

# A Thin-plate CAD Mesh Model Splitting Approach Based on Fitting Primitives

Chun Geng<sup>1</sup>, Hiromasa Suzuki<sup>1</sup>, Dong-Ming Yan<sup>2,3</sup>, Takashi Michikawa<sup>1</sup>, Yuichi Sato<sup>4</sup>, Masayoshi Hashima<sup>4</sup> and Eiji Ohta<sup>5</sup>

<sup>1</sup> The University of Tokyo, Tokyo, Japan

<sup>2</sup> Project Alice, INRIA, Villers les Nancy, France

<sup>3</sup> University of Hong Kong, Hong Kong, China

<sup>4</sup> Fujitsu Laboratories Ltd., Tokyo, Japan

<sup>5</sup> Fujitsu Advanced Technologies Ltd., Tokyo, Japan

---

## Abstract

*Extracting structural information from mesh models is crucial for Simulation Driven Design (SDD) in industrial applications. Focusing on thin-plate CAD mesh models (the most commonly used parts in electronic products such as PCs, mobile phones and so on), we present an algorithm based on primitive fitting for segmenting thin-plate CAD mesh models into parts of three different types, two of which are extruding surfaces and the other is a lateral surface. This method can be used for solid model reconstruction in the SDD process. Our approach involves two steps. First, a completely automatic method for accurate primitive fitting on CAD meshes is proposed based on the hierarchical primitive fitting framework. In the second step, a novel procedure is proposed for splitting thin-plate CAD mesh models by detecting parallel extruding surfaces and lateral surfaces. The method presented here has been proved to work smoothly in applications of real product design.*

Categories and Subject Descriptors (according to ACM CCS): I.3.6 [Computing Methodologies]: Computer Graphics Methodology and Techniques

---

## 1. Introduction

In industrial manufacturing, digital 3D models are widely used in product design as well as in product simulation and visualization. Two well-known 3D model representations are *Boundary Representation* (B-Rep) and tessellated meshes. B-Rep is an internal representation for most solid modelling systems, while tessellated meshes are always used in product simulation or visualization. Such meshes can be obtained using 3D digital scanners, and can also be generated directly from solid models. However, reverse conversion (i.e., from mesh to solid representation) is more difficult, even if the underlying solid model of the input mesh contains only simple primitives. The algorithm for such reverse processing may show characteristic changes resulting from input data differences between scanned meshes and meshes generated from solid models. Here we focus on the latter case for reconstructing special models with different levels of detail for CAE analysis.

Thin-plate CAD models are a special kind of models such as electromechanical components, sheet metals and other purposes. Such models can be formed by extrusion, bending or stamping, and their underlying surfaces mainly consist of simple primitive surfaces such as those found in PCs, mobile phones and so on.

In this paper, we propose an efficient framework to split thin-plate CAD mesh models into two extruding surfaces and a set of lateral surfaces. The extruding surfaces are fit using simple primitives. As an application for Simulation Driven Design (SDD), we present an algorithm that enables automatic solid model reconstruction from thin-plate CAD meshes by extruding one of the split extruding surfaces. Here, we consider situations where only tessellated mesh models are available rather than the original CAD models. Such situations are not rare in actual design processes in industry, as there are various reasons why it may not be possible to obtain the original models.

## 2. Related Work

As a key step in reverse engineering systems [APP\*07, LZHM06], mesh segmentation has received plenty of study over the last decade. The readers are referred to [AKM\*06, Sha08] for more details of the state-of-art mesh segmentation techniques. Here we focus on mesh segmentation for CAD models.

A top-down approach for primitive segmentation is provided by Benko et al. [BKV02], but the method is complicated for determining multiple surface regions and requires initial estimation, which may affect the segmentation results. Gelfand et al. [GG04] outlined a greedy clustering method for primitive segmentation using local slippage analysis. Although their approach is robust for scanned data, it is not suitable for models with low band meshes, such as those in our case.

