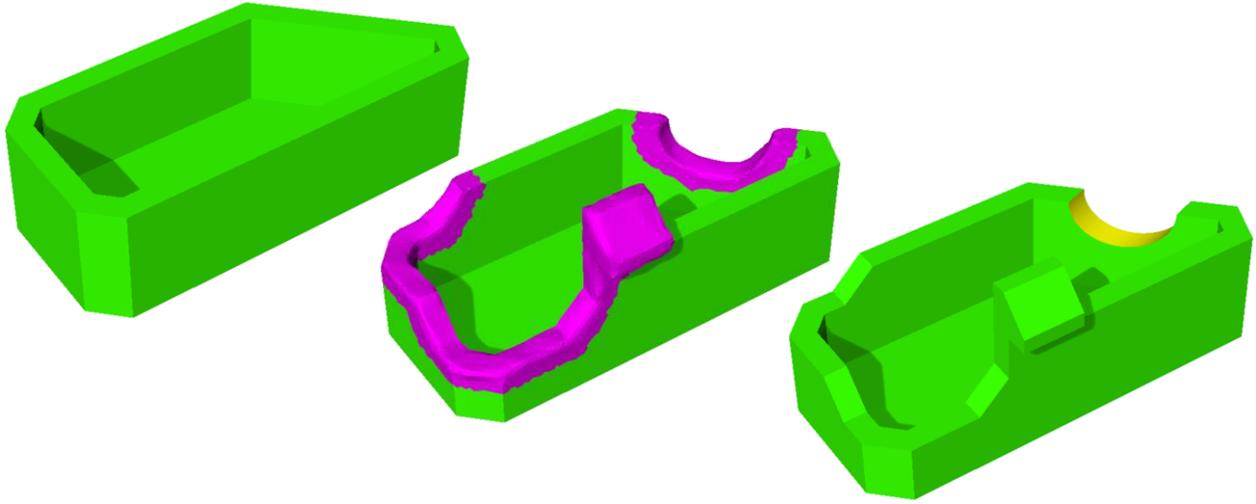


Figure 7.6 Case 5: hybrid CAD modeler supplementary example 1.

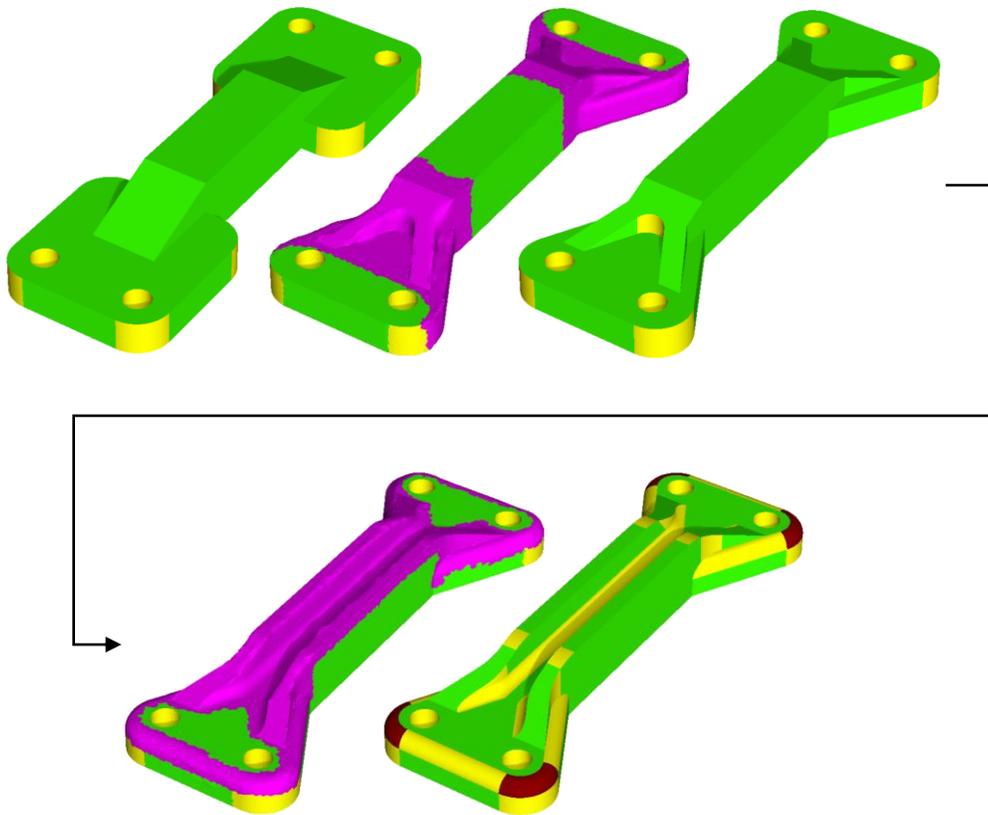


Figure 7.7 Case 6: hybrid CAD modeler supplementary example 2.

