International Journal of CAD/CAM Vol. 6, No. 1, pp. 73~79 (2006)

INTERNATIONAL JOURNAL OF CAD/CAM
www.ijcc.org

Mesh Simplification and Adaptive LOD for Finite Element Mesh Generation

Hiroaki Date[1]*, Satoshi Kanai[1] Takeshi Kishinami[1] and Ichiro Nishigaki[2]

[1] *Graduate School of Information Science and Technology, Hokkaido University, Sapporo, Japan*

[2] *Hitachi Ltd. Mechanical Engineering Research Laboratory, Ibaraki, Japan*

Abstract – In this paper, we propose a new triangular finite element mesh generation method based on simplification of high-density mesh and adaptive Level-of-Detail (LOD) methods for efficient CAE. In our method, mesh simplification is used to control the mesh properties required for FE mesh, such as the number of triangular elements, element shape quality and size while keeping the specified approximation tolerance. Adaptive LOD methods based on vertex hierarchy according to curvature and region of interest, and global LOD method preserving density distributions are also proposed in order to construct a mesh more appropriate for CAE purpose. These methods enable efficient generation of FE meshes with properties appropriate for analysis purpose from a high-density mesh. Finally, the effectiveness of our approach is shown through evaluations of the FE meshes for practical use.

***Key Words*: Finite element analysis, Triangular meshes, Mesh simplification, Adaptive LOD, Mesh property control**

1. Introduction

For efficient product development, it is effective to perform CAE at the stage of the design as early as possible. One of the key issues for realizing this is efficient mesh generation of product shape for FEA. In CAE, several kinds of FE meshes for one product shape are required corresponding to different analysis purposes, such as rough or fine, structural or dynamical analysis, because the processing time and accuracy of analysis depend on several mesh properties, such as the number of elements, quality of elements and mesh density. Therefore, an efficient generation method for several property-controlled meshes of the product shape is desired for achieving efficient CAE.

In this paper, we first propose a method for finite element mesh generation from a high-density triangular mesh based on mesh simplification using edge collapse. The overview of our research is shown in Fig. 1. We assume that the input mesh has high quality and high-density. The approaches for obtaining such meshes are described in Section 3. In contrast with several existing mesh simplification approaches [7, 8, 11], our mesh simplification can explicitly manage the geometric and topological mesh properties required for FE meshes (geometric errors, valence, element shape quality and size) by specifying thresholds for each property. Vertex based mesh hierarchy is constructed as a result of simplification, therefore the number of elements can also be easily increased or decreased (Section 4).

Using mesh hierarchy, adaptive LOD methods for FE mesh generation are also proposed: curvature based and region specification based method. Finally, a global LOD method preserving mesh density distributions is presented in order to control the number of triangle elements without loss of density distributions (Section 5).

In our research, we handle only triangular meshes, and they are generally used in shell analysis. However, the analysis for the 3D solid object requires tetrahedral volume meshes. The volume mesh generation from the surface mesh can be done by using well-known meshing method such as Delaunay or advancing front method [15]. Therefore, we use one of these methods in order to perform volume mesh analysis using the surface mesh resulting from our method as shown in Fig. 1. We assume that the input meshes are limited to conforming triangular meshes.

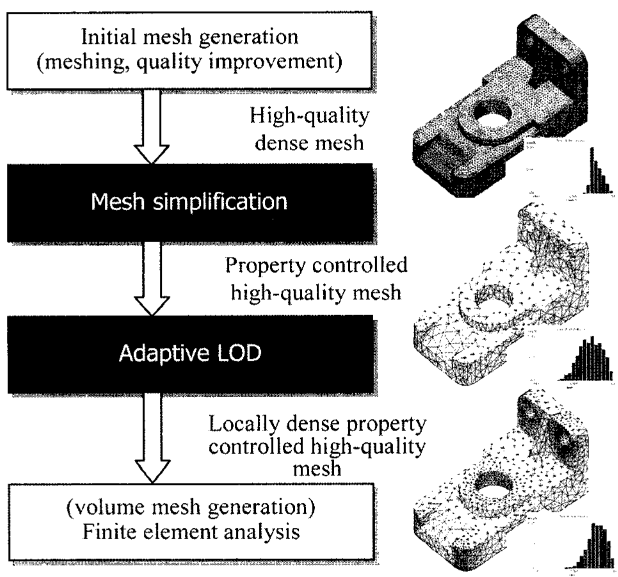


Fig. 1. An overview of our approach.