Variational approaches have gained increasing popularity recently. Cohen-Steiner et al. [CSAD04] proposed a variational shape approximation (VSA) framework that approximates an input object with planar proxies. Wu and Kobbelt extended the VSA framework by adding spheres and cylinders as basic proxy types [WK05], while Yan et al. extended it using general quadric surfaces as basic fitting primitives [YLW06]. The VSA approach needs random (or manual) initial seed selection, which may affect the quality of the segmentation results.

One of the most efficient approaches for primitive fitting is the hierarchical face clustering (HFC)-based technique, which was first proposed by Garland et al. [GWH01] to cluster an input mesh into planar regions for use in computing radiosity. This technique (see Section 3.2) is simple and efficient. A mesh segmentation approach known as hierarchical mesh segmentation based on fitting primitives (HFP), contributed by Attene et al. [AFS06], is an extension of HFC. The HFP method is one of the few techniques that can be used to segment a surface into primitive surfaces for semantic interpretation purposes. HFC (HFP) is a suitable framework for mesh segmentation, but remains insufficient for a complete automatic processing in CAD mesh reconstruction for the following reasons:

1. The HFP approach does not produce optimal mesh segmentation results. The outcome is a sequence of clusters corresponding to the clustering level of hierarchical mesh segmentation, meaning that the user must interactively select the final result from the sequence.
2. It approximates only a narrow range of surfaces (planes, spheres and cylinders).
3. Due to the greedy nature of the algorithm, the clustering result for each level cannot be changed at a later level. Accumulated errors therefore grow fast, and the optimal result for primitives cannot be guaranteed.
4. Approximation errors are large in the early steps of clustering for primitive fitting because only a few samples are computed.

Some studies [Pra97, QS02, Qua08] focus on analyzing thin-plate model for creating mid-surface by Chordal Axis Transformation (CAT) approach, but little study discuss splitting thin-plate model.

In the next part of this paper, we outline the preliminaries in Section 3. The hybrid segmentation algorithm is presented in Section 4, and the primitive fitting algorithms are explained in Section 5. We introduce the thin-plate CAD model segmentation algorithm in Section 6. The experimental results are shown in Section 7, and we draw our conclusion in Section 8.

## 3. Preliminaries

We outline some preliminaries in this section.

### 3.1. Thin-plate Mesh Model

A thin-plate mesh model consists of top surfaces, bottom surfaces and lateral surfaces located between them. The term *extruding surface* refers to a top surface or a bottom surface. A top surface and a bottom surface in parallel are referred to as *parallel surfaces*, which may be found in planes, cylinders, tori, etc.

### 3.2. HFC Framework

Our segmentation algorithm is based on the original Hierarchical Face Clustering (HFC) approach [GWH01], which is briefly described here.

First, a dual graph of the input mesh is built, with each node corresponding to a triangle. There is a dual edge between two nodes if the corresponding triangles share a common edge. By considering each node in the dual graph as a representation of a cluster, two adjacent clusters can be merged into one by contracting the dual edge between them. Each dual edge is assigned a contraction cost, and a priority queue is then created in which all dual edges are sorted by their contraction cost. Finally, the process of merging sets of triangles into single representative clusters is iterated until the entire model becomes a single cluster. At each step, the dual edge with the lowest cost is popped out from the queue and contracted, and all the edges incident to the new representative node are updated. Different applications define various contraction costs [AFS06, AMSF08, GWH01, MK05].

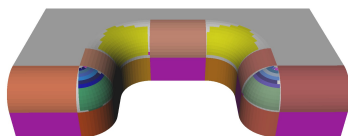
## 4. Multi-pass Primitive Fitting

In this section, we describe our CAD mesh segmentation algorithm based on primitives fitting. Although this algorithm is based on the HFP method [AFS06], in fact it is quite different. Firstly, we use a hybrid mesh segmentation framework involving a feature cutting approach and a heuristic quad clustering approach for preprocessing. Then, we propose an automatic step-by-step primitive fitting approach that improves primitives fitting quality.

