Int J Interact Des Manuf
DOI 10.1007/s12008-012-0145-0

ORIGINAL PAPER

Appearance preserving simplification of 3D CAD model with large-scale assembly structures

Satoshi Kanai · Daisuke Iyoda · Yui Endo · Hideki Sakamoto · Naoki Kanatani

Received: 16 May 2011 / Accepted: 20 January 2012

Abstract Currently commercial 3D CAD systems which had only been used in upper stage of design scenes become widely used in lower stages such as rough sales purposes, model exporting to external users or e-commerce on Web. Usually, in the design stage, solid models even of very fine parts or full-detailed shapes have been built in the CAD systems, and the assembly models tend to have a huge number of parts and very complex inner structures. Moreover, for achieving the light-weight and strengthened parts, the inner structures of the housing such as ribs or bosses have had very complex geometries. However, when they are used for browsing, styling review and sales purposes, there is hardly the occasion where is full-detailed assembly models are required, and the primary purpose of the systems is often to fast render external shapes rather than to render detailed inner structures. Appearance preserving simplification of large scale assembly model available to the commercial 3D CAD systems is strongly needed for these purposes. Therefore, this paper proposes several appearance preserving simplification methods of 3D CAD model with large-scale assembly structures. Three simplification methods are proposed in the paper; (1) only by removing invisible parts from the assembly, (2) by removing both invisible form features from the part surface and invisible parts themselves from the assembly, and (3) by removing both form features and parts which are invisible even when position and orientation of movable parts change in the assembly. Our methods are based on an algorithm which can directly detect invisible parts or features by pre-rendering the models from multiple view directions and reading the rendered results from the frame buffer. Our algorithm can be carried out regardless of CAD systems. Thanks to using the current GPU, invisible parts or features detection is robust and fast in the algorithm. If needed, geometric dependency among the features in the assembly can be kept even in the simplification. The performances of these simplification methods in model size reduction and the processing time are examined.

Keywords Model simplification · CAD · Geometric modeling · Assembly model · Solid model · Form feature · Dimensional constraints

S. Kanai (✉) · D. Iyoda
Graduate School of Information Science and Technology,
Hokkaido University, Kita-14, Nishi-9, Sapporo, 060-0814, Japan
e-mail: kanai@ssi.ist.hokudai.ac.jp

Y. Endo
Digital Human Research Centre, Advanced Industrial and Technology, 2-3-26 Aoumi, Kotoku-ku, Tokyo, 135-0064, Japan
e-mail: y.endo@aist.go.jp

H. Sakamoto · N. Kanatani
Kozo Keikaku Engineering Inc., 4-38-13 Honcho,
Nakano-ku, Tokyo, 164-0012, Japan
e-mail: sakamoto@kke.co.jp

N. Kanatani
e-mail: kanatani@kke.co.jp

1 Introduction

1.1 Background

Nowadays commercial 3D CAD systems are being widely used in product design and manufacturing stages at various industries, and large-scale solid models with complex assembly structures have been built in the CAD systems. Assembly models having more than several thousand parts are even modeled in the systems for designing automotive components, power electric facilities, plant equipments, industrial machines and medical equipments. Moreover, even in middle or small-scale products, the solid models even of very fine

Published online: 08 February 2012

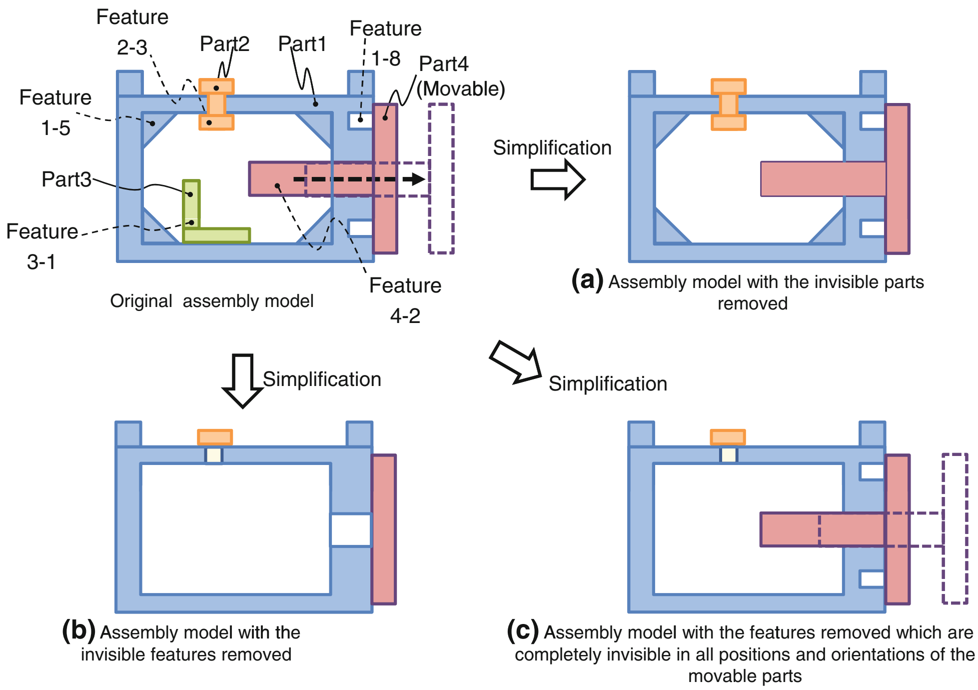


Fig. 1 Simplification methods of large-scale assembly model

parts or full-detailed shapes have been built as parts of the assembly model in the CAD systems in order to accurately carry out mass property calculation or clearance check and to create BOMs without omissions. As a result, the assembly models have had a large number of parts and very complex inner structures. Moreover, for achieving the light-weight and strengthened parts, the inner structures of the housing model have had very complex geometries such as a large set of bosses, ribs, cuts and etc. in case of the products made of injection molding, die casting or cutting.

On the other hand, along with improved performance of computers, the CAD systems have even started being used in processes other than design or manufacturing stages; styling review, sales purpose, model exporting to external users or e-commerce on Web. However, when these large-scale assembly models with huge number of parts or complex geometries are used for browsing, styling review and sales purposes, there is hardly the occasion where the full-detailed assembly models are required, and their primary application purpose is to fast render external shapes rather than to render detailed inner structures. Therefore, appearance preserving simplification of large scale assembly model is strongly needed for these purposes.

The requirements for the appearance preserving simplification of a large-scale assembly model can be considered as the followings;

- The format of a simplified assembly model should be identical to the one originally modeled in a commercial CAD system for avoiding unstable data-format transformation processes and for the sake of users' convenience.
- As inside of part has many complex form features, the simplification should perform not only part-by-part elimination of invisible parts (Fig. 1a) but feature-by-feature elimination of invisible form features (Fig. 1b).
- As shown in Fig. 1c, as the visibility of the form features and the parts could change when the position and the orientation of movable parts change in the assembly, the simplification method should not remove too much features and parts which could be exposed to outside at particular parts' positions and orientations.
- The simplified assembly model should still work as the original one. If some parts or features are invisible but are needed for defining other visible feature geometries or other geometric constraints among visible parts, they should not be removed in the simplification.
- The simplification method should have robustness, high-speed performance and portability for various commercial CAD systems.

1.2 Related works

So far several simplification methods of 3D geometric model have been proposed. These techniques are classified roughly into mesh model simplification and feature-based solid model simplification.

