CAD model simplification using feature simplifications

Yeoun YOON* and Byung Chul KIM**

*ICT Convergence Technology Center, Samsung Heavy Industries Co., Ltd.,
80, Jangpyeong 3-ro, Geoje, 53261, Republic of Korea
**Department of Mechanical Engineering, Dong-A University,
37, Nakdong-daero 550beon-gil, Saha-gu, Busan, 49315, Republic of Korea
E-mail: mir7942@dau.ac.kr

Received 22 January 2016

Abstract

We propose an enhanced method for feature-based simplification of feature-based CAD models. In previous feature-based simplification methods, a CAD model is progressively simplified by suppressing individual features with low importance one at a time, whereas in the proposed method, each feature itself is progressively simplified based on its feature type. The proposed method provides a user with a further subdivided level of simplification. In the proposed method, features are categorized into sketch-based features and non-sketch-based features. The sketch-based features are simplified using three sketch simplification operations. The non-sketch-based features are simplified by converting them into simpler features, or changing their parameters. To demonstrate the proposed method, we present the implementation and experiment results.

Key words : Feature based simplification, Feature simplification, Sketch based feature, CAD model simplification

1. Introduction

Using a simplified CAD model is often useful during the design phase. For example, in engineering analysis, engineers try to reduce the total analysis time by employing a simplified CAD model that retains the important details and eliminates the irrelevant ones (Lee, 2005b; Thakur et al., 2009). In addition, a simplified CAD model can be used in other engineering activities such as network-based design (Chu et al., 2008) and virtual prototyping.

Many methods for automatically simplifying a CAD model have been proposed. Among them, feature-based simplification is known to be suitable for the aforementioned engineering activities (Lee, 2005a; Lee and Lee, 2012). In feature-based simplification, a CAD model is simplified by sequentially suppressing features of low importance. Therefore, the level of simplification of a CAD model is dependent on its number of features. The same shape can be modeled using a different number of features. If a CAD model is defined by a small number of features, even though it has a complex shape, the level of simplification that can be controlled by a user is limited. As a result, it is difficult to obtain a simplified CAD model suitable for a given user application.

To resolve this problem, we propose an enhanced feature-based simplification method called unit feature simplification, in which each individual feature is progressively simplified based on its feature type. For this purpose, features are categorized into sketch-based features and non-sketch-based features. The sketch-based features are simplified using three sketch simplification operations. The non-sketch-based features are simplified by converting them into simpler features, or by changing their parameters. In addition, we present a method for integrating unit feature simplification with the feature-based simplification framework.

Although Lee (2005b) showed that feature-based simplification can be used for engineering analysis, feature-based simplification methods (including the method proposed in this study) do not target a specific application. Instead, the proposed method gives the user a high degree of freedom for simplification, such that the proposed method can be applied to various applications.

The remainder of this paper is organized as follows. In Section 2, related studies and their limitations are
summarized. In Section 3, the proposed unit feature simplification method is described. In Section 4, the integration of unit feature simplification with feature-based simplification is explained, and experimental results are provided. In Section 5, the implementation result is demonstrated. In Section 6, a summary and directions for future work are presented.

2. Related works

Based on the representation of a CAD model, we classified CAD model simplification methods into mesh-based simplification methods, boundary representation model-based simplification methods, and feature-based simplification methods.

Mesh-based simplification methods (Schroeder et al., 1992; Hoppe et al., 1993; Rossignac and Borrel, 1993; Soucy and Laurendeau, 1996; Hoppe, 1996; Garland and Heckbert, 1997; Gao et al., 2010; Xiao et al., 2013), which are the most popular, simplify a CAD model, which is represented by a triangular mesh, by decimating triangles. Triangles are reduced by removing or collapsing edges or vertices. When a triangular mesh is regular and dense, or the shape is smooth, this method produces a good result. These triangular meshes are usually obtained from laser scanners or medical imaging devices. A CAD model, however, has sharp edges, and the triangle mesh it generates is not dense, and is irregular. When mesh-based methods are applied to a CAD model, the shape may be distorted (Gao et al., 2010).

In boundary representation model-based simplification methods, a CAD model is simplified using topological and geometrical information of a CAD model. There are dimension reduction methods (Rezayat, 1998; Donaghy et al., 2000; Sheen et al., 2010), which reduce the dimension of a shape, and feature removal methods (Venkataraman et al., 2002; Zhu and Menq, 2002; Sun et al., 2009), which remove the small or unnecessary features of a shape. In dimension reduction methods, thin-walled solids are converted into faces by extracting the mid-surfaces, and long cylindrical solids are converted into edges. In the feature removal methods, the features that do not greatly affect engineering analyses such as small fillets, small round sections, and small holes are removed. However, the boundary representation model-based methods can be applied only to a limited number of shapes, and it is difficult to progressively simplify a CAD model in this way.