#### 4.1. Preprocessing

There are three steps in preprocessing: normalization, feature detection and heuristic quad clustering.

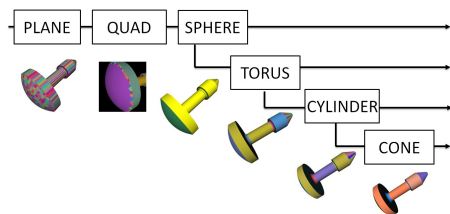
1. We first normalize the model into a unit cube  $[0, 1]^3$ . This step helps in the treatment of different models under the same standard of approximation error.
2. Then, we detect sharp features of the input model. Edges with a dihedral angle larger than a certain threshold ( $40^\circ$  in our experiments) are marked as features, and the mesh can be split into regions bounded by these features. This is widely used in many segmentation approach. [BMV01]
3. In the heuristic quad clustering approach, we collect paired similar triangles that can become quads and iteratively find neighbouring quads with the same dihedral angle. This works well for detecting primitives like spheres, cylinders and toru. Figure 1 shows the results of heuristic quad clustering.



**Figure 1:** Results of heuristic quad clustering. (purple:quad, gray:plane, orange:cylinder, blue:cone, green:sphere, yellow:torus)

#### 4.2. Step-by-Step Primitive Fitting

This is the main procedure of primitive fitting, which segments meshes to primitive surfaces. The step-by-step primitive fitting process is shown in Figure 2. In the process, we fit only one kind of primitive under a given fitting error.



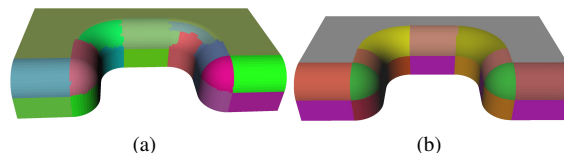
**Figure 2:** Step-by-step primitive fitting.

Our step-by-step primitive fitting approach is an improved version of the HFP method. There are three points to this process.

1. Contrasting the HFP approach in terms of the clustering results, our approach has only one result, while the HFP approach has serial results in a hierarchical structure.
2. This improvement makes the process automatically executable without interactive operation.

3. To obtain accurate results for primitive surfaces, we fit one type of primitive under a given fitting error in one HFC iteration. This helps to prevent two different kinds of primitives from being clustered in the early stages of clustering under an approximation tolerance.

Figure 3 shows different results from the original HFP and the step-by-step primitive fitting process.



**Figure 3:** (a) The clustering result of HFP with manual assignment for 22 uncompleted clusters. [AFS06], (b) our result with 23 clusters by automatic segmentation.

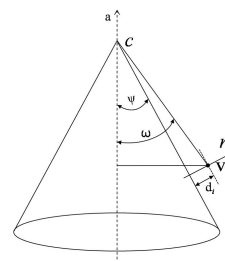
#### 5. Primitive Fitting

We use the fitting algorithm outlined in [AFS06] for planes, spheres and cylinders. The details are omitted here. We explain only the fitting algorithms for cones and tori in this section.

Suppose that the input of the fitting algorithm is a set of triangles  $\{t_i\}_{i=1}^m$ . The unit normal of each triangle  $t_i$  is denoted by  $\mathbf{n}_i$  and the area of  $t_i$  is denoted by  $|t_i|$ .

##### 5.1. Cone Fitting

A cone can be defined by an axis  $\mathbf{a}$ , an apex  $\mathbf{p}$  and an angle  $\psi$ . Assume that  $\bar{\mathbf{n}}$  is the average unit normal of all the triangles. We know that  $(\mathbf{n}_i - \bar{\mathbf{n}})^T \cdot \mathbf{a} = 0$  when we map all the  $\mathbf{n}_i$  to a Gaussian sphere. The axis  $\mathbf{a}$  can be computed as the eigenvector corresponding to the smallest eigenvalue of the covariance matrix  $\mathbf{N} \cdot \mathbf{N}^T$ , where  $\mathbf{N} = (\mathbf{n}_1 - \bar{\mathbf{n}}, \mathbf{n}_2 - \bar{\mathbf{n}}, \dots, \mathbf{n}_m - \bar{\mathbf{n}})$ . The apex  $\mathbf{p}$  of a polygonal cone surface can be simply determined using the QEM method [GH97]. The cone angle  $\psi$  can be calculated from  $\psi = \sum_{i=1}^n w_i$ , where  $w_i = \arcsin(|\mathbf{n}_i \cdot \mathbf{a}|)$ .