The former techniques gradually reduce the complexity of a triangular mesh, for example, by using *edge-collapse*

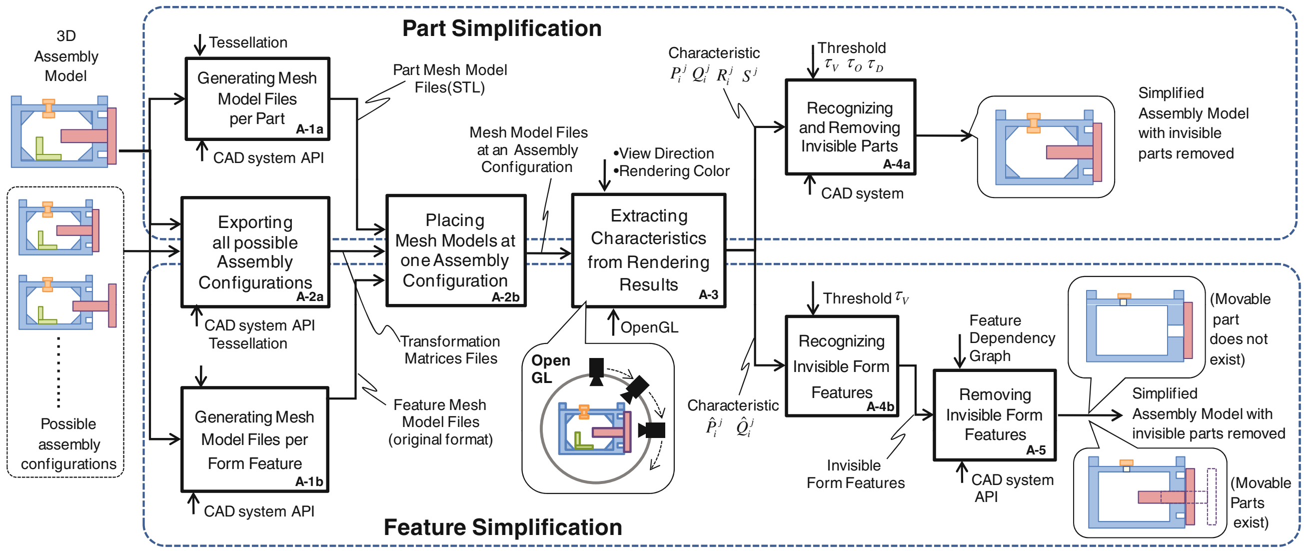


Fig. 2 Overview of the two appearance preserving simplification methods

[1], *vertex-pair-collapse* [2] or *vertex-removal* [3] operations. Topology-changing simplification of triangular mesh model was also proposed using *alpha-hull* over polygonal objects [4]. Moreover, an adaptive real-time level-of-detail control algorithm of triangular mesh model was proposed where the model simplifications were dependent on view direction, lighting and visibility [5]. The detailed and comprehensive review of the simplification algorithms of triangular mesh models was already published as [6]. However, all of these simplification techniques did not aim for simplification of feature-based solid model of commercial CAD systems.

While as for the latter techniques, Seo et.al [7] proposed wrap-around operation to simplify B-rep solid model. In this technique, subtractive form features such as holes or cut-off regions are first recognized as detailed shape, and they are removed from the original model according to given a LOD. The shape of the simplified model is just like convex-hull shape of the object. Also, Lee [8] proposed feature-based multi-resolution modeling of solid model. In this technique, effective volumes which are volume of features affecting model shape are recognized first and then they are arbitrarily rearranged according to a LOD criterion to generate the simplified solid model. Even, these methods can be applied to an assembly model, but they did not aim for appearance preserving simplification of assembly models. Moreover, the method of Lee [8], which adopted non-manifold topology structure, cannot be compatible with commercial 3D CAD systems.

On the other hand, as the other simplification technique, Lattice technology Inc. proposed the one which can convert CAD model into unique light-weight model format called XVL [9]. It can reduce the data amount of the solid model about one hundred times from that of the original one by approximating the model by smooth rounded surfaces. However, for this reason, the simplified CAD model format has been changed to the one completely different from the original, and they cannot be viewed or modified in the original CAD system.

Moreover, all of these related works described above did not discuss how to efficiently detect invisible parts and invisible form features from the outside in assembly model automatically.

1.3 Purpose and algorithm overview

In this paper, as shown in Fig. 2, two appearance preserving simplification methods are proposed where they remove invisible parts or invisible form features from assembly models in commercial 3D CAD systems. Hereafter, *part simplification* refers to the simplification method by removing invisible parts, and *feature simplification* refers to the one by removing invisible form features. Pre-rendering the models from multiple view directions and reading the rendered results from frame buffer are the basic idea of invisible part and feature detection in the methods. The core simplification algorithm is common in two methods.

Moreover, as an optional function, the methods can handle the case where the assembly model has some movable parts and the visibility of the form features and the parts could change depending on their positions and orientations in the assembly. In this case, as a result of the simplification, the methods correctly keep parts or features which are invisible at some positions and orientations but are visible at the other ones.

As for processes of the algorithm, firstly, triangular mesh models per part or form feature are generated by faceting the surfaces of a 3D assembly model and are outputted to separate mesh model files from a commercial 3D CAD system (A-1.a), (A-1.b).

If the model has some movable parts, the possible set of their relative positions and orientations (assembly configurations) are also exported from the CAD system (A-2.a). And each of the mesh models is respectively placed at its right positions and orientations to express an assembly at a particular configuration (A-2.b).

Next, in order to recognize invisible parts or invisible form features, each part or form feature is rendered by a different color from multiple view directions using OpenGL functions, and then four characteristics representing rendered results are extracted from the frame buffer (A-3).

Finally, three evaluation indices are evaluated per part or form feature to finally determine which of them are invisible according to user-specified thresholds (A-4.a), (A-4.b). In case of part simplification, the invisible parts are manually removed from the original assembly model in commercial CAD system (A-4.a).

While in case of the feature simplification, the invisible form features does not necessarily become features to be removed. Invisible form features which are not needed for defining visible form feature geometries can only be removed. This test is performed by API of the CAD system and the unwanted form features can be automatically removed from each part geometry of the assembly model (A-5). In this study, the proposed methods are implemented as add-on C++ software for a commercial 3D CAD system (SolidWorks2008).

In the following sections of this paper, Sect. 2 describes the part simplification method, and Sect. 3 does the feature simplification method. Section 4 described the feature simplification method in case that some parts are movable in the assembly model and the visibility of the parts and the features changes depending on the assembly configurations. Section 5 presents the simplification results of above three cases, and their performances are compared. And Sect. 6 presents conclusions.

2 Part simplification method

2.1 Generating mesh model per part

In order to detect invisible parts, the proposed simplification algorithm uses the graphic rendering result of 3D assembly model. As it is difficult to directly access to the frame buffer during running a CAD system, pre-rendering of the model is carried out on another viewport by our developed OpenGL application software.

For this reason, mesh models per part in the assembly are generated by tessellating surfaces of 3D CAD model and are outputted to STL format files called *part mesh model files*. This operation can be easily carried out using file exporting function which commercial CAD systems almost have.

At the same time, each part in the assembly model is given an identification number having 24-bit integer value which enables the simplification algorithm to differentiate about 16 million parts.

2.2 Extracting characteristics from rendering results

To detect invisible parts, four characteristics are evaluated from the results of mesh model rendering from multiple view directions. In this subsection, the procedure of the rendering and the evaluation of characteristics are shown.

2.2.1 View point setting and color allocation

As the pre-process of mesh model rendering, a set of view points and directions are specified and a unique rendering color is assigned to each part model.

In each view point, whole assembly model has to be rendered on the fixed size of frame buffer (in our case, 640 × 640 pixels). Therefore, as shown in Fig. 3a, the rendering scale factor is automatically determined as L_V/L_B where L_V denotes the height and width of the viewport of the frame buffer, and L_B denotes the diagonal length of the axis-aligned bounding-box of the assembly model. This scaling enables all of the rendering results from any view point to be fit into full viewport area as large as possible and without loss of rendering.

As shown in Fig. 3b, the view points are placed at the vertices of polyhedron approximating a sphere, and the point of gaze is located at the sphere center which is identical to the centroid of the assembly model. In this study, as shown in Fig. 3b, 26 view points are used for the rendering.

When rendering the mesh model, 24-bit RGB value converted from a 24-bit identification integer is allocated to each part as the rendering color.

In this way, if we read the RGB values of the pixels from the frame buffer which stores rendering results, we can detect whether or not a part is visible in a simple way.

2.2.2 Extracting characteristics of rendering results

The rendering of mesh models carried out the following two times in each view; the rendering of each part model alone and the rendering of whole assembly model. Then, as shown in Figs. 4 and 5, the following four characteristics are evaluated from the RGB values of the pixels left in the frame buffer by calling *glReadPixels* function of OpenGL.