Chapter 8: Conclusions and Future Work

The parametric CAD paradigm has dominated the CAD scene for many decades. It has been the most powerful method of creating geometric design models for prismatic mechanical parts. Its feature-based model representation fundamentally possesses the trait to “put a number on everything”. It ensures geometric design models are accurate digital reflection of their intended designs. The features on the models are defined through accurate individualized mathematical expressions that suitably facilitate the encapsulation of design information. Coupled with its efficient use in the computerized environment, parametric CAD software is used for a wide array of applications in a wide array of industries. However, due to its approach (via model representation) to bring about precision, accuracy and feature documentation capabilities, parametric CAD modelers have heavily hindered its own modeling flexibility. Designers have to adhere to very specific modeling approaches dictated by the modelers. Very little flexibility exists in terms of modeling approaches. To alleviate such rigidity, this thesis proposed a novel modeling scheme that increases flexibility for CAD modeling without sacrificing precision and accuracy and feature documentation capabilities for prismatic mechanical parts.

This reduction in rigidity is achieved through the proposed triangle mesh based hybrid modeling scheme that combines the advantages of computer-graphics modeling techniques with that of CAD modeling methods. The hybrid approach separates the notion of feature from that of model representation in order to reduce modeling rigidity. Features exist as an information layer that merely controls the triangle mesh. This allows the model the ability and freedom to adapt and take on any prismatic feature definition at any mesh location. With this approach and its two

sub-modules (feature segmentation and feature idealization), unconstrained feature-free modeling modification in a CAD environment is possible. By utilizing triangle mesh model representation with the notion of feature imposed as a separate but associated information layer, unconstrained feature-free model modification and feature-based model editing are both achievable in the same modeler. Users are no longer constrained to one single modeling paradigm where either flexibility is sacrificed for precision and accuracy or precision and accuracy is sacrificed for flexibility. Modeling flexibility and model precision and accuracy are achieved simultaneously via the hybrid modeler. With this combination, two unique but now integrated modeling approaches are available to the users for the flexible modeling of prismatic mechanical parts.

Furthermore, the proposed modeling scheme is a tool that facilitates a constant forward evolving approach to design. Rather than deleting and recreating certain aspects of a design for design improvement purposes, the hybrid modeling method always imposes the design betterments directly on the current geometry. The information flow and growth is always advancing, enabling a simpler monotonic path to the optimal design.

In addition, the proposed modeling scheme also allows for better integration of state-of-the-art shape capturing and analysis tool and prototyping and manufacturing technologies. Physical objects can be scanned, meshed then idealized and edited directly in the hybrid CAD modeler without format translation. Design changes, determined through utilization of mesh-based analysis tools, can be quickly implemented in the hybrid CAD modeler and the part can be printed again for further testing. The proposed discrete modeling system much better integrates

the state-of-art design, analysis and manufacturing tools leading to a more streamlined design environment.

With further development, the proposed hybrid modeler can become even more versatile. Non-prismatic geometry processing capability (free-form surface features) can be developed and extensive feature-based modeling operations should also be integrated to fully realize the proposed hybrid modeler. However, these integrations can be challenging. Topics such as free-form surface feature processing are non-trivial since segmentation methods that typically function well for free-form shapes usually do not function well for prismatic geometry [52,53]. Furthermore, with the integration of non-prismatic features, more exhaustive idealization algorithm capable of processing these non-prismatic geometry is required. The need to together idealize prismatic features and free-form geometry while ensuring proper inter-feature relationship (design intent) highly increases the idealization complexity.

Nevertheless, the proposed segmentation and idealization algorithm has the potential of processing free-form features together with prismatic features. For feature segmentation, free-form features manifest themselves as baseline offsets in the per-vertex mean curvature histogram. By separating the baseline offset with the rest of the per-vertex mean curvature histogram, free-form features could be segmented. The feasibility of such baseline offset separation evidently needs to be studied further. As for the increase in the complexity of the idealization problem due to the inclusion of free-form features, the extendibility of the current idealization algorithm needs to be explored. The ability for the sequential idealization approach to deal with such increase in complexity is unclear at the current stage. For the integration of

additional feature-based operations, it is not far from an implementation exercise since most feature-based modeling operations are well-developed as seen in current feature-based parametric CAD modelers.