**Figure 4:** Illustration of the cone angle calculation.

The fitting error is computed using the following formula:

$$E = \sum_{i=1}^m |t_i| (d_{i_1}^2 + d_{i_2}^2 + d_{i_3}^2 + d_{i_c}^2) / \sum_{i=1}^m |t_i|, \quad (1)$$

where  $d_{i_1}, d_{i_2}, d_{i_3}$  and  $d_{i_c}$  are the distance from the three vertices and the barycenter of the triangle  $t_i$  to the fitting surface, respectively.

## 5.2. Torus Fitting

A torus can be defined by an axis  $\mathbf{d}$ , a centre  $\mathbf{q}$ , a ring radius  $R$  and a tube radius  $r$ .

We use *line geometry* [PW01] to estimate the axis of a torus. A *line element* consists of a line  $L$  in  $R^3$  and a point  $\mathbf{p}$  on it. It can be represented by *normalized Plucker coordinates*  $(\mathbf{l}, \bar{\mathbf{l}})$ , where  $\mathbf{l}$  is the unit vector parallel to  $L$ , and  $\bar{\mathbf{l}} = \mathbf{l} \times \mathbf{p}$  is the momentum vector of the line. A *linear complex*  $(\mathbf{c}, \bar{\mathbf{c}})$  of line elements is defined as the set of line elements whose Plucker coordinates  $(\mathbf{l}, \bar{\mathbf{l}})$  satisfy the linear equation:

$$\mathbf{c} \cdot \bar{\mathbf{l}} + \bar{\mathbf{c}} \cdot \mathbf{l} = 0$$

One of the propositions of a linear complex is that the normals of a given surface are contained in a linear complex  $(\mathbf{c}, \bar{\mathbf{c}})$  if (and only if) the surface is helical, rotational or cylindrical. Based on this proposition, we calculate the axis of the torus and a point on this axis.

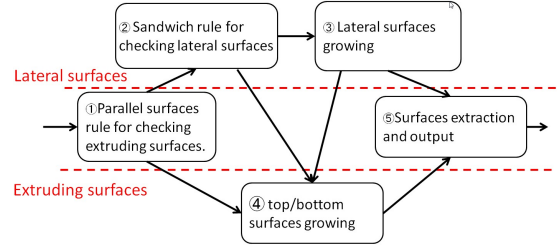
Assuming  $\mathbf{l}_i$  is the unit normal vector of point  $\mathbf{p}_i$  on the torus surface, we denote  $\bar{\mathbf{l}}_i = \mathbf{l}_i \times \mathbf{p}_i$ , and the error function is defined by:

$$F(\mathbf{c}, \bar{\mathbf{c}}) = \sum_i [\mathbf{c} \cdot \bar{\mathbf{l}}_i + \bar{\mathbf{c}} \cdot \mathbf{l}_i]^2. \quad (2)$$

We minimize the above energy function under the constrain  $\mathbf{c}^2 + \bar{\mathbf{c}}^2 = 1$ . The axis  $\mathbf{d}$  and any point  $\mathbf{o}$  on  $\mathbf{d}$  can be directly obtained from the minimizer  $(\mathbf{c}, \bar{\mathbf{c}})$ . We minimize the above energy function under the constraint  $\mathbf{c}^2 + \bar{\mathbf{c}}^2 = 1$ . Torus axis  $\mathbf{d}$  and any point  $\mathbf{o}$  on  $\mathbf{d}$  can be directly obtained from the minimizer  $(\mathbf{c}, \bar{\mathbf{c}})$ . We then project all the vertices on a reference plane defined by the axis, and fit a 2D circle onto the plane. The other parameters of the torus can be easily computed from the fitted 2D circle. The fitting error is computed using Eqn.1.