(1) P_i^j: the number of pixels with ith part's color when ith part is rendered alone at jth view, (Fig. 4a)

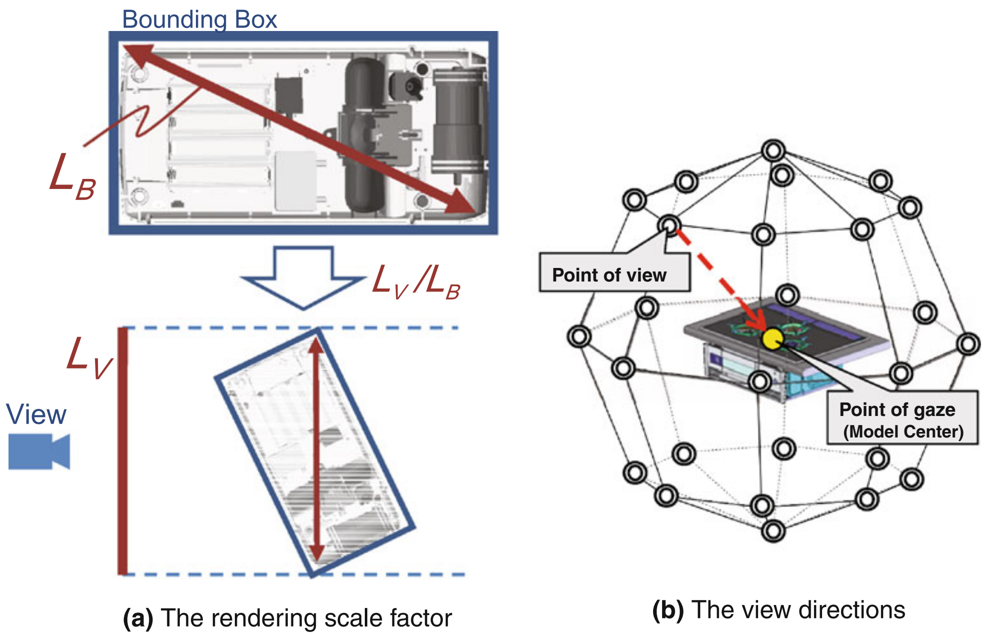


Fig. 3 The pre-process of rendering

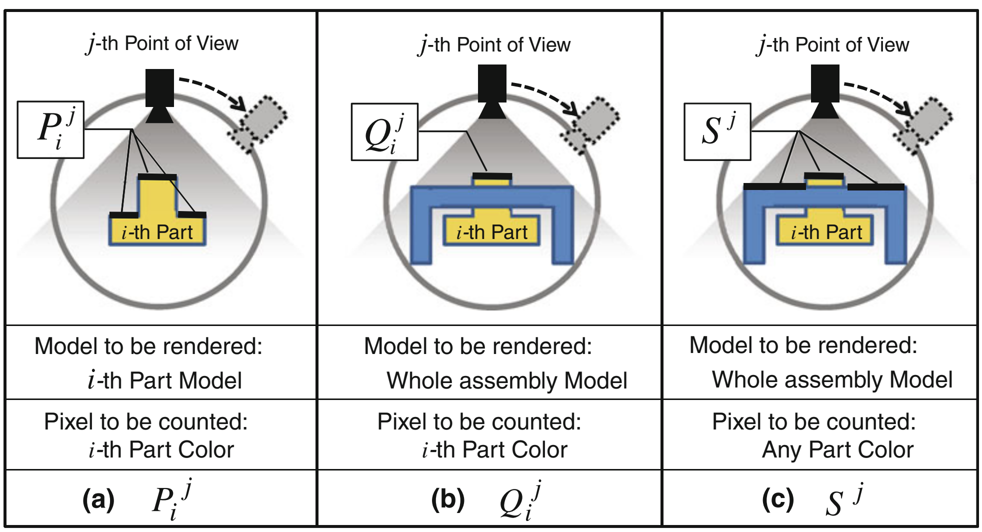


Fig. 4 Three characteristics of rendering results (P_i^j, Q_i^j, S^j)

(2) Q_i^j : the number of pixels with ith part's color when whole assembly model is rendered at jth view, (Fig. 4b)
(3) S^j: the number of pixels with any part's color when whole assembly model is rendered at jth view, (Fig. 4c)
(4) R_i^j: the largest number among the number of pixels with ith part's color at each depth level, when whole assembly model is rendered at jth view (Fig. 5).

As shown in Fig. 5, a *depth-level* means an interval generated by dividing the depth range from d_{min}^j to $d_{min}^j + d_{lim}$ by specified interval number N_d, where d_{min}^j is the nearest depth value of the assembly model at jth view, and d_{lim} is the user-specified depth-distance. The depth value has been normalized between 0.0 and 1.0. The number of rendered pixels at a specified depth-level can be extracted by calling glReadPixels. In this study, $d_{lim} = 0.05$ and $N_d = 20$ are used.

2.3 Recognition and removal of invisible parts

Using the four characteristics discussed above, the following three evaluation indices defined by equations (1), (2) and (3) are calculated for ith part to determine the visibility of the each part in the assembly model.

(a) The maximum exposure rate of ith part among all view,

$$F_i^{visible} = \max(Q_i^j / P_i^j) \quad (1)$$

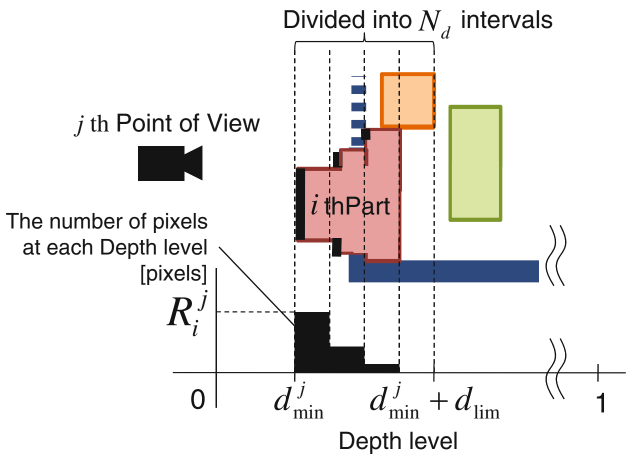


Fig. 5 A characteristic of rendering results (R_i^j) which indicates the number of pixels at each depth level

(b) The maximum occupation rate of ith part among all view,

$$F_i^{occupied} = \max(Q_i^j / S^j) \tag{2}$$

(c) The maximum contribution of ith part to exterior shape of assembly,

$$F_i^{depth} = \max(R_i^j) \tag{3}$$

Consequently, using these evaluation indices and user-specified thresholds $\tau_V (0 \leq \tau_V \leq 1)$, $\tau_O (0 \leq \tau_O \leq 1)$ and $\tau_D (0 \leq \tau_D \leq 640^2)$, the part which satisfies the following condition (4) is recognized as completely or nearly "invisible".

$$(F_i^{visible} = 0) \vee \left\{ (F_i^{visible} < \tau_V) \wedge (F_i^{occupied} < \tau_O) \wedge (F_i^{depth} < \tau_D) \right\} \tag{4}$$

The condition $F_i^{visible} = 0$ means that ith part is completely invisible parts to be removed. The condition $F_i^{Visible} < \tau_V$ means that the part is determined to be "nearly invisible" where the ratio of exposure of the part rendered along with the assembly to the one rendered in the part alone is less than τ_V, and it should be removed. The condition $F_i^{occupied} < \tau_O$ means that the part is determined to be "nearly invisible" where the ratio of exposure of the part rendered along with the assembly to the exposure of whole assembly is less than τ_O, and it should be removed. The condition $F_i^{depth} < \tau_D$ means that the parts located near the exposed surface of the assembly but having a small exposure less than τ_D should be removed.

Finally, by actually deleting the invisible parts and the nearly invisible parts from the assembly model, we can obtain simplified model. At the moment, in the case of part simplification, eliminating task is manually operated, which is the engineering pending work.

By adjusting these thresholds τ_V, τ_O and τ_D appropriately, the simplified 3D assembly model can be generated where different numbers of nearly invisible parts are eliminated and the rate of model reduction can be controlled.

3 Feature simplification method

In this section, the simplification method by removing invisible form features is described. The fundamental algorithm is as same as the one of the part simplification, so the processes which are fundamentally different from the part simplification are only explained in the section.

3.1 Generating mesh model per form feature

Mesh models per form feature (*feature mesh model files*) are generated by faceting boundary faces which make up each form feature. Because generally there is no standard function for exporting a triangular mesh model per form feature in commercial CAD systems, as shown in Fig. 6, we developed special C++ program for exhaustively retrieving each features from the original assembly model by calling the API functions and for tessellating them to obtain mesh models. In this program, when exporting the feature mesh model files, each form feature is also given a unique identification number having 24-bit integer value which enables the system to differentiate 16 million form features.