On the model representation side, triangle mesh refinement should be implemented in order to best control the triangle mesh with respect to engineering tolerances. The triangle mesh model should robustly refine its mesh density and quality in order to automatically satisfy tolerance requirement for engineering design work. A 3D triangle mesh engineering design model should always satisfy the required tolerances.

Overall, the proposed work demonstrated a hybrid CAD modeler that is more flexible than the current parametric CAD software when processing models with prismatic features. The work is still in its infancy, but its rudimentary form clearly shows the advantages of adopting mesh model representation with integrated feature imposition. With its triangle mesh representation and its four modules, the proposed flexible hybrid CAD modeling scheme provides users with multiple modeling approaches. Users are no longer bounded to one rigid method of creating engineering design models for prismatic mechanical parts. Designs can be more flexibly mocked up and evolved to the desired geometry.

References

- [1] Lalit Narayan, K., Mallikarjuna Rao, K. M., and Sarcar, M. M. M., 2008, Computer-Aided Design and manufacturing, Prentice-Hall, India.
- [2] Rao, P. N., 2004, CAD/CAM Principles and Applications, Tata McGraw-Hill, New Delhi.
- [3] Pottmann, H., Brell-Cokcan, S., and Wallner, J., 2007, “Discrete surfaces for architectural design,” *Curves and Surfaces Design: Avignon 2006*, Nashbroro Press, pp. 213-234.
- [4] Marsh, D., 2005, Applied Geometry for Computer Graphics and CAD: second edition, Springer-Verlag, United Kingdom.
- [5] Shapiro, V., 2002, “Solid Modeling,” *Handbook of Computer Aided Geometric Design*, (G. Farin, J. Hoschek, M.-S. Kim, eds.), Elsevier Science Publishers, pp. 473 – 518.
- [6] Austin Cottrell, J., Hughes, T. J. R., and Bazilevs, Y., 2009, Isogeometric Analysis Toward Integration of CAD and FEA, John Wiley & Sons Ltd, United Kingdom.
- [7] Farin, G., Hoschek, J., and Kim, M.-S., 2002, Handbook of computer aided geometric design, Elsevier, Netherlands.
- [8] Ault, H. K., 1999, “3-D geometric modeling for the 21st century,” *Engineering Design Graphics Journal*, **63**(2), pp. 33-42.
- [9] Ault, H. K., 1999, “Using Geometric Constraints to Capture Design Intent,” *Journal for Geometry and Graphics*, **3**(1), pp. 39-45.
- [10] Kim, J., and Pratt., M. J., 2008, “Standardized data exchange of CAD models with design intent,” *Computer-Aided Design*, **40**(7), pp. 760-777.
- [11] Schoonmaker, S. J., 2003, The CAD guidebook: a basic manual for understanding and improving computer-aided design, Marcel Dekker, Switzerland.
- [12] Srinivasan, V., 2004, Theory of dimensioning: an introduction to parameterizing geometric models, Marcel Dekker, Switzerland.
- [13] Stroud, I. A., and Nagy, H., 2011, Solid Modeling and CAD Systems: How to Survive a CAD System, Springer-Verlag, London.
- [14] Sansoni, G., Trebeschi, M., and Docchio, F., 2009, “State-of-the-art and applications of 3D imaging sensors in industry, cultural heritage, medicine, and criminal investigation,” *Sensors*, **9**(1), pp. 568-601.
- [15] Vergeest, J. S. M., Horváth, I., and Spanjaard, S., 2001, “Parameterization of freeform features, in: A. Pasko, M. Spagnuolo (Eds.),” *Proceedings of the International Conference on Shape Modeling and Applications, Shape Modelling International 2001*, pp. 20–29.