In feature-based simplification methods (Lee et al., 2004; Lee, 2005a; Lee and Lee, 2012; Kang et al., 2014; Kim and Mun, 2014), a feature-based model is used as an input. A feature-based model is represented by an ordered set of features, called a modeling history. Each feature has a shape and an engineering meaning. The resultant shape of a feature-based model is obtained by sequentially applying Boolean operations to the previous resultant shape. An additive feature adds the feature shape to the previous resultant shape using the Boolean union operation (denoted by $\cup$), and a subtractive feature subtracts the feature shape from the previous resultant shape using the Boolean difference operation (denoted by $\setminus$). In feature-based methods, the shape of a CAD model is simplified by sequentially suppressing features with low importance.

Because the feature-based simplification methods use a feature as a simplification unit, the level of simplification greatly depends on the number of features. When a CAD model is designed using four features, the CAD model has four levels of simplification. When the same CAD model is designed using two features, the CAD model has two level of simplification.

Figure 1 shows feature-based simplification of a Y-shaped model, which consists of four features, as shown in Fig. 1(a). In Fig. 1, $F_0, F_1, F_2,$ and $F_3$ denote the features. Using feature-based simplification with a volume criterion as the feature importance, the Y-shaped model is progressively simplified as shown in Fig. 1(b). The simplified shapes in Fig. 1(b), however, are not satisfactory for two reasons. The first reason is that each feature has a complex shape such that the resultant shape is represented by only four features. As a result, the level of simplification of the Y-shaped model cannot be more than four. The second reason is that the majority of the features in the Y-shaped model are additive. Feature-based simplification is most effective when a CAD model has both additive features and subtractive features. In this paper, we focus on solving this problem, and propose unit feature simplification as an alternative.
3. Unit feature simplification

Figure 2 shows the procedures for feature-based simplification and unit feature simplification. In unit feature simplification, simplification of features is performed before feature rearrangement. The simplified features are used as an input of the framework in feature-based simplification.

![Diagram of feature-based simplification and unit feature simplification](image)
Fig. 3 Feature category. Features are categorized into sketch-based features and non-sketch-based features.

Based on a feature type, the different simplification strategies are applied to each feature. In this study, features are categorized into sketch-based features and non-sketch-based features. Figure 3 shows the feature category used in this study.

3.1 Simplification of sketch-based features

Sketch-based features are defined by sketches. There are two types of sketches: profile sketches and path sketches. Profile sketches are used to represent the sections of a feature shape, including one or more closed loops that consist of lines and circular arcs. When a closed loop is enclosed by another closed loop, the enclosed loop is called an inner loop, and the enclosing loop is called an outer loop. Inner loops form the removed parts of a feature shape. On the other hand, a path sketch is used to represent the path along which a profile sketch moves, and is represented by a single open loop.

Sketch-based features can be simplified by simplifying the sketches of the feature. In this study, we defined three sketch simplification operations, i.e., inner loop removal, concave vertex removal, and fillet/round removal, as shown in Fig. 4.

The inner loop removal operation finds and removes inner loops in a sketch. As a result, only the outer loops remain. The sketch in Fig. 4(a) has one outer loop and two inner loops. After applying the inner loop removal operation to the sketch in Fig. 4(a), the sketch in Fig. 4(b) is obtained.

The concave vertex removal operation removes concave vertices in a sketch. When the angle of two edges sharing a vertex is between 0 degrees and 180 degrees, the vertex is concave. When a concave vertex is removed, the neighboring edges are also removed. To form a closed loop, the neighboring edges of the removed edges are extended. The sketch in Fig. 4(a) has one concave vertex. After applying the concave vertex removal operation to the sketch in Fig. 4(a), the sketch in Fig. 4(c) is obtained.

The fillet/round section removal operation removes fillets or round sections that have a radius that is smaller than the user-specified radius. After a fillet or a round section is removed, its neighboring edges are extended to form a closed loop. The sketch in Fig. 4(a) has two round sections. After applying the fillet/round section removal operation to
the sketch in Fig. 4(a), the sketch in Fig. 4(d) is obtained.

The sketches of the sketch-based features are simplified by applying three sketch simplification operations. We describe the simplification strategies of each sketch-based feature in the following paragraph. Although commercial CAD systems provide various sketch-based features, we limit the scope to the most commonly used features, i.e., an extrusion feature, a revolution feature, a sweep feature, and a loft feature.