3.2 Extracting characteristics from rendering results

A set of view points and directions, and rendering color are similarly determined by the setting described in Sect. 2.2.1.

The rendering of feature mesh models is carried out two times in each view; the rendering of each form feature alone and the rendering of whole assembly model. And then the only two characteristics are evaluated from the RGB values of the pixels left in the frame buffer.

(1) $\hat{P}_i^j$: the number of pixels with ith form feature's color when ith feature is rendered alone at jth view,
(2) $\hat{Q}_i^j$: the number of pixels with ith form feature's color when whole assembly model is rendered at jth view.

3.3 Recognition and removal of invisible form features

3.3.1 Classify invisible parts and form features

Using characteristics $\hat{P}_i^j$, $\hat{Q}_i^j$, the following evaluation index defined by equation (5) is calculated for ith form feature to determine its visibility in the assembly model.

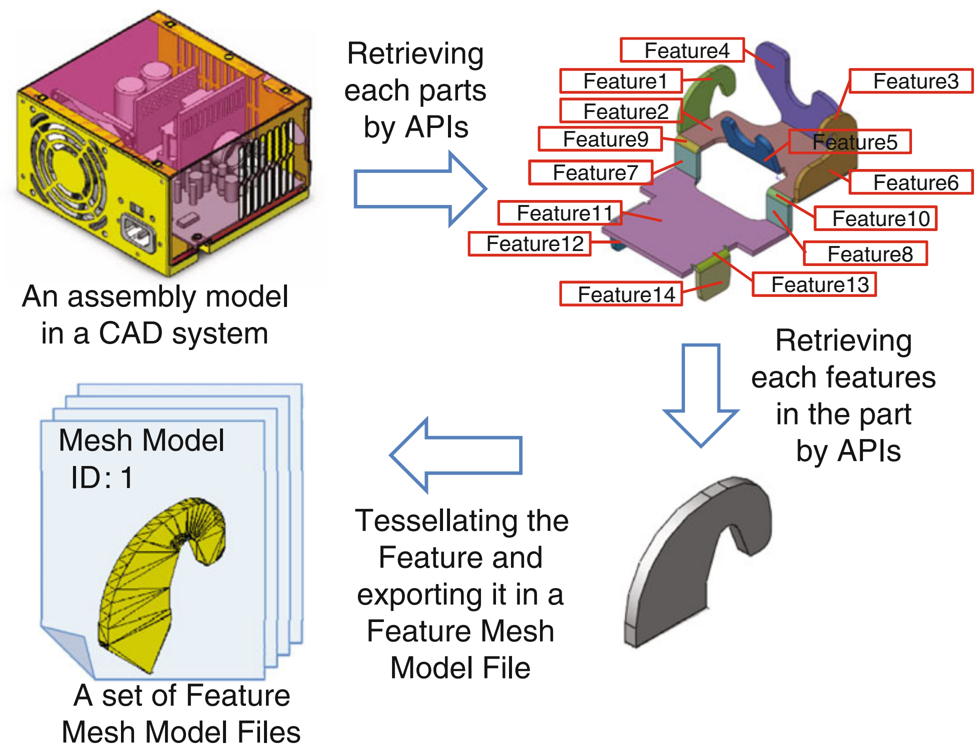


Fig. 6 Generation process of feature mesh model files

(d) The maximum exposure rate of *i*th form feature among all view,

$$\hat{F}_i^{visible} = \max(\hat{Q}_i^j / \hat{P}_i^j) \tag{5}$$

Consequently, using user-specified threshold τ_V, the form feature which satisfies the following condition (6) is determined as completely or nearly "invisible".

$$\hat{F}_i^{visible} < \tau_V \tag{6}$$

3.3.2 Invisible form feature removal using feature dependency graph

While invisible parts can be removed directly in Sect. 2, in case of feature simplification, the invisible form features does not necessarily become features to be removed. Invisible form features which are not needed for defining of visible form feature geometries can only be removed. In our study, such dependency of geometric modeling among the features can be expressed as a digraph called *feature dependency graph*. This graph is created per one part.

As shown in an example of Fig. 7, a directed edge of the graph shows that parent-features at the starting point of the edge need to be defined before the child-features at the end point of the edge are defined. Also the property of visibility (visible/invisible) of the feature is attached to each feature node of the graph. Figure 7 shows an example of the feature dependency graph with the visibility properties.

Invisible features to be removed are determined based on the following procedures and the feature dependency graph;

(1) Obtain a set of nodes without child-node from all nodes in the feature dependency graph, and create the node subset.
(2) If there are nodes having invisible property in the subset, remove these nodes and their incident edges from the graph, and go back to (1). Otherwise, proceed to (3).
(3) Repeat above (1) and (2) for every feature dependency graph corresponding to each part in the assembly.

In Fig. 6, two invisible form features f_4 and f_5 are only removed from the assembly model, because they are invisible and are not needed for defining of visible form features f_3.

By eliminating form features to be removed from the assembly model, we can obtain the final simplified 3D assembly model. This removal process can be automatically done by our developed API application of a commercial 3D CAD software.