## 6. Thin-plate CAD Model Segmentation

Here we present the algorithm for splitting a thin-plate CAD mesh model. The input data for this process is the segmentation result obtained as described in Sec.4. There are three main steps to this algorithm. We first detect the parallel surfaces of the input model. Then, lateral surfaces between these parallel surfaces are extracted, and finally we group the lateral surfaces and extruding surfaces. Figure 5 shows a flowchart of the algorithm presented, and the details of each step are described below.



**Figure 5:** The thin-plate CAD model segmentation processes.

### 6.1. Determining Parallel Surfaces

We first estimate the thickness  $\delta$  of the thin-plate model by computing the distance between the two opposing planes with the largest area.

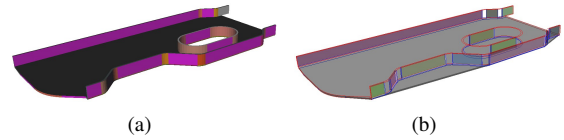
Next we group the primitives by type, i.e., planes, spheres, cylinders, etc. We use the following conditions to determine a pair of parallel planes.

- (a) Two normals of two planes are in opposing direction.
- (b) The distance between the planes is equal to  $\delta$ .
- (c) The planes are in a proper bounding box.

The parallel cylinders can be determined using the following criteria:

- (a) The cylinders have the same axes.
- (b) The radius difference between two cylinders is equal to  $\delta$ .

Parallel spheres, cones and tori can also be detected using similar rules. Figure 6 shows the detected parallel surfaces in a thin-plate mesh model.

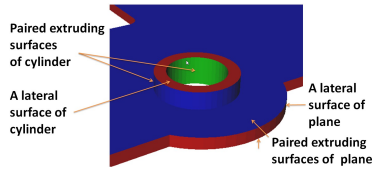


**Figure 6:** Parallel planes detection.

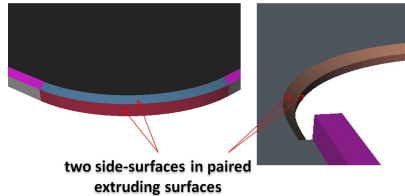
### 6.2. Determining Lateral Surfaces

From the dual graph, it is possible to detect the lateral surfaces between a pair of given parallel surfaces assumed to be extruding surfaces. Normally, a lateral surface is directly connected to a pair of extruding surfaces (see Figure 7).

In some cases, there may be more than two lateral surfaces between paired extruding surfaces (see Figure 8). Lateral surfaces in this case can also be detected from the dual graph using primitive relationships.



**Figure 7:** Lateral surfaces connected to parallel extruding surfaces.



**Figure 8:** Cases with more than two lateral surfaces connected to paired extruding surfaces.

### 6.3. Growing Lateral Surfaces and Extruding Surfaces

**Growing lateral surfaces.** Lateral surface growth is performed before extruding surface growth because the connections between the opposite extruding surfaces will be separated by the whole of the lateral surfacing.

As some seed lateral surfaces have already been detected, these are simply grown according to the primitive relationships in the dual graph to form all lateral surfaces using the following rules:

- (a) Growth is stopped when detected extruding surfaces are met.
- (b) Growth is stopped when detected lateral surfaces are met.
- (c) Surfaces connected to the seed lateral surfaces are grown from the dual graph.

**Growing extruding surfaces.** If the process of growing lateral surfaces works well, it is easy to grow the seeds of extruding surfaces detected in the last step from the dual graph.

## 7. Experimental Results

We tested more than 50 thin-plate CAD mesh models from industrial parts used in manufacturing product design. More than 90% of them were successfully split. Some selected examples are shown in Figure 9 and related statistics are given in Table 1. The results show that although some primitive fitting failed, the splitting of extruding surfaces and lateral surfaces was successful.