- [16] van den Berg, E., Bronsvort, W. F., and Vergeest, J.S.M., 2002, "Freeform feature modeling: concepts and prospects," *Computer & Industrial Engineering*, **49**(2), pp. 217–233.
- [17] Siemens PLM Software, 2013, NX8.5, <www.plm.automation.siemens.com>, retrieved on: August 20, 2013.
- [18] PTC Creo, 2013, FMX, <www.ptc.com>, retrieved on: October 10, 2013.
- [19] Autodesk, 2014, Fusion 360, <fusion360.autodesk.com>, retrieved on: December 3, 2014.
- [20] Cheon, S. U., Kim, B. C., Mun, D., and Han, S., 2012, "A procedural method to exchange editable 3D data from a free-hand 2D sketch modeling system into 3D mechanical CAD systems," *Computer-Aided Design*, **44**(2), pp. 123-131.
- [21] Gharib, I., and Qin, S., 2013, "Integration of sketch-based conceptual design and commercial CAD systems for manufacturing," *The International Journal of Advanced Manufacturing Technologies*, **68**(9-12), pp. 2669-2681.
- [22] Xu, B., Chang, W., Sheffer, A., Bousseau, A., Mccrae, J., and Singh, K., 2014, "True2Form: 3D curve networks from 2D sketches via selective regularization," *ACM Transactions on Graphics*, **33**(4).
- [23] Masry, M., Kang, D., and Lipson H., 2005, "A freehand sketching interface for progressive construction of 3D objects," *Computers & Graphics*, **29**(4), pp. 563-575.
- [24] Lipson, H., and Kuman, M., 2013, *Fabricated: The new world of 3D Printing*, John Wiley & Sons, Indianapolis.
- [25] Berman, B., 2012, "3-D printing: The new industrial revolution", *Business horizons*, **55**(2), pp. 155-162.
- [26] Spencer, S., 2012, *ZBrush Creature Design: Creating Dynamic Concept Imagery for Film and Games*, John Wiley & Sons, Indianapolis.
- [27] El Kady, A. M., Mahfouz, A. E., and Taher, M. F., 2010, "Mechanical design of an anthropomorphic prosthetic hand for shape memory alloy actuation," *In Biomedical Engineering Conference (CIBEC), 2010 5th Cairo International, IEEE*, pp. 86-89.
- [28] Autodesk, 2015, 3ds Max, <<http://www.autodesk.com/products/3ds-max/>>, retrieved on: Jun 1, 2015.
- [29] Autodesk, 2015, Maya, <<http://www.autodesk.com/products/maya/>>, retrieved on: Jun 1, 2015.
- [30] Blender, 2015, Blender, <<https://www.blender.org/>>, retrieved on: Jun 1, 2015.
- [31] Autodesk 123D, 2015, MeshMixer, <<http://www.meshmixer.com/>>, retrieved on: Jun 1, 2015.
- [32] Blain, J. M., 2014, *The Complete Guide to Blender Graphics: Computer Modeling and Animation*, CRC Press, Florida.

- [33] Murdock, K.L., 2014, Kelly L. Murdock's Autodesk 3ds Max 2015 Complete Reference Guide, SDC Publications, Kansas.
- [34] Harper, J., 2012, Mastering Autodesk 3ds Max 2013, John Wiley & Sons, Indianapolis.
- [35] Fisher, G., 2012, Blender 3D Basics, Packt Publishing Ltd, United Kingdom.
- [36] Hubeli, A. and Gross, M., 2000, A survey of surface representations for geometric modeling. Technical Report 335, ETH Zurich, Institute of Scientific Computing.
- [37] Stanculescu, L., Chaine, R., and Cani, M. -P., 2011, "Freestyle: Sculpting meshes with self-adaptive topology," *Computers & Graphics*, **35**(3), pp. 614-622.
- [38] Nealen, A., Sorkine, O., Alexa, M., and Cohen-Or, D., 2007, "A sketch-based interface for detail-preserving mesh editing," *ACM SIGGRAPH 2007 courses*, p. 42.
- [39] Fabio, R., 2003, "From point cloud to surface: the modeling and visualization problem," *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, **34**(5), W10.
- [40] Bommers, D., Lévy, B., Pietroni, N., Puppo, E., Silva, C., Tarini, M., and Zorin, D., 2013, "Quad-Mesh Generation and Processing: A Survey," *In Computer Graphics Forum*, **32**(6), pp. 51-76.
- [41] Scherer, M., 2011, ZBrush 4 Sculpting for Games: Beginner's Guide: Sculpt Machines, Environments, and Creatures for Your Game Development Projects, Packt Publishing Ltd, United Kingdom.
- [42] Lefebvre, P. P. and Lauwers, B., 2004, "STL model segmentation for multi-axis machining operations planning," *Computer-Aided Design and Applications*, **1**(1-4), pp. 277-284.
- [43] Sunil, V. B. and Pande, S. S., 2008, "Automatic recognition of features from freeform surface CAD models," *Computer-Aided Design*, **40** (4), pp. 502-517.
- [44] Xiao, D., Lin, H., Xian, C. and Gao, S., 2011, "CAD mesh model segmentation by clustering," *Computers & Graphics*, **35**(3), pp. 685-691.
- [45] Gao, S., Zhao, W., Lin, H., Yang, F. and Chen X., 2010, "Feature suppression based CAD mesh model simplification," *Computer-Aided Design*, **42**(12), pp. 1178-1188.
- [46] Shlafman, S., Tal, A. and Katz, S., 2002, "Metamorphosis of polyhedral surfaces using decomposition," *In: Proceedings of Eurographics 2002*, pp. 219-228.
- [47] Lavoué, G., Dupont, F. and Baskurt, A., 2004, "Curvature tensor based triangle mesh segmentation with boundary rectification," *In: Proceedings of IEEE Computer Graphics International*, pp. 10-17.
- [48] Shapira, L., Shamir, A. and Cohen-Or, D., 2008, "Consistent mesh partitioning and skeletonization using the shape diameter function," *The Visual Computer*, **24**(4), 249-259.
- [49] Cohen-Steiner, D., Alliez, P. and Desbrun, M., 2004, "Variational shape approximation," *ACM Transactions on Graphics (TOG)*, **23**(3), pp. 905-914.