*Corresponding author:
Tel: +81-11-706-6437
Fax: +81-11-706-6437
E-mail: hdate@ssi.ist.hokudai.ac.jp

2. Related Works

General mesh generation for CAE is done by meshing of a solid model of product shape, and much research on automatic mesh generation has been done [15]. Delaunay, octree and advancing front method are becoming common, and many commercial systems for finite element mesh generation adopt these methods. However, in these mesh generation approaches, the meshes with different properties have to be generated by individually meshing the solid model, and controlling the mesh property therefore becomes time-consuming.

On the other hand, several approaches for generating high-quality meshes from low-quality meshes have been developed, for example, mesh quality improvement based on facet clustering, vertex insertions and local Delaunay triangulations [2, 3], mesh subdivision and simplification based quality improvement [4], and re-meshing using mesh parameterizations [5, 12] and geodesic distances [13]. However, simultaneously satisfying the requirements of controlling the properties for FE mesh was difficult in these approaches: explicit managing of geometric errors and element shape quality, preserving sharp edges and flexible control of the number of elements and local mesh density.

In our approach, mesh simplification is used to control the mesh properties required for FE meshes. Much research for mesh simplification has been conducted in the area of computer graphics [7, 8, 11, 14], and these studies have become common and indispensable to many mesh applications. Their main purpose is to improve rendering speeds [7, 8], to compress model data, to transfer data efficiently [14] and to reduce the cost of several geometric calculations [10]. These methods propose attractive mesh simplification from the aspects of fast computation, accurate geometric approximation and high visual fidelity in simplified mesh and topological changes of the model. However, the quality or properties of mesh suitable for analysis could not be considered in these studies.

Several adaptive LOD methods that can effectively generate locally dense mesh after mesh simplification were also proposed in computer graphics area, for example, view dependent LODs, which generate locally dense mesh at region of interest [6, 9]. They also provided data structures to achieve efficient LOD. These approaches are suitable for keeping the rendering quality, but criteria of LOD is not appropriate for FE mesh generation use. So, we propose new LOD criteria that can generate the mesh efficiently according to the analysis purposes.

3. High-Quality Dense Mesh Generation

As mentioned in section 1, the input of our method is the high-quality dense mesh. Such mesh can be obtained using existing meshing algorithms of the solid model [15]. Meshing the solid model often fails when the model geometry is complex. However, on the contrary, a high-density mesh can be generated more stably from a solid model than the low-density one.

Several existing mesh quality improvement and re-meshing methods described in previous section [2-5, 12, 13] can also be used to generate high-quality mesh from low-quality mesh. If the resulting mesh is coarse, we can apply the simple one-to-four splitting of the triangular elements to it for generating high-quality dense mesh. The quality of the resulting mesh is still high because this type of subdivision can hold the element shape similarities.

4. Mesh Simplification for FE Mesh Generation

4.1 Mesh property metrics

The following metrics are used to evaluate the properties of FE mesh: geometric errors in edge collapse operation [8], element size and element shape quality.

1) Geometric errors in simplification

In order to evaluate the geometric error caused by applying edge collapse $(i, j) \rightarrow k$ using the extension of Garland and Heckbert [7]. The error $d_{ij}(k)$ for an edge (i, j) can be written as follows:

$$d_{ij}(k) = \mathbf{p}_k^T(\mathbf{A}_i + \mathbf{A}_j)\mathbf{p}_k^T + 2(\mathbf{B}_i + \mathbf{B}_j)\mathbf{p}_k + C_i + C_j \quad (1)$$

where,

$$\mathbf{A}_i = \sum_{f\in f^*(i)} \mathbf{n}_f \mathbf{n}_f^T + w_e \sum_{e\in \hat{e}^*(i)} \begin{pmatrix} 1-d_{ex}^2 & -d_{ex}d_{ey} & d_{ex}d_{ez} \\ -d_{ex}d_{ey} & 1-d_{ey}^2 & -d_{ey}d_{ez} \\ -d_{ex}d_{ez} & -d_{ey}d_{ez} & 1-d_{ez}^2 \end{pmatrix} \quad (2)$$

$$\mathbf{B}_i = -\sum_{f\in f^*(i)} (\mathbf{n}_f^T\mathbf{p}_i)\mathbf{n}_f^T + w_e \sum_{e\in \hat{e}^*(i)} (-\mathbf{p}_i + (\mathbf{d}_e^T\mathbf{p}_i)^2\mathbf{d}_e)^T \quad (3)$$

$$C_i = \sum_{f\in f^*(i)} (\mathbf{n}_f^T\mathbf{p}_i)^2 + w_e \sum_{e\in \hat{e}^*(i)} (\mathbf{p}_i^T\mathbf{p}_i + (\mathbf{d}_e^T\mathbf{p}_i)^2). \quad (4)$$