## 8. Conclusion

We have proposed a framework for splitting thin-plate CAD mesh models into two extruding surfaces and a set of lateral

**Table 1:** Primitive fitting result for Part A,B,C,D

	A	B	C	D
Planes	2	59	10	189
Quads	48	46	26	198
Spheres	-	-	-	3
Cylinders	21	50	36	207
Cones	-	1	-	-
Tori	-	2	8	6
Unknown surfaces	-	10	-	-

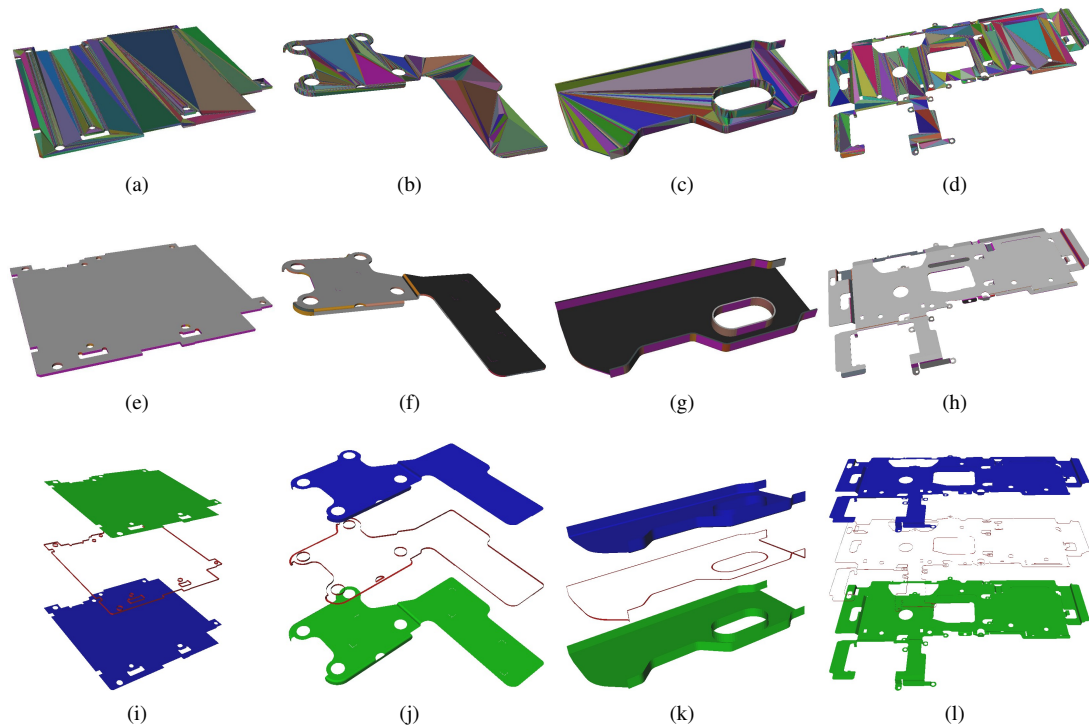
surfaces based on fitting primitives. We improved the original HFC and HFP approaches by using efficient preprocessing steps and introducing new fitting primitives, i.e., cones and tori. The experimental results obtained from testing real parts used in product design for manufacturing show that the algorithm works well. We plan to improve the primitive fitting algorithm in the next step. The processing of thin-plate models with non-uniform thicknesses is also an interesting direction for future investigation.

## Acknowledgements

The project for solid reconstruction from CAD meshes started in 2008 as a joint research program between the University of Tokyo and Fujitsu Laboratories Ltd. The project now continues with the support of the Japan Science and Technology Agency(JST)'s SEED-V program. Dong-Ming Yan is partially supported by the ANR-NSFC(60911130368) and European Research Council(GOODSHAPE FP7-ERC-StG-205693).