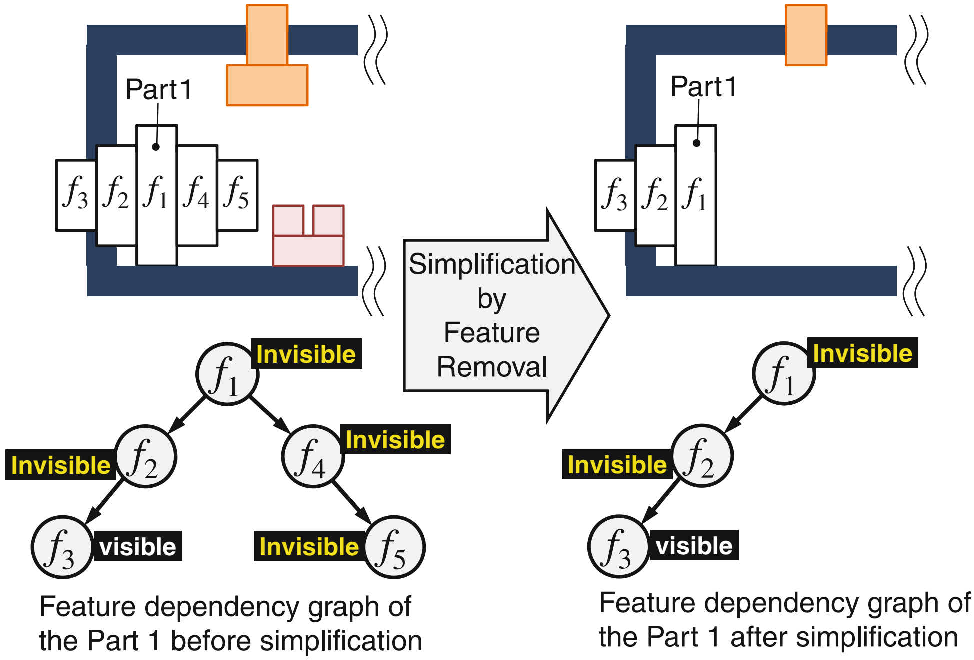


Fig. 7 Feature removal using feature dependency graph

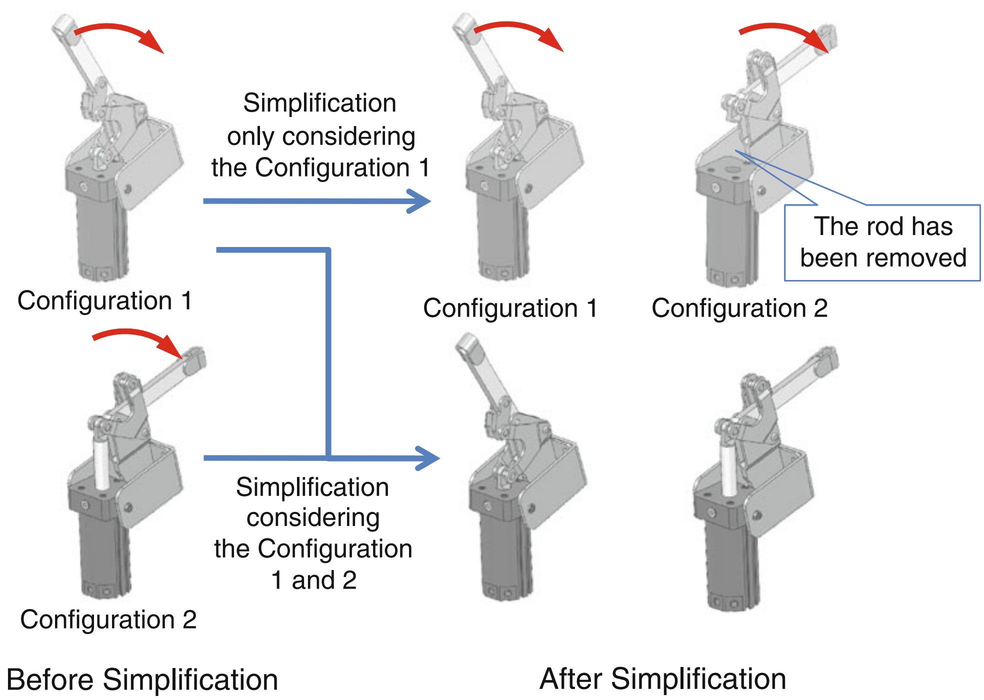


Fig. 8 Simplification considering visibility change of parts and form features

4 Feature simplification method considering visibility change caused by movable parts

The visibility of the form features and the parts could change as the position and the orientation of movable parts (assembly configurations) change in an assembly. So, as shown in Fig. 8, the simplification method should not remove too much features and parts which could be exposed to outside at particular parts' positions and orientations. Therefore, in this section, the feature simplification method considering the visibility change caused by the movable parts is described. The fundamental algorithm is as same as the one of the feature simplification, but an additional function for changing assembly configuration is needed. The processes which are

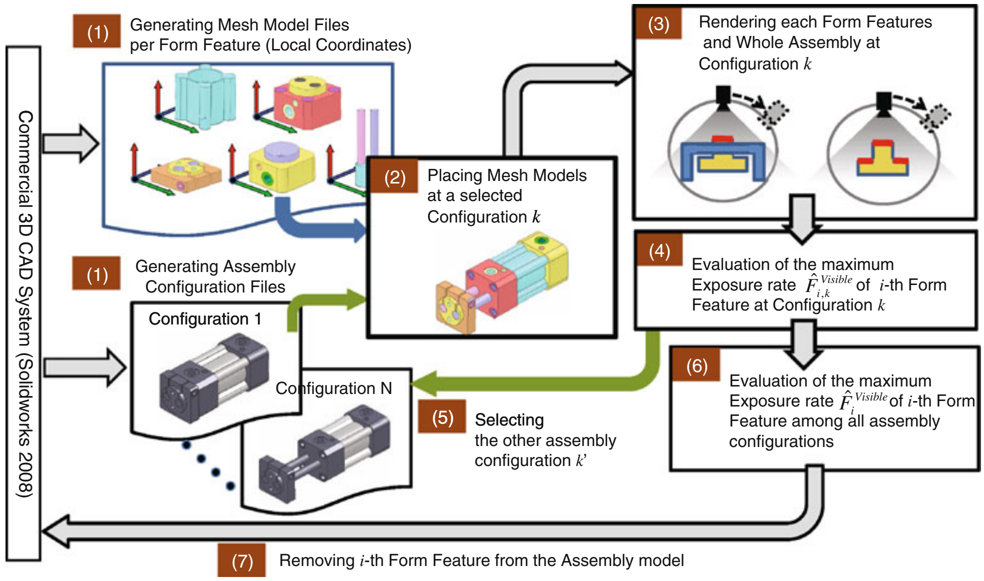


Fig. 9 Processes of feature simplification considering visibility change

fundamentally different from the original feature simplification are only explained in this section.

Figure 9 shows processes of the feature simplification method modified for the visibility change caused by movable parts. The process is done by the following procedures from (1) to (7);

(1) Triangular mesh models per form feature are generated and are outputted to separate mesh model files from commercial 3D CAD software. At the same time, the possible set of their relative positions and orientations (assembly configurations) are exported to a set of assembly configuration files from the CAD software in the form of transformation matrices. In our current implementation, the possible assembly configurations are manually specified by the user in the CAD software.
(2) One assembly configuration k is selected from the ones specified in an assembly configuration file, and each of the mesh models of the features are placed at its right position and orientation to express an assembly at this selected configuration.
(3) Using the mesh models build in (2), both each form feature alone and whole assembly model are rendered in the same way as Sect. 3.2.
(4) The maximum exposure rate of ith form feature among all view at this $k-$th assembly configuration $\hat{F}_{i,k}^{visible}$ is evaluated in the same way as Sect. 3.3.1.
(5) The other assembly configuration k' is selected, and the steps (2), (3) and (4) are repeated until the maximum exposure rates of the form features at all assembly configurations are evaluated.
(6) The maximum exposure rate of ith form feature $\hat{F}_{i}^{visible}$ among all assembly configurations is derived as

$$\hat{F}_{i}^{visible} = \max_{k}\left(\hat{F}_{i,k}^{visible}\right) \tag{7}$$

Similar to the Eq. (6), using user-specified threshold τ_V, the form feature which satisfies the Eq. (8) is determined as completely or nearly "invisible" when some parts in the assembly moves.

$$\hat{F}_{i}^{visible} < \tau_V \tag{8}$$

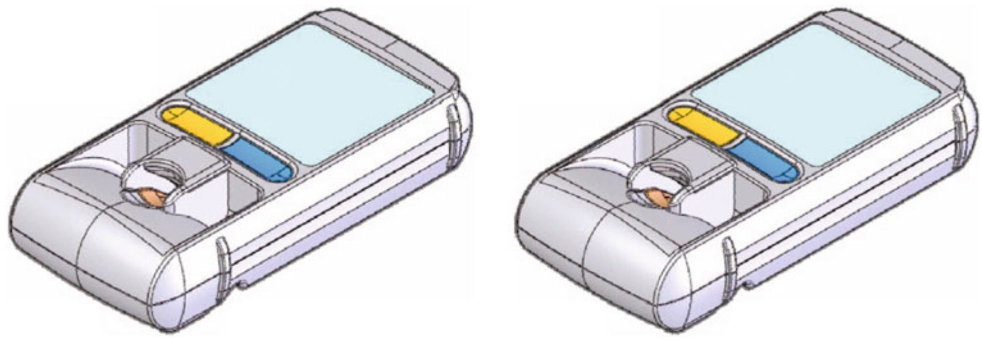
(7) Using the feature dependency graph, the features which can be removed are automatically deleted from the 3D CAD model in the same manner as the one described in Sect. 3.3.2.

5 Results

5.1 The result of the part simplification

The simplification performance was evaluated for an original assembly CAD model made by SolidWorks2008. As shown in Fig. 10, this assembly model was built for a real commercial product of "Sphygmograph" which has 59 parts including the upper and the lower housings and many inner parts.

Figure 10a, b show the appearances of sphygmograph models before and after the part simplification. It was found that the appearances of the simplified model were unchanged as the original one. Additionally inner part structures of the models are shown in Fig. 11. By comparing Fig. 11a.1 with Fig. 11b.1, we could confirm that the invisible inner parts were entirely eliminated. However, many detailed invisible



(a) Before Simplification **(b)** After Simplification

Fig. 10 Appearance of assembly models of sphygmograph

features such as bosses or ribs still remain on the back side of the upper and the lower housings when using the part simplification since the housings of the simplified model (Fig. 11b.2, b.3) remain unchanged as the original ones (Fig. 11a.2, a.3).

5.2 The result of the feature simplification

The result of the feature simplification is shown in Fig. 12. The appearance of the simplified model shown in Fig. 12a was unchanged as the original one shown in Fig. 10a. By comparing Fig. 11a.1 with Fig. 12b, we could confirm that inner invisible parts and invisible form features were entirely eliminated. Moreover, by comparing Fig. 11a.2, a.3 with Fig. 12c, d, many detailed invisible features were almost eliminated from back side surfaces of the upper and the lower housings. Note that the feature simplification includes simplification effect of the parts simplification, even if the feature simplification is only carried out.

Figure 13 shows the other comparisons of the appearances and the inner structures of assembly models before and after the feature simplification. It was shown that the external appearances did not change but many parts and form features inside the housings were correctly removed after the simplifications.

5.3 The performance comparison of both results

The simplification results on the other three assembly models are shown in Table 1. Each threshold for simplification was adjusted appropriately so that the model appearances were unchanged. Each lower row of Table 1 describes the rate of reduction r which is calculated from Eq. (9).

$$r = \{1 - (M_{simple}/M_{original})\} \times 100(\%) \tag{9}$$

where, $M_{original}$ is the number of entities, such as parts, form features and faces, in the original assembly model, and M_{simple} is those in the simplified model.