- [50] Wu, J. H. and Kobbelt, L., 2005, "Structure recovery via hybrid variational surface approximation," *Computer Graphics Forum*, **24**(3), pp. 277-284.
- [51] Yang, C., Suzuki, H., Ohtake, H. and Michikawa, T., 2009, "Boundary smoothing for mesh segmentation," *In: Proceedings of the 11th IEEE International Conference on Computer-Aided Design and Computer Graphics*, pp. 241-248.
- [52] Reniers, D. and Telea, A., 2007, "Skeleton-based hierarchical shape segmentation," *In: Proceedings of the IEEE International Conference on Shape Modeling and Applications*, pp. 179-188.
- [53] Lai, Y., Hu, S., Martin, R. and Rosin, P., 2008, "Fast mesh segmentation using random walks," *In: Proceedings of the Symposium on Solid and Physical Modeling*, pp. 183-191.
- [54] Attene, M., Katz, S., Mortara, M., Patané, G., Spagnuolo, G. and Tal, A., 2006, "Mesh segmentation, a comparative study," *In: Proceedings of IEEE 2006 International Conference on Shape Modeling and Applications*, Japan, pp. 7-18.
- [55] Shamir, A., 2006, "Segmentation and shape extraction of 3D boundary meshes," *In: Proceedings Eurographics 2006*, pp. 137-149.
- [56] Agathos, A., Pratikakis, I., Perantonis, S., Sapidis, N. and Azariadis, P., 2007, "3D mesh segmentation methodologies for CAD applications," *Computer-Aided Design and Applications*, **4**(6), pp. 827-841.
- [57] Terek, Z. and Varady, T., 2005, "Digital shape reconstruction using a variety of local geometric filters," *In: Proceedings of the Third Hungarian Conference on Computer Graphics and Geometry*.
- [58] Attene, M., Falcidieno, B. and Spagnuolo, M., 2006, "Hierarchical mesh segmentation based on fitting primitives," *The Visual Computer*, **22**(3), pp. 181-193.
- [59] Mizoguchi, T., Date, H., Kanai, S. and Kishinami, T., 2006, "Segmentation of scanned mesh into analytic surfaces based on robust curvature estimation and region growing," *Geometric Modeling and Processing*, Springer, pp. 644-654.
- [60] Wang, J. and Yu, Z., 2011, "Surface feature based mesh segmentation," *Computers & Graphics*, **35**(3), pp. 661-667.
- [61] Yan, D. -M., Wang, W., Liu, Y. and Yang, Z., 2012, "Variational mesh segmentation via quadric surface fitting," *Computer-Aided Design*, **44**(11), pp. 1072-1082.
- [62] Lai, Y. K., Hu, S. M., Martin, R. R., and Rosin, P. L., 2009, "Rapid and effective segmentation of 3D models using random walks," *Computer Aided Geometric Design*, **26**(6), pp. 665-679.
- [63] Cai, Y. Y., Nee, A. Y. C. and Loh, H. T., 1996, "Geometric feature detection for reverse engineering using range imaging," *Journal of Visual Communication and Image Representation*, **7**(3), pp. 205-216.

