Appearance preserving simplification of large scale assembly model by invisible part and form feature removal

Daisuke Iyoda [1], Satoshi Kanai [1], Yui Endo [2], Hideki Sakamoto [3] and Naoki Kanatani [3]

(1) : Graduate School of Information Science and Technology, Hokkaido University
Kita-14, Nishi-9, Sapporo 060-0814, Japan
Phone +81-11-706-6449 / Fax +81-11-706-7210
E-mail : d_iyoda@sdm.ssi.ist.hokudai.ac.jp
kanai@sdm.ssi.ist.hokudai.ac.jp

(2) : Digital Human Research Centre,
Advanced Industrial and Technology(AIST)
2-3-26 Aoumi, Kotoku-ku, Tokyo,
135-0064, JAPAN
Phone +81-3-3599-8504
E-mail : y.endo@aist.go.jp

(3) : Kozo Keikaku Engineering Inc.
4-38-13 Honcho, Nakano-ku, Tokyo, 164-0012, JAPAN
Phone: +81-3-5342-1122
E-mail : {sakamoto, kanatani}@kke.co.jp

Abstract: Recently, 3D CAD models have been widely used not only in product development but in design review. And large scale assembly models, which have full-detailed inner structures, force 3D CAD systems to take a fair amount of time to read and render them. However, when they are used for browsing, styling review and sales purposes, there is hardly the occasion where full-detailed assembly models are required, the primary purpose of the systems 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.
We propose two appearance preserving simplification methods of large scale assembly model by removing invisible parts or invisible form features from the model in commercial 3D CAD systems. Our methods are based on an algorithm which can detect invisible parts or features by pre-rendering the models from multiple view directions and by reading the rendered results from frame buffer. Our algorithm can be carried out regardless of CAD systems. Invisible parts or features detection is robust and fast in the algorithm. The performances of both methods are discussed.

Key words: 3D CAD, assembly model, solid model, form feature , model simplification.

1- Introduction

Currently 3D CAD systems which had only been used in upper stage of manufacturing scenes become widely used in lower stages such as styling review, sales purposes, model exporting to external users or e-commerce on Web along with improved performance of computers. In order to accurately carry out mass property calculation or clearance check and to create BOMs without omissions, solid models even of very fine parts or full-detailed shapes have been built in the 3D CAD systems, and the assembly models have had 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 have had very complex geometries such as bosses, ribs, cuts and etc.. 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, the primary purpose of the systems 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.

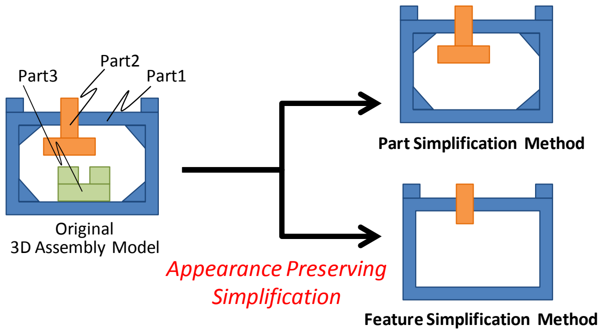


Figure 1: The proposed simplification methods of assembly model

The requirements for the appearance preserving simplification can be considered as the followings;

—The format of simplified models should be as same as the original one for the sake of users' convenience.

—As shown in figure 1, as inside of part has many complex form features, the simplification should perform not only part-by-part elimination of invisible parts but feature-by-feature elimination of invisible features.

—The simplification method should have robustness, high-speed performance and portability for various commercial CAD systems.