## References

- [AFS06] ATTENE M., FALCIDIENO B., SPAGNUOLO M.: Hierarchical mesh segmentation based on fitting primitives. *The Visual Computer* 22, 3 (2006), 181–193. 2, 3
- [AKM\*06] ATTENE M., KATZ S., MORTARA M., PATANE G., SPAGNUOLO M., TAL A.: Mesh segmentation - a comparative study. In *Proceedings of Shape Modelling International* (2006), pp. 14–25. 2
- [AMSF08] ATTENE M., MORTARA M., SPAGNUOLO M., FALCIDIENO B.: Hierarchical convex approximation of 3D shapes for fast region selection. *Computer Graphics Forum* 27, 5 (2008), 1323–1333. (Proc. SGP'08). 2
- [APP\*07] AGATHOS A., PRATIKAKIS I., PERANTORIS S., SAPIDIS N., AZARIADIS P.: 3D mesh segmentation methodologies for CAD applications. *Computer-Aided Design & Applications* 4, 6 (2007), 827–841. 2
- [BKV02] BENKO P., KOS G., VARADY T.: Constrained fitting in reverse engineering. *Computer-Aided Geometric Design* 19, 3 (2002), 173–205. 2



**Figure 9:** Top row: input thin-plate CAD models; mid row: segmentation results; bottom row: splitting Results.

- [BMV01] BENKO P., MARTIN R., VARADY T.: Algorithms for reverse engineering boundary representation models. *Computer-Aided Design* 33, 11 (2001), 839–851. [3](#)
- [CSAD04] COHEN-STEINER D., ALLIEZ P., DESBRUN M.: Variational shape approximation. *ACM Trans. on Graphics* 23, 3 (2004), 905–914. [2](#)
- [GG04] GELFAND N., GUIBAS L.: Shape segmentation using local slippage analysis. In *Proceedings of the Eurographics/ACM SIGGRAPH symposium on Geometry processing* (2004), pp. 214 – 223. [2](#)
- [GH97] GARLAND M., HECKBERT P.: Surface simplification using quadric error metrics. In *Proc. SIGGRAPH '97* (1997). [3](#)
- [GWH01] GARLAND M., WILLMOTT A., HECKBERT P.: Hierarchical face clustering on polygonal surfaces. In *Proceedings of the symposium on Interactive 3D graphics* (2001), pp. 49 – 58. [2](#)
- [LZHM06] LAI Y.-K., ZHOU Q.-Y., HU S.-M., MARTIN R. R.: Feature sensitive mesh segmentation. In *Symposium on Solid and Physical Modeling* (2006), pp. 17–25. [2](#)
- [MK05] MARINOV M., KOBELT L.: Automatic generation of structure preserving multiresolution models. *Computer Graphics Forum* 24, 3 (2005), 479–486. [2](#)
- [Pra97] PRASAD L.: Morphological analysis of shapes. *CNLS Newsletter* 139 (1997), 1–18. [2](#)
- [PW01] POTTMANN H., WALLNER J.: *Computational Line geometry*. Springer, 2001. [4](#)
- [QS02] QUADROS W., SHIMADA K.: Hex-layer: Layered all-hex mesh generation on thin section solids via chordal surface transformation. In *Proceedings of 11th International Meshing Roundtable* (2002), pp. 169–180. [2](#)
- [Qua08] QUADROS W.: An approach for extracting non-manifold mid-surfaces of thin-wall solids using chordal axis transform. *Engineering with Computers* 24, 3 (2008), 305–319. [2](#)
- [Sha08] SHAMIR A.: A survey on mesh segmentation techniques. *Computer Graphics Forum* 27, 6 (2008), 1539–1556. [2](#)
- [WK05] WU J., KOBELT L.: Structure recovery via hybrid variational surface approximation. *Computer Graphics Forum* 24, 3 (2005), 277–284. [2](#)
- [YLW06] YAN D.-M., LIU Y., WANG W.: Quadric surface extraction by variational shape approximation. In *Geometric Modeling and Processing* (2006), pp. 73–86. [2](#)