- [64] Chen, Y. H. and Liu, C. Y., 1999, "Quadric surface extraction using genetic algorithm," *Computer-Aided Design*, **31**(2), pp. 101-110.
- [65] Alrashdan, A., Motavalli, S. and Fallahi, B., 2000, "Automatic segmentation of digitized data for reverse engineering applications," *IIE Transactions*, **32**(1), pp. 59-69.
- [66] Benko, P. and Varady, T., 2004, "Segmentation methods for smooth point regions of conventional engineering objects," *Computer-Aided Design*, **36**(6), pp. 511-523.
- [67] Chen, C. C. and Stamos, I., 2007, "Range image segmentation for modelling and object detection in urban scenes," *In: Proceedings of the Sixth International Conference on 3-D Digital Imaging and Modeling*, pp. 185-192.
- [68] Liu, Y. and Xiong, Y., 2008, "Automatic segmentation of unorganized noisy point clouds based on the Gaussian map," *Computer-Aided Design*, **40**(5), pp. 579-597.
- [69] Weber, C., Hahmann, S., and Hagen, H., 2010, "Sharp feature detection in point clouds," *In: Proceedings of the 2010 Shape Modeling International Conference*, pp. 175-186.
- [70] Langbein, F. C., Marshal, A. D., and Martin, R. R., 2004, "Choosing consistent constraints for beautification of reverse engineered geometric models," *Computer-Aided Design*, **36**(3), pp. 261-278.
- [71] Langbein, F. C., Gao, C. H., Mills, B. I., Marshall, A. D., and Martin, R. R., 2004, "Topological and geometric beautification of reverse engineering model," *In: Proc. ACM Symp. Solid Modelling and Applications 2004*, pp. 255-260.
- [72] Gao, C. H., Langbein, F. C., Marshall, A. D., and Martin, R. R., 2003, "Approximate congruence detection of model features for reverse engineering," *In: Proc. Int. Conf. Shape Modeling and Applications 2003*, pp. 69-77.
- [73] Li, Y., Wu, X., Chrysathou, Y., Sharf, A., Cohen-Or, D., and Mitra, N. J., 2011, "Globfit: Consistently fitting primitives by discovering global relations," *ACM Transactions on Graphics (TOG)*, **30**(4), pp. 52.2-52.11
- [74] Karniel, A., Belsky, Y., and Reich, Y., 2005, "Decomposing the problem of constrained surface fitting in reverse engineering," *Computer-Aided Design*, **37**(4), pp. 399-417.
- [75] Wang, J., Gu, D., Gao, Z., Yu, Z., Tan, C., and Zhou, L., 2013, "Feature-based solid model reconstruction," *Journal of Computing and Information Science in Engineering*, **13**(1), 011004.
- [76] Barbero, B. R., 2009, "The recovery of design intent in reverse engineering problems," *Computers & Industrial Engineering*, **56**(4), pp. 1265-1275.
- [77] Wang, J., Gu, D., Gao, Z., Yu, Z., Tan, C., and Zhou, L., 2012, "A framework for 3D model reconstruction in reverse engineering," *Computers & Industrial Engineering*, **63**(4), pp. 1189-1200.
- [78] Ke, Y., Fan, S., Zhu, W., Li, A., Liu, F., and Shi, X., 2006, "Feature-based reverse modeling strategies," *Computer-Aided Design*, **38**(5), pp. 485-506.