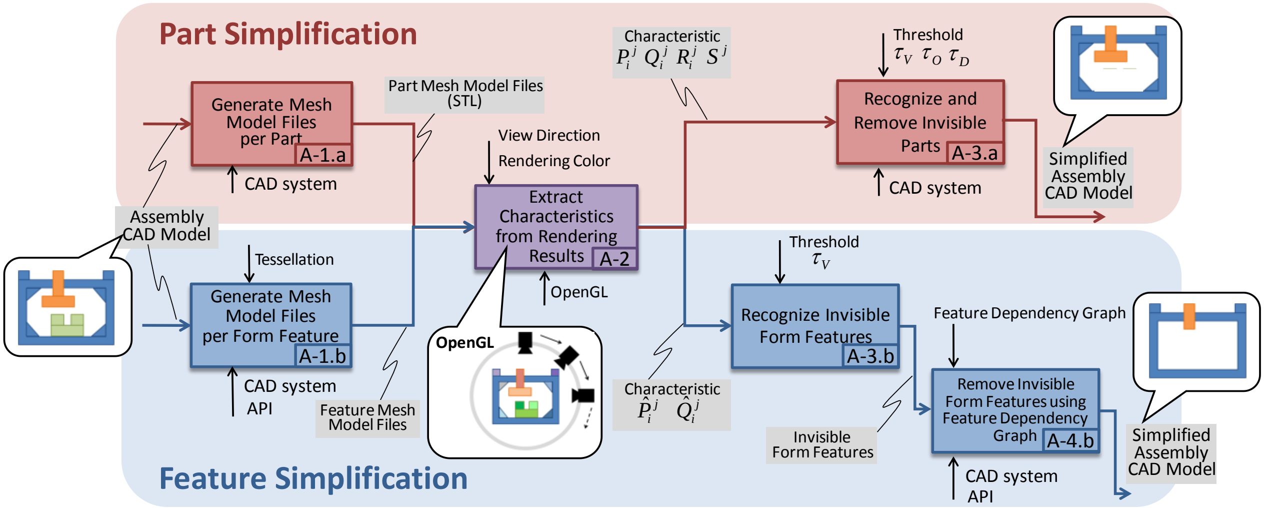


Figure 2: Overview of the two appearance preserving simplification methods

1.1 – Related works

So far many techniques have been proposed for simplification of 3D geometric model. 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*[HD1], *vertex-pair-collapse*[GH1] or *vertex-removal*[SZ1] operations. Topology-changing simplification of triangular mesh model was also proposed using *alpha-hull* over polygonal objects [EV1]. However, these simplification techniques cannot be applied to the feature-based solid model of commercial CAD systems.

While as for the latter techniques, Seo et.al [SS1] 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 [L1] 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 didn't aim for appearance preserving simplification of assembly models. Moreover, the method of Lee [L1], 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. 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 rounded surfaces. However, for this reason, the simplified CAD model format is 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 detect invisible parts and invisible form features in assembly model automatically.

1.2 – Purpose and algorithm overview

In this paper, as shown in Figure 2, we propose two appearance preserving simplification methods by removing 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. Our methods are based on an algorithm which can detect invisible parts or invisible features by pre-rendering the models from multiple view directions and by reading the rendered results from frame buffer. The core simplification algorithm is common in two methods.

In 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 (A-1.a), (A-1.b). Secondly, 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-2). 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-3.a), (A-3.b). In case of part simplification, the invisible parts are manually removed from the original assembly model in commercial CAD system (A-3.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-4.b). In this study, our methods are implemented as add-on C++ software for SolidWorks2008.

In the following sections of this paper, section 2 describes the part simplification method, and section 3 does the feature simplification method. Section 4 presents the results of above two methods, and the performances of them are compared. Section 5 presents conclusions.

2- Part simplification method

2.1 –Generating mesh model per part

In order to detect invisible parts, our algorithm uses the rendering result of 3D assembly model. As it is difficult to access to the frame buffer of CAD system, pre-rendering of the model is carried out on another viewport by OpenGL. For this reason, mesh models per part in the assembly are generated by faceting surfaces of CAD model and are outputted to STL format files (part mesh model files). This operation can be carried out using standard exporting function of general commercial CAD systems.
In addition, each part in the assembly model is given an identification number having 24-bit integer value which enables the system to differentiate about 16 million parts.

2.2 –Rendering parts and 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.

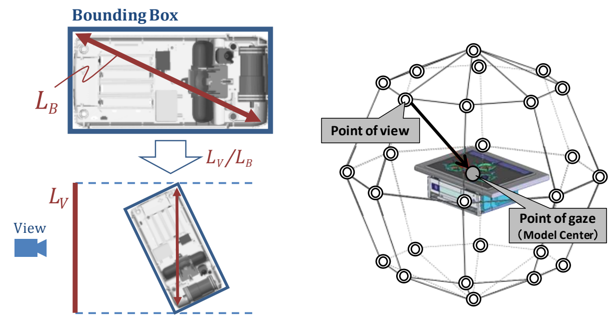
2.2.1 –View point setting and color allocation

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 predetermined size of frame buffer (640×640 pixels). Therefore, as shown in Figure 3(a), 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 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 Figure 3(b), the view points are placed at the vertices of polyhedron approximated a sphere, and the point of gaze is located at the sphere center whose location is identical to the centroid of the assembly model. In this study, as shown in Figure 3(b), 26 view points are used for the rendering. 24-bit RGB value converted from a 24-bit identification number 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 invisible in a simple way.