In case of the part simplification, the rate of reduction increases as the number of parts in the assembly increases. The average rate of reduction of parts is almost 60%, and that of faces resulting from the parts reduction is almost 40–50%.

On the other hand, the feature simplification has enough availability for data reduction of the model even which has a small number of parts. In case of "power supply" assembly, the part model of the printed circuit board inside the housing has a large number of detailed features. While in case of "Sphygmograph" assembly, the large number of invisible detailed features such as bosses exists on back side of the injection molded housings. Most of these detailed features could be efficiently eliminated by the feature simplification, and high rate of reduction can be expected when being applied to such kind of assembly model.

The average rate of reduction of features is almost over 60%, and that of faces resulting from the features reduction is almost 65–70%. The feature simplification is superior to the part simplification in terms of reduction efficiency.

5.4 The processing time

The processing time of both methods at each processing step in Fig. 2 are summarized in Table 2.

The time for rendering parts or form features from 26 view points(A-3) and recognizing invisible ones (A-4 a,b) which are the essential steps in our algorithm took less than a few minutes even in case of a large scale assembly model with 451 parts ("stove burner"), and can be considered very fast.

On the other hand, the time of generating mesh model (A-1 a,b) by the feature simplification took six or seven times than that of the part simplification. And the removing invisible features (A-5) also took a large amount of time. This is because generating feature mesh model and removing features from assembly needs many nested API function calls of the CAD system. Improving the implementation of these steps may make the processing time more efficient and is left as a future work.

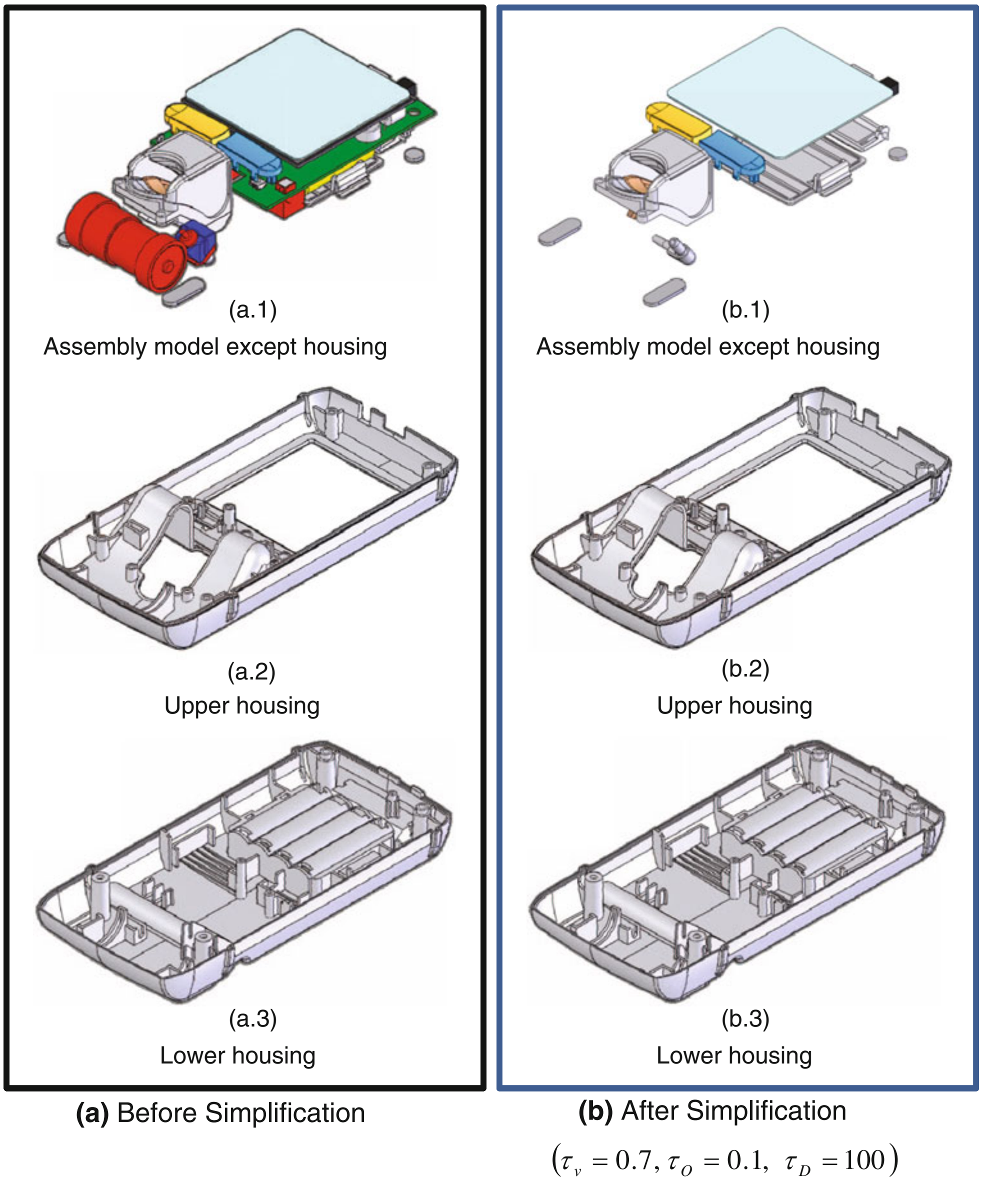


Fig. 11 Inner part structures of assembly models of sphygmograph

5.5 The result and performance of the feature simplification considering the visibility change caused by the movable parts

Figure 14a shows the appearances of pneumatic cylinder models before and after the feature simplification method where the visibility change by the movable parts was not considered. As shown in the figure, when the "piston" part moves forth from the cylinder part, the portion of the simplified model which have been hidden by the piston are exposed to outside. And it is clear that two "rod" parts and four thread-hole features on a fixed part which should be preserved have been removed after the simplification and that the appearance of the assembly model is remarkably disfigured.

On the contrary, Fig. 14b shows the appearances of them before and after the feature simplification method where the visibility change is considered. In this example, two assembly configurations were selected in the simplification where the piston part was positioned at back and forth ends. As shown in the figure, the external appearance of the assembly model after the simplification was correctly preserved so as to be as same as the original one.

Figure 15 shows another example of the feature simplification for sphygmograph models. Two assembly configurations were specified in the simplification; a normal usage configuration (Fig. 15a) and a configuration of service situation where the positions of two buttons on the top were moved upward and the battery case lid moved downward (Fig. 15b).

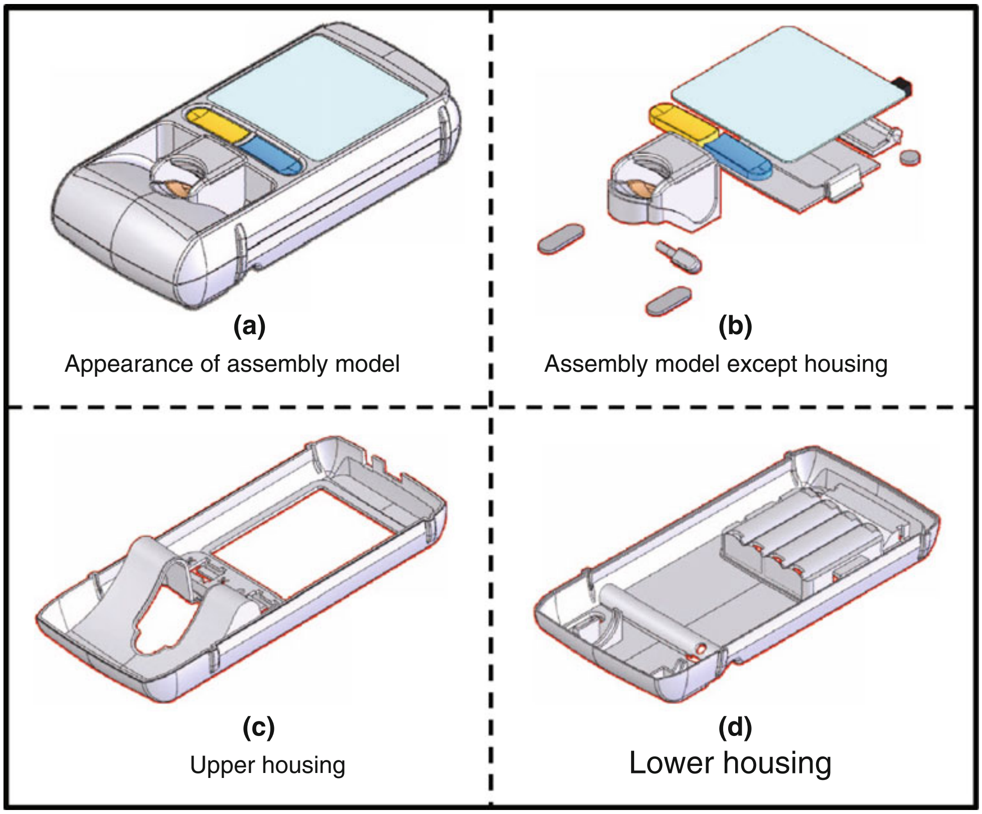


Fig. 12 A result of the feature simplification method of the sphygmograph assembly model ($\tau_v = 0.1$)