- [79] Mohanghegh, K., Sadeghi, M.H., and Abdullah, A., 2007, "Reverse engineering of turbine blades based on design intent," *The International Journal of Advanced Manufacturing Technology*, **32**(9-10), pp. 1009-1020.
- [80] Benko, P., Kos, G., Varady, T., Andor, L., and Martin, R., 2002, "Constrained fitting in reverse engineering," *Computer Aided Geometric Design*, **19**(3), pp. 173-205.
- [81] Benko, P., Martin, R., and Varady, T., 2001, "Algorithms for reverse engineering boundary representation models," *Computer-Aided Design*, **33**, pp. 839-851.
- [82] Belyaev, A., and Ohtake, Y., 2003, "A comparison of mesh smoothing methods," *In Israel-Korea Bi-national conference on geometric modeling and computer graphics*, vol. 2.
- [83] Song, X., and Juttler, B., 2009, "Modeling and 3D object reconstruction by implicitly defined surfaces with sharp features," *Computers & Graphics*, **33**(3), pp. 321-330.
- [84] Fan, H., Yu, Y., and Peng, Q., 2010, "Robust feature-preserving mesh denoising based on consistent subneighborhoods," *Visualization and Computer Graphics*, **16**(2), pp. 312-324.
- [85] Zhang, S., Huang, J., and Metaxas, D. N., 2011, "Robust mesh editing using Laplacian coordinates," *Graphical Models*, **73**(1), pp. 10-19.
- [86] Eyiurekli, M., and Breen, D., 2010, "Interactive free-form level-set surface-editing operators," *Computers & Graphics*, **34**(5), pp.621-638.
- [87] Huang, J., Chen, L., Liu, X., and Bao, H., 2009, "Efficient mesh deformation using tetrahedron control mesh," *Computer Aided Geometric Design*, **26**(6), pp. 617-626.
- [88] Mezger, J., Thomaszewski, B., Pabst, S., and StraBer, W., 2008, "Interactive physically-based shape editing," *Computer Aided Geometric Design*, **26**(6), pp. 680-694.
- [89] Masuda, H., Yoshioka., Y., and Furukawa, Y., 2007, "Preserving form features in interactive mesh deformation," *Computers-Aided Design*, **39**(5), pp. 361-368.
- [90] Masuda, H., Yoshioka., Y., and Furukawa, Y., 2006, "Interactive mesh deformation using equality-constrained least squares," *Computers & Graphics*, **30**(6), pp. 936-946.
- [91] Kazmi, I. K., You, L., Yang, X., Jin, X., and Zhang, J. J., 2015, "Efficient sketch-based creation of detailed character models through data-driven mesh deformations," *Computer Animation and Virtual Worlds*, **26**(3-4), pp. 469-481.
- [92] Liu, Y. J., Ma, C. X., and Zhang, D. L., 2009, "EasyToy: plush toy design using editable sketching curves," *IEEE Computer Graphics and Applications*, (2), pp. 49-57.
- [93] Olsen, L., Samavati, F. F., Sousa, M. C., and Jorge, J. A., 2009, "Sketch-based modeling: A survey," *Computers & Graphics*, **33**(1), pp. 85-103.
- [94] Dewaele, G., and Cani, M. -P., 2004, "Interactive global and local deformations for virtual clay," *Graphical Models*, **66**(6), pp. 352-369.
- [95] Ferley, E., Cani, M. -P., and Gascuel, J. -D., 2001, "Resolution adaptive volume sculpting," *Graphical Models*, **63**(6), pp. 459-478.

- [96] Alcaide-Marzal, J., Diego-Más, J. A., Asensio-Cuesta, S., and Piqueras-Fiszman, B., 2013, "An exploratory study on the use of digital sculpting in conceptual product design," *Design Studies*, **34**(2), pp. 264-284.
- [97] McCord, G., Wünsche, B., Plimmer, B., Gilbert, G., and Hirsch, C., 2008, "A Pen and Paper Metaphor for Orchid Modeling," *In GRAPP* (pp. 119-124).
- [98] Bettig, B., and Hoffmann, C. M., 2011, "Geometric constraint solving in parametric computer-aided design," *Journal of Computing and Information Science in Engineering*, **11**(2), 021001.
- [99] Date, H., and Onosato, M., 2008, "Triangular mesh deformation based on dimensions," *Computer-Aided Design and Applications*, **5**(1-4), pp. 287-295.
- [100] Xian, C., Gao, S., Lin, H., Liu, Y., and Xiao, D., 2009, "FEA-mesh editing with feature constrained," *In Proceedings of International Conference on Computer-Aided Design and Computer Graphics 2009*, pp. 340-343.
- [101] Sawai, Y., and Aoyama, H., 2010, "Efficient design method based on evaluation using CAE system," *Journal of Advanced Mechanical Design, Systems and Manufacturing*, **4**(5), pp. 827-837.
- [102] He, J., Zhang, C., Wei, Y., and Li, W., 2011, "Feature sensitive deformation for triangular mesh models," *Computer Animation and Virtual Worlds*, **22**(1), pp. 15-25.
- [103] Tang, W. -S., and Hui, K. C., 2013, "Deformation of objects with non-linear constraints," *International Journal of Computer Integrated Manufacturing*, **26**(10), pp. 928-938.
- [104] Gal, R., Sorkine, O., Mitra, N. J., and Cohen-Or, D., 2009, "iWIRES: an analyze-and-edit approach to shape manipulation," *ACM Transactions on Graphics (TOG)*, **28**(3), Article 33.
- [105] Sharf, A., Blumenkrants, M., Shamir, A., and Cohen-Or, D., 2006, "SnapPaste: An interactive technique for easy mesh composition," *The Visual Computer: International Journal of Computer Graphics*, **22**(9), pp. 835-844.
- [106] Meyer, M., Desbrun, M., Schröder, P. and Barr, A.H., 2002, "Discrete differential-geometry operators for triangulated 2-manifolds," *In: Proceedings of VisMath'02*, pp. 35-57.
- [107] Besl, P.J. and Jain, R.C., 1986, "Invariant Surface Characteristics for 3D Object Recognition in Range Images. Computer Vision," *Graphics and Image Processing*, **33**(1), pp. 33-80.
- [108] Demarsin, K., Vanderstraeten, D. and Roose, D., 2008, "Meshless extraction of closed feature lines using histogram thresholding," *Computer-Aided Design and Application*, **5**(5), pp. 589-600.
- [109] Sun, X., Rosin, P. L., Martin, R. R. and Langbein, F.C., 2009, "Noise analysis and synthesis for 3D laser depth scanners," *Graphical Models*, **71**(2), pp. 34-48.