2.2.2 – Extracting characteristics of rendering results

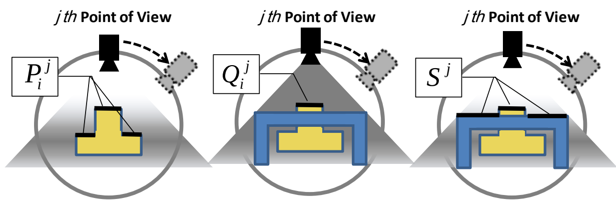
The rendering of mesh models carried out the following two times in each view; the rendering of each part model alone and the rendering of whole assembly model. Then, the following four characteristics are evaluated from the RGB values of the pixels left in the frame buffer by calling *glReadPixels*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

1) P_i^j : the number of pixels with i-th part's color when i-th part is rendered alone at j-th view, (See Figure 4(a))
2) Q_i^j : the number of pixels with i-th part's color when whole assembly model is rendered at j-th view, (See Figure 4(b))
3) S^j : the number of pixels with any part's color when whole assembly model is rendered at j-th view. (See Figure 4(c))
4) R_i^j : the largest number among the number of pixels with i-th part's color at each depth level, when whole assembly model is rendered at j-th view. (See Figure 5)



(a)The rendering scale factor **(b) The view directions**

Figure 3: The pre-process of rendering.



(a) P_i^j **(b) Q_i^j** **(c) S^j**

Figure 4: The characteristics.

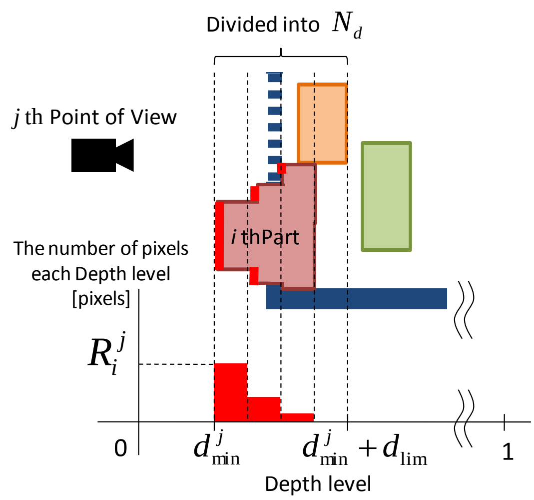


Figure 5: The number of pixels each depth level.

As shown in Figure 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 j-th 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, N_d=20 are used.

2.3 – Recognition and removal of invisible parts

Using four the characteristics discussed above, the following three evaluation indices are calculated for i-th part to determine the visibility of the each part in the assembly model.

a) The maximum exposure rate of i-th part among all view,

$$F_i^{visible} \equiv \max(Q_i^j / P_i^j) \tag{1}$$

b) The maximum occupation rate of i-th part among all view,

$$F_i^{Occupied} \equiv \max(Q_i^j / S^j) \tag{2}$$

c) The maximum contribution of i-th part to exterior shape of assembly,

$$F_i^{Depth} \equiv \max(R_i^j) \tag{3}$$

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

$$(F_i^{Visible} = 0) \vee \{(F_i^{Visible} < \tau_V) \wedge (F_i^{Occupied} < \tau_O) \wedge (F_i^{Depth} < \tau_D)\} \tag{4}$$

The condition $F_i^{Visible}$=0 means that i-th 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 eliminating 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 appropriately, the simplified 3D assembly model can be generated where different numbers of nearly invisible parts are eliminated.

3- Feature simplification method

In this section, the simplification method by removing invisible form features is described whose fundamental algorithm is as same as the one of the part simplification.

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 by using API. In addition, each form feature in the assembly model is given an identification number having 24-bit integer value.