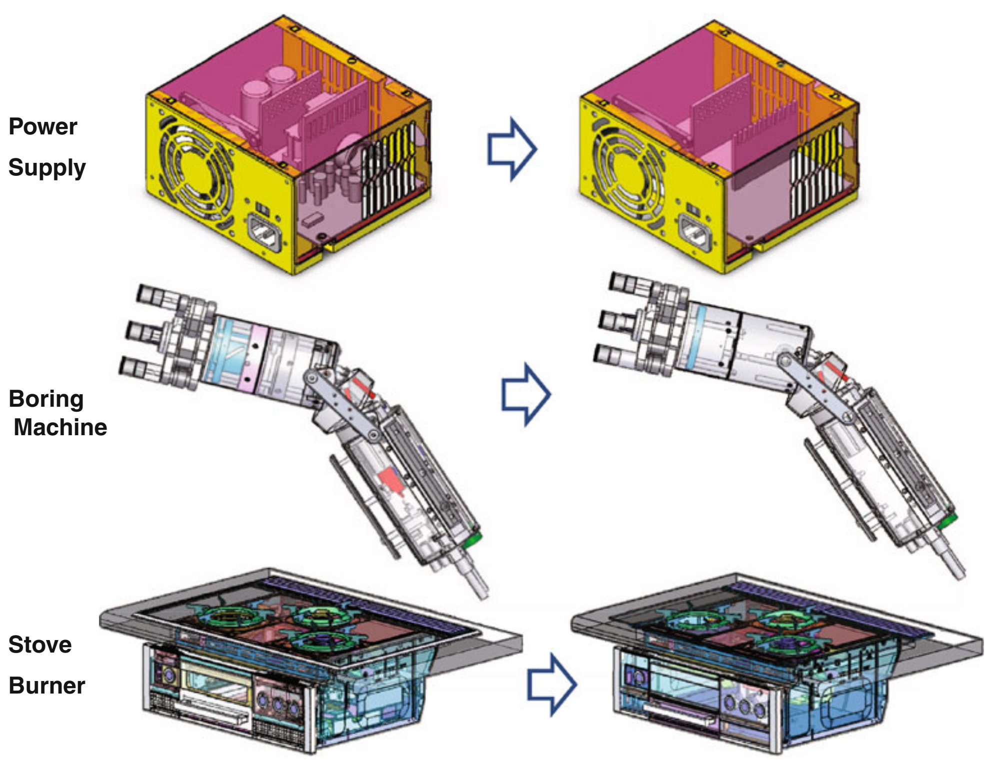


Fig. 13 The appearances and inner structures of assembly models before and after the feature simplification (external housings are displayed as semitransparent materials)

Table 1 The results of simplified assembly models

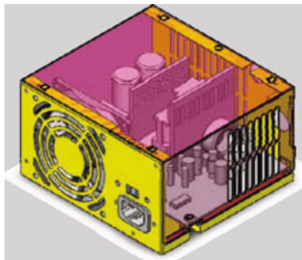
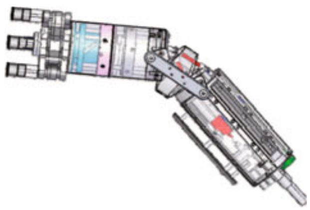
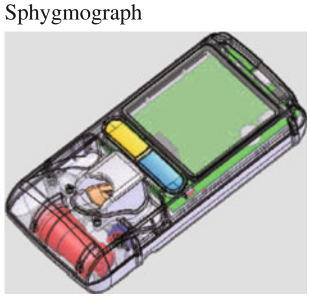
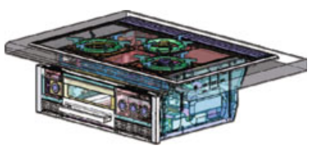
Model	Simplification method	# of parts	# of features	# of faces
Power supply 	Original Model	6	161	1,800
	Part simplification	6	161	1,800
	Rate of reduction (%)	0	0	0
	Feature simplification	4	58	530
	Rate of reduction (%)	33	64	70
Boring machine 	Original model	415	421	7,295
	Part simplification	192	192	3,964
	Rate of reduction (%)	53	54	48
	Feature simplification	192	178	3,726
	Rate of reduction (%)	53	58	49
Sphygmograph 	Original model	59	538	9,153
	Part simplification	27	398	6,601
	Rate of reduction (%)	59	26	28
	Feature simplification	18	232	2,860
	Rate of reduction (%)	68	56	69
Stove burner 	Original model	451	6,537	57,211
	Part simplification	135	2,782	24,993
	Rate of reduction (%)	68	57	56
	Feature simplification	108	1,508	13,074
	Rate of reduction (%)	76	77	77

Table 2 The processing time of simplification methods (s)

Model	Simplification method	Generating mesh models	Rendering and recognizing invisible parts/features	Removing features
Power supply	Part simplification	8	6	
	Feature simplification	50	6	24
Boring machine	Part simplification	62	50	
	Feature simplification	400	16	230
Sphygmograph	Part simplification	41	5	
	Feature simplification	300	13	240
Stove burner	Part simplification	350	88	
	Feature simplification	2,375	58	3,900

As shown in Fig. 15c, it was confirmed that the form features inside the battery housing which become visible only at the service situation was correctly preserved and that the appearance of the assembly model after the simplification was correctly preserved.

Finally, the simplification performances in above cases are summarized in Table 3. Because the pneumatic cylinder models consisted of only five parts all of which were already exposed to outside, the rate of reduction was very small. On the other hand, because the sphygmograph models

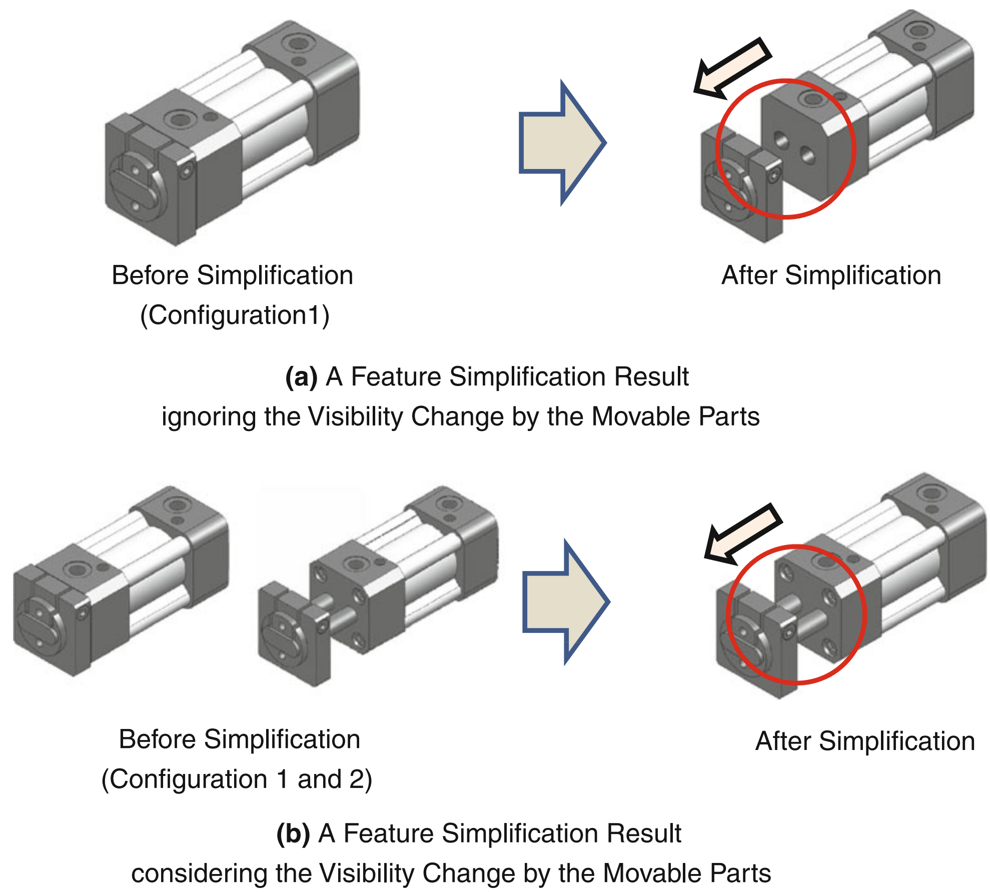


Fig. 14 Comparison of the feature simplification results considering the visibility change