In each equation, the vector $\mathbf{p}_i = (x, y, z)^T$ shows the position of the vertex i, $\mathbf{n}_f$ is the unit normal vector of the triangular element f, and $f^*(i)$ indicates the set of triangular elements connected to vertex i. Vector $\mathbf{d}_e = (d_{ex}, d_{ey}, d_{ez})^T$ shows the unit direction vector of an edge e, and $\hat{e}^*(i)$ is a set of incident edges for the vertex i on the sharp or boundary edges of the model.

The first terms in Eqs (2)-(4) represent the sum of squared distances between the new vertex position $\mathbf{p}_k$ and the planes, which are defined by triangular elements connected to old vertices i and j. The second terms in Eqs (2)-(4) evaluate the squared distances from the new vertex position and the line defined by the edge on feature edges. Here, the feature edges mean sharp, boundary, and non-manifold edges. We add these terms to Garland's measure [7] in order to evaluate error on the edges, because preservation of the feature edges is very important in finite element mesh.

Weight w_e controls the strength of feature edge preservation. In our simple implementation, we identify the sharp edge by evaluating the dihedral angles between two neighboring triangles, and determine the weight w_e according to them (it takes a larger value for sharper edges).

2) Element size

The size of a triangular element is defined as the longest edge length of the triangular element f, and is denoted by $Sz(f) = \max_{e \in e^*(f)} l_e$, where l_e is the length of the edge e and $e^*(f)$ is a set of edges (sides) of f.

3) Element shape quality

The shape quality of a triangular element is evaluated using the stretch, which is one of the measures of element shape quality commonly used in the CAE area. Stretch is calculated by $S_t(f) = (\sqrt{12} / \max_{e \in e^*(f)} l_e) \sqrt{\prod_{e \in e^*(f)} (s-l_e)/s}$, where s is half of the sum of three edge lengths of a triangular element f. The value of $S_t(f)$ is 1 for an equilateral triangle and decreases gradually for a distorted triangle.

4.2 Mesh Simplification Algorithm

Much research for mesh simplification has been conducted in the area of computer graphics [7, 8, 11, 14], and these studies have become common and indispensable to many mesh applications. These methods offer attractive mesh simplification from the aspects of fast computation, accurate geometric approximation and high visual fidelity. However, the quality and other properties of mesh required for FE analysis were not considered.

In our research, we apply the mesh simplification to high-density meshes in order to control the properties of finite element mesh. Comparing with existing simplification methods, the quality of mesh and the other important properties for analysis in the simplified mesh can be explicitly controlled in our method by specifying thresholds for tolerance τ_{TL}, stretch τ_{ST}, element size τ_{SZ} and valence τ_{VL}.

Our mesh simplification consists of the following four steps, as shown in Fig. 2.:

Step1: New vertex position calculation

For all edges (or edges modified by edge collapse), the new vertex positions resulting from edge collapse are calculated.

Step2: Valid edge extraction

All edges (or edges modified by edge collapse) are checked for whether edge collapse can be applied or not. If the edge satisfies the user-specified thresholds for mesh properties, the edge is tagged as valid edge, and becomes candidates for edge collapse. If there is no valid edge, the algorithm terminates.

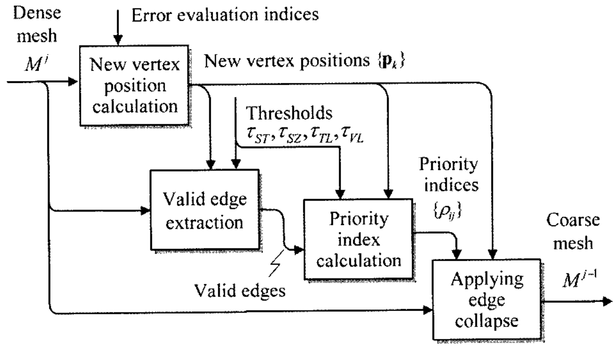


Fig. 2. Mesh simplification algorithm.

Step3: Priority index calculation

For each valid edge, the priority index for determining the order of edge collapse operations is calculated.

Step4: Applying edge collapse

Edge collapse is applied to the edge with the largest