3.2 – Extracting characteristics of rendering

A set of view points and directions, and rendering color are similarly determined by the setting described in 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 i-th form feature’s color when i-th feature is rendered alone at j-th view,

2) $\hat{Q}_i^j$: the number of pixels with i-th form feature’s color when whole assembly model is rendered at j-th 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 is calculated for i-th form feature to determine its visibility in the assembly model.

a) The maximum exposure rate of i-th form feature among all view,

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

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 section 2, in case of feature simplification, the invisible form features

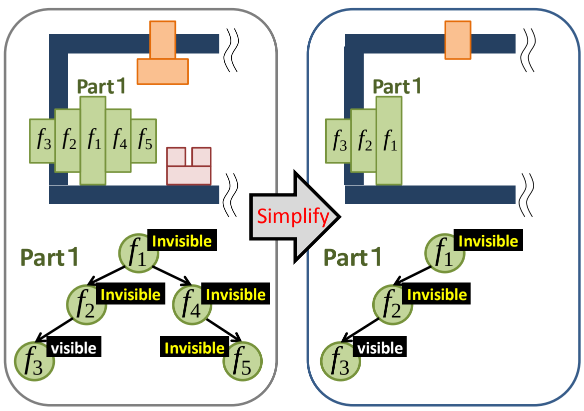


Figure 6: Removal using feature dependency graph.

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.
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 6 shows an example of the feature dependency graph.
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 Figure 6, two invisible form features(f_4, f_5) are only removed from the assembly model.
By eliminating form features to be removed from the assembly model, we can obtain the final simplified 3D assembly model.

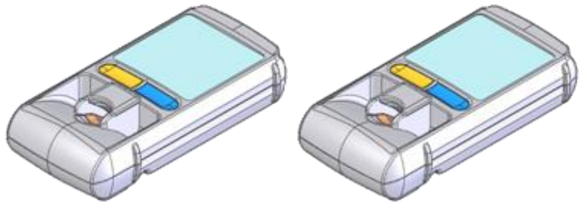
4- Results

4.1 – The result of the part simplification

The simplification performance was evaluated for an original assembly CAD model made by SolidWorks2008. 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.
Figure7(a),(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 Figure 8. By comparing Figure 8 (a.1) with Figure 8(b.1), we could confirm that the invisible inner parts were entirely eliminated. However, many detailed invisible 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 (Figure 8(b.2) and (b.3)) remain unchanged as the original ones (Figure 8(a.2) and (a.3)).

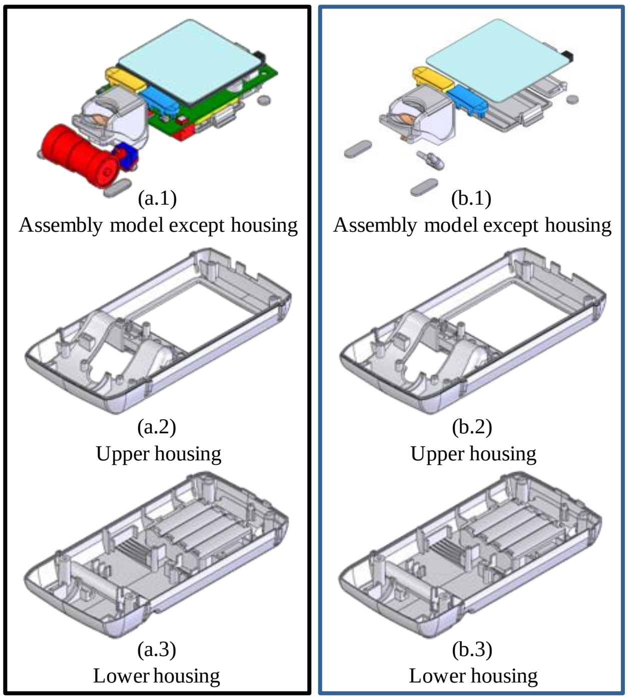
4.2 – The result of the feature simplification

The result of the feature simplification is shown in Figure 9. The appearance of the simplified model shown in Figure 9 (a) was unchanged as the original one shown in Figure 7(a). By comparing Figure 8(a.1) with Figure 9(b), we could confirm that inner invisible parts and invisible form features were entirely eliminated. Moreover, by comparing Figure 8(a.2) and (a.3) with Figure 9(c) and (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.