The extrusion feature and the revolution feature have one profile sketch. The sweep feature has one profile sketch and one path sketch. The loft feature has two or more profile sketches. Although a given loft feature can have several path sketches (called guide curves), for simplicity, we ignore them in our method. A profile sketch can be simplified by applying three sketch simplification operations. A path sketch is simplified by applying the fillet/round section removal operation. The profile sketch of a revolution feature is further simplified by filling a gap between the outer loop of the profile sketch and the revolution axis. The simplification operations are separately applied. Therefore, a sketch-based feature can have several simplified feature shapes. Table 1 shows sketch-based feature simplification examples.

Table 1 Simplification examples for sketch-based features.

<table>
<thead>
<tr>
<th>Original feature</th>
<th>Simplified feature 1</th>
<th>Simplified feature 2</th>
<th>Simplified feature 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extrusion feature</td>
<td>Inner loop removal</td>
<td>Concave vertex removal</td>
<td>Fillet/round section removal</td>
</tr>
<tr>
<td>Revolution feature</td>
<td>Concave vertex removal</td>
<td>Filling a gap between a profile sketch and an axis</td>
<td></td>
</tr>
<tr>
<td>Sweep feature</td>
<td>Concave vertex removal of a profile sketch</td>
<td>Fillet/round section removal of a path sketch</td>
<td></td>
</tr>
<tr>
<td>Loft feature</td>
<td>Concave vertex removal of profile sketches</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

3.2 Simplification of non-sketch-based features

Non-sketch-based features are simplified by converting them into simpler features, or changing their parameters. In this study, we consider only fillet features, pattern features, and hole features, as shown in Fig. 3. The proposed methods, however, can be easily extended to other type of features.

There are two types of fillet features: a constant radius fillet feature and a variable radius fillet feature. A variable radius fillet feature is defined by more than one radius. On the other hand, a constant radius fillet feature is defined by only one radius. Generally, a variable radius fillet feature has a more complex feature shape than a constant radius fillet. When simplifying fillet features, a variable radius fillet feature is converted into a constant radius fillet feature.
radii of the variable radius fillet feature are averaged.

Pattern features are defined by a reference feature and a number of instances. The reference feature is simplified using the methods proposed in this section. In addition, pattern features are simplified by reducing the number of instances.

We classify hole features into simple hole features, which have a cylindrical shape only, and compound hole features such as counter-bored holes, tapered holes, and so on. Generally, compound hole features have more complex feature shapes than a simple hole feature has. When simplifying hole features, compound hole features are converted into simple hole features. Table 2 shows simplification examples of the non-sketch-based features.

Modification features such as fillet features, draft features, and shell features can sometimes cause errors after changing the feature parameter values. These errors occur because the changed features are not inconsistent with neighboring shapes. In this study, we did not automatically treat such cases. Instead, user interaction is needed to recover from the error.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Original feature</th>
<th>Simplified feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fillet feature</td>
<td>Variable radius fillet</td>
<td>Constant radius fillet</td>
</tr>
<tr>
<td>Pattern feature</td>
<td>Circular pattern with 8 instances</td>
<td>Circular pattern with 4 instances</td>
</tr>
<tr>
<td>Hole feature</td>
<td>Counter-bored hole</td>
<td>Simple hole</td>
</tr>
</tbody>
</table>

4. Integration with feature-based simplification

In this section, we describe how to use the unit feature simplification method in the framework of feature-based simplification. Each feature of a CAD model is simplified using the feature simplification operations in Section 3. The feature simplification operations are sequentially applied to each feature. Because the simplified feature shapes are to be rearranged at the rearrangement stage, the sequence of the operations is not important. When the simplified feature shapes are placed in reverse order, we obtain the subdivided modeling history of a feature. For example, feature $F_0$ in Fig. 1(a) is simplified by sequentially applying three sketch simplification operations, as shown in Fig. 5(a). The three simplified shapes of feature $F_0$ are placed in reverse order, as shown in Fig. 5(b). In Fig. 5(b), $F_0^{(0)}$, $F_0^{(1)}$, $F_0^{(2)}$, and $F_0^{(3)}$ are the delta volumes of three simplified shapes of feature $F_0$, which are called sub-features of $F_0$. When the same process is applied to every feature in Fig. 1, the modeling history of the Y-shaped model is presented, as shown in Fig. 6.
Fig. 5 (a) Simplified shapes of feature $F_0$ in the Y-shaped model and (b) its subdivided modeling history of it.

Fig. 6 Subdivided modeling history of the Y-shaped model.

The sub-features in the subdivided modeling history are re-ordered based on their importance. The importance of a sub-feature is dependent on the object’s applications. Lee (2005a), Kang et al. (2014), and Kim and Mun (2014) presented several rearrangement strategies. In this study, the importance of a sub-feature is evaluated based on the following criterion; 1) additive sub-features are given higher importance than subtractive sub-features, and then 2) a feature with a larger volume has higher importance than a feature with a smaller volume.