- [110] Vollmer, J., Mencl, R. and Müller, H., 1999, "Improved Laplacian smoothing of noisy surface meshes," *Computer Graphics Forum*, **18**(3), pp. 131-138.
- [110] Pixologic, 2011, Sculpttris, <pixologic.com/sculpttris/>, retrieved on: January 22, 2014.
- [111] Lieu, D., and Sorby, S., 2009, *Visualization, Modeling, and Graphics for Engineering*, Cengage Learning, Delmar.
- [112] Hristake, V., 2008, *Basics of Solid Modeling: Your Guide to 3D*, Aurthor-House, Bloomington.
- [113] Shih, R. H., 2012, *AutoCAD 2013 Tutorial – Second Level: 3D Modeling*, SDC Publications, Kansas.
- [114] Chang, K. -H., *Product Design Modeling using CAD/CAE: The Computer Aided Engineering Design Series*, Elsevier, Oxford.
- [115] Besl, P. J., and Jain, R. C., 1986, "Invariant surface characteristics for 3D object recognition in range images," *Computer Vision, Graphics, and Image Processing*, **33**(1), pp. 33-80.
- [116] Pixologic, 2014, ZBrush, <pixologic.com/zbrush/>, retrieved on: June 2, 2015.
- [117] Autodesk, 2014, Mudbox, <www.autodesk.com/products/mudbox>, retrieved on: June 2, 2015.
- [118] Resolve, 2014, Cinema 4D, <resolve.ca/products/cinema-4d-r16/>, retrieved on: June 2, 2015.
- [119] Resolve, 2014, Modo, <resolve.ca/products/modo-901/>, retrieved on: June 2, 2015.
- [120] 3D Coat, 2015, 3D Coat, <3d-coat.com/>, retrieved on: June 2, 2015.
- [121] 3D Systems, 2014, Geomagic Sculpt, <www.geomagic.com/en/products/sculpt/overview/>, retrieved on: June 2, 2015.
- [122] Floater, M.S., and Hormann, K., 2005, "Advances in Multiresolution for Geometric Modelling," Chapter: *Surface Parameterization: A Tutorial and Survey*, Springer, pp. 157-186.
- [123] Isenburg, M., Gumhold, S., Gotsman, C., 2001, "Connectivity in shapes," *In: Proceedings of IEEE Visualization 2001*, pp. 135-142.
- [124] Groot, E., Wyvill, B., Barthe, L., Nasri, A., and Lalonde, P., 2014, "Implicit decals: Interactive editing of repetitive patterns on surfaces," *Computer Graphics Forum*, **33**(1), pp. 141-151.
- [125] Zhao, W., Gao, S., and Lin, H., 2007, "A robust hole-filling algorithm for triangular mesh," *The Visual Computer*, **23**(12), pp. 987-997.
- [126] Yoshizawa, S., Belyaev, A. G., and Seidel, H. -P., 2003, "Free-form skeleton-driven mesh deformations," *In: Proceedings of the 8th ACM Symposium on Solid Modeling and Applications*, pp. 247-253.

- [127] MeshLab, 2014, MeshLab, <meshlab.sourceforge.net/>, retrieved on: May 21, 2014.
- [128] Visual Computing Lab, 2014, Visualization and Computer Graphics Library, <<http://vcg.isti.cnr.it/vcglib/>>, retrieved on: June 2, 2015.
- [129] 3D Systems, 2010, Geomagic Studio 12, <www.geomagic.com/en/>, retrieved on: August 16, 2011.