(a) Before simplification (b) After part simplification

Figure 7: Appearances of assembly CAD models. (Sphygmograph)



(a) Before simplification (b) After part simplification (τ_V =0.7, τ_O =0.1, τ_D =100)

Figure 8: Inner part structures of the assembly CAD models. (Sphygmograph)

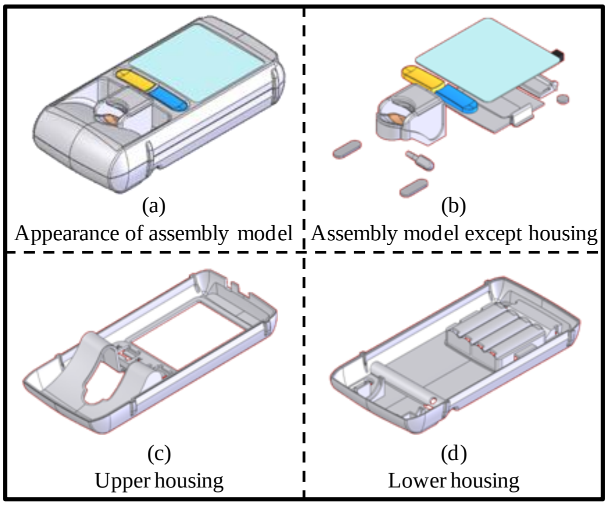


Figure 9: A result of the feature simplification method.
(τ_V =0.1)

4.3 – 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 equation (7).

$$r = \{1 - (M_{simple} / M_{original})\} \times 100 \quad [\%] \qquad (7)$$

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.

Table 1: The results of simplified assembly models.
(lower row : rate of reduction)

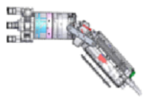
Model	Method	Parts	Features	Faces
Power Supply 	Original model	6	161	1800
	Part Simplification	6	161	1800
		0%	0%	0%
	Feature Simplification	4	58	530
		33%	64%	70%
Boring Machine 	Original model	415	421	7295
	Part Simplification	192	192	3964
		53%	54%	46%
	Feature Simplification	192	178	3726
		53%	58%	49%
Sphygmograph 	Original model	59	538	9153
	Part Simplification	27	398	6601
		59%	26%	28%
	Feature Simplification	18	232	2860
		68%	56%	69%
Stove Burner 	Original model	451	6537	57211
	Part Simplification	135	2782	24993
		68%	57%	56%
	Feature Simplification	108	1508	13074
		76%	77%	77%

Table 2: The processing time of simplification methods.

[s]

Model	Method	A-1	A-2 A-3	A-4
Power Supply	Part	8	6	
	Feature	50	6	24
Sphygmograph	Part	41	5	
	Feature	300	13	240
Boring Machine	Part	62	50	
	Feature	400	16	230
Stove Burner	Part	350	88	
	Feature	2375	58	3900

4.4 – The processing time

The processing time of both methods at each processing step in Figure 2 are summarized in Table 2.
The time for rendering parts or form features from 26 view points(A-2) and recognizing invisible ones (A-3) 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) by the feature simplification took six or seven times than that of the part simplification. And the removing invisible features (A-4) 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.

5- 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.
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 5000 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 planes.

References

[EV1] El-Sana, J. ,Varshney, A. Topology simplification for polygonal virtual environments, IEEE Transactions on Visualization and Computer Graphics, 4(2):133-144, 1998.

[GH1] Garland, M and Heckbert, P. Surface Simplification Using Quadric Error Matrics, Proceedings of SIGGRAPH 97, 209-216, 1997.

[HD1] Hoppe, H. , DeRose, T. , Duchamp, T. ,McDonald, J. , Stuetzle, W. Mesh Optimization. Proceedings of SIGGRAPH 93, 19-26, 1993.

[L1] Lee,S. Feature-based multiresolution modeling of solids. ACM Trans. Graph, 24(4): 1417-1441, 2005.

[SS1] Seo, J. , Song , Y. , Kim, S. , Lee, K. , Choi, Y. and Chae, S. Wrap-around Operation for Multi-resolution CAD Model, Computer-Aided Design & Applications, 2(1-4):67-76, 2005.

[SZ1] Schroeder, J. , Zage, J and Lorensen, W. Decimation of Triangle Meshes. Proceedings of SIGGRAPH 92, 65-70, 1992.

[XE1] Xia, C. , El-Sana, J. , Varshney, A. Adaptive Real-Time Level-of-Detail-Based Rendering for Polygonal Models, IEEE Transactions on Visualization and Computer Graphics, 3(2):171-183, 1997.