Because Boolean subtraction is not commutative, the resultant shape after rearranging the sub-features may be changed. For example, if the order of $F_1^{(1)}$ and $F_2^{(0)}$ in Fig. 6 is exchanged, the resultant shape is changed, because

$$S - F_1^{(1)} \cup F_2^{(0)} \neq S \cup F_2^{(0)} - F_1^{(1)}$$

(1)

where $S$ is the result of the step from $F_0^{(0)}$ to $F_1^{(0)}$. To resolve this problem, Lee (2005a) proposed the concept of the effective volume of a feature. In summary, if the order of two features $F_1$ and $F_2$ is exchanged, then the result of exchanging two Boolean operations can be described as follows:

$$F_0 \cup F_1 - F_2 = F_0 - F_2 \cup F_1' = F_0 - F_2 \cup (F_1 - F_2)$$

(2)

$$F_0 - F_1 \cup F_2 = F_0 \cup F_2' - F_1 = F_0 \cup F_2 - (F_1 - F_2)$$

(3)

According to Eq. (3), Eq. (1) should be expressed as

$$S - F_1^{(1)} \cup F_2^{(0)} = S \cup F_2^{(0)} - (F_1^{(1)} - F_2^{(0)})$$

(4)
That is, instead of $F_1^{(1)}$, $(F_1^{(1)} - F_2^{(0)})$, which is the effective volume of $F_1^{(1)}$, should be subtracted. Figure 7 shows the modeling history after rearranging the sub-features of the Y-shaped model.

After rearranging the sub-features, sub-features with low importance are suppressed. Sequentially suppressing the sub-features with low importance generates a progressively simplified CAD model shape. Figure 8 shows the simplified shapes of the Y-shaped model.

Figure 9 shows another example, which are simplified shapes of the cover model.
As shown in Fig. 8 and Fig. 9(b), the level of simplification (LOS) increased, as compared to Fig. 1(b) and Fig. 9(a), respectively. Using unit feature simplification, a user has more control over the simplification. In Fig. 1(b), a user can have four LOSs, whereas a user can have 9 LOS in Fig. 8. In Fig. 9(a), a user can have 7 LOSs, whereas a user can have 10 LOSs in Fig. 9(b). In addition, unit feature simplification provides more naturally simplified shapes than feature-based simplification.

Figure 10 shows the simplification of the engine block model, which has 129 LOS. Until 91 LOS, the fillets are removed. Until 73 LOS, the small inner loops are removed. At 70 and 43 LOSs, concave shapes are removed. As can be seen in Fig. 10, the proposed method can be applied to practical models.

When the shape is progressively simplified, dramatic changes occur at low LOSs, as shown in Fig. 8 and 9. Because the shape at low LOS is significantly different from the original shape, it is seldom used. Instead, a user usually selects shapes at high LOS.

5. Implementation

To verify the unit feature simplification, we implemented a prototype program. The program was implemented using C++, the SolidWorks API, the ACIS geometric modeling kernel, and the HOOPS 3D visualization library. Input CAD models used for the experiments were modeled in SolidWorks, and the SolidWorks API was used to obtain the modeling history and feature information for the model. ACIS was used to process the feature shapes, and HOOPS 3D was used to visualize the shapes. Figure 10 shows a screenshot of the prototype program.

6. Conclusion

In this paper, we proposed the unit feature simplification method to enhance the feature-based simplification method. In unit feature simplification, features are categorized into sketch-based features and non-sketch-based features. Sketch-based features are simplified using three sketch simplification operations. Non-sketch-based features are simplified by converting them into simpler features, or changing their parameters. In addition, we presented a
method for integrating unit feature simplification with the framework of feature-based simplification. Compared to feature-based simplification, unit feature simplification can generate simpler CAD model shapes. The proposed method provides a designer with a wider array of options for choosing simplified shapes. In addition, this method provides more naturally simplified shapes.

However, we restricted the scope of this investigation to the features listed in Fig. 3. Because commercial CAD systems provide more features, it is necessary to extend the proposed method to other features. In addition, a formal representation and mathematical proof of the proposed method are also needed.

Fig. 10 Simplified shapes of the engine block model.
Fig. 11 Screenshot of the prototype program and the engine block model.

Acknowledgement

This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (NRF-2014R1A1A1006181) of the Korean Government and the Plant Research Program funded by the Land, Infrastructure and Transport (14IFIP-B091004-01) of the Korean Government.

References


Kang, Y., Kim, B. C., Mun, D. and Han, S., Method to simplify ship outfitting and offshore plant equipment three-dimensional (3-D) computer-aided design (CAD) data for construction of equipment catalog, Journal of Marine Science and Technology, Vol.19, No.2 (2014), pp.185–196.


Lee, S. H., A CAD-CAE integration approach using feature-based multiresolution and multi-abstraction modelling