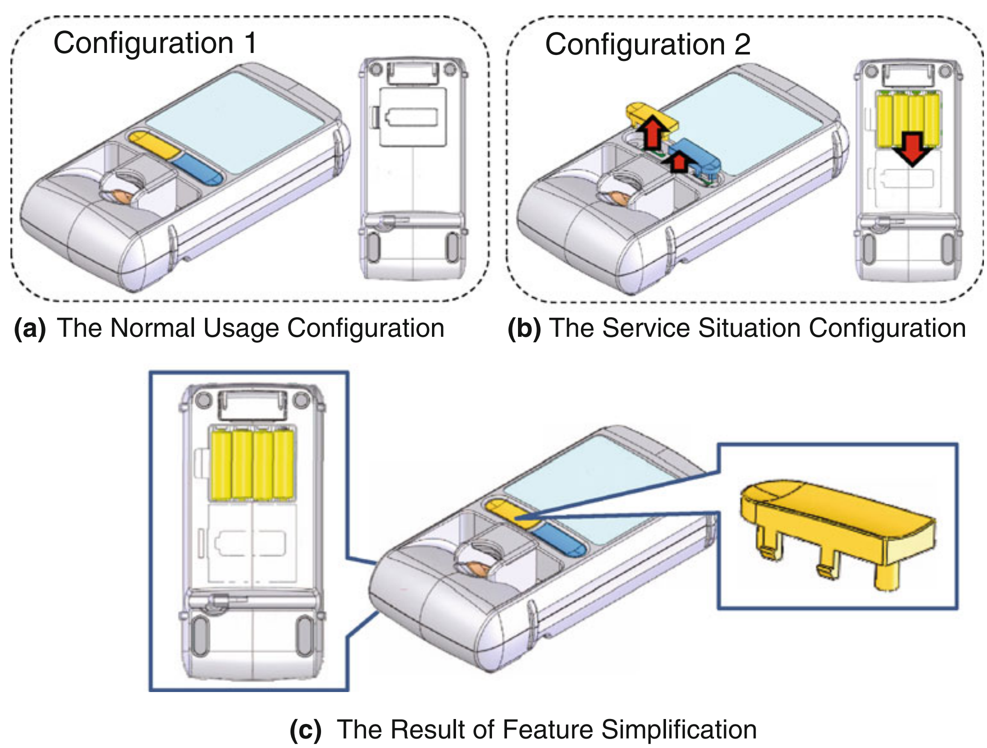
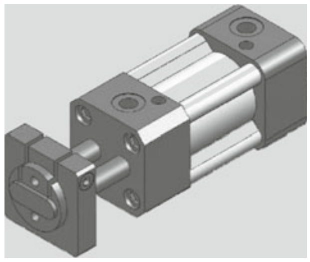
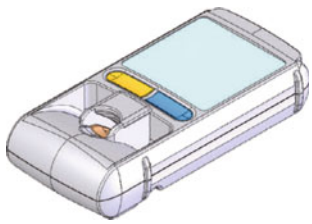


Fig. 15 The feature simplification results considering two assembly configurations

Table 3 The results of simplified assembly models considering the visibility change

Model	Simplification method	# of parts	# of features	# of faces
Pneumatic cylinder	Original model	5	19	240
	Feature simplification ignoring visibility change	4	6	189
	Rate of reduction (%)	20	68	21
	Feature simplification considering	5	18	237
	Rate of reduction (%) visibility change	0	5	1
	Rate of reduction (%)	0	5	1
Sphygmograph	Original model	59	538	9,153
	Feature simplification ignoring visibility change	18	232	2,860
	Rate of reduction (%)	68	56	69
	Feature simplification considering visibility change	34	276	4,245
	Rate of reduction (%)	42	49	54

included many invisible parts and features inside the housing, the rate of reduction of parts, features and faces still remained large even when multiple assembly configurations of movable parts were considered in the simplification. This showed that the proposed simplification method can still work effectively when an assembly model includes a large number of parts inside the housing which are not exposed to outside in any assembly configuration.

6 Conclusions

Two appearance preserving simplification methods for 3D assembly CAD model was proposed by recognizing and removing invisible parts or form features from the model using the graphical rendering results left on the frame buffer.

The proposed part simplification method could reduce the number of parts by almost 60% and faces by almost 40–50%, which is good for the rapid and rough simplification. On the other hand, the feature simplification method could reduce the number of form features by over 60% and faces by almost 65–70%, which is good for archiving the efficient reduction rate. Recognizing invisible parts or form features to be removed could be finished within a practical time frame.

The feature simplification method considering the visibility change at many assembly configurations was also implemented and tested where the movable parts change their positions and orientations in the assembly model. In the case that the assembly model includes a large number of invisible parts inside the housing, it was confirmed that the rate of reduction of parts, features and faces almost remained as large as the one where the visibility change was not considered.

In this performance experiments, since the assembly models have about 500 parts at most, occlusion culling using modern GPUs may enable faster rendering of CAD models than the proposed method does. However current general large scale assembly CAD models such as medical equipments sometimes have 5,000 parts or more. In these cases, we need not only to increase the rendering speed of the assembly models in CAD systems, but also to shorten the time for reading the models and constructing the model at the system start-up. Using modern GPUs cannot fulfill the latter requirement, but our proposed simplification methods are very effective to solve it.

As our simplification methods only aim for appearance preserving simplification of the large scale CAD models, the simplified model by our methods is not useful to generate a cutaway model or 2D drawings. Appearance preserving simplification suitable for these purposes will be one of our future plans.

References

1. Hoppe, H., DeRose, T., Duchamp, T., McDonald, J., Stuetzle, W.: Mesh optimization. In: Proceedings of SIGGRAPH 93, Anaheim, pp. 19–26 (1993)
2. Garland, M., Heckbert, P.: Surface simplification using quadric error metrics. In: Proceedings of SIGGRAPH 97, Los Angeles, pp. 209–216 (1997)
3. Schroeder, J., Zarge, J., Lorensen, W.: Decimation of triangle meshes. In: Proceedings of SIGGRAPH 92, Chicago, pp. 65–70 (1992)
4. El-Sana, J., Varshney, A.: Topology simplification for polygonal virtual environments. IEEE Trans. Vis. Comp. Graph. **4**(2), 133–144 (1998)

5. Xia, C., El-Sana, J., Varshney, A.: Adaptive real-time level-of-detail-based rendering for polygonal models. IEEE Trans. Vis. Comp. Graph. **3**(2), 171–183 (1997)
6. Luebke D., Reddy M., Cohen J.D., Varshney A., Watson B., Huebner R.: Level of detail for 3D graphics, Morgan Kaufmann, San Francisco (2003)
7. Seo, J., Song, Y., Kim, S., Lee, K., Choi, Y., Chae, S.: Wrap-around operation for multi-resolution CAD model. Comput.-Aided Des. Appl. **2**(1–4), 67–76 (2005)
8. Lee, S.: Feature-based multiresolution modeling of solids. ACM Trans. Graph. **24**(4), 1417–1441 (2005)
9. Wakita, A., Yajima, M., Harada, T., Toriya, H., Chiyokura, H.: XVL: a compact and qualified 3D representation with lattice mesh and surface for the Internet. In: Proceedings of the 5th symposium on virtual reality modeling language 2000/Web 3D, Monterey, pp. 45–51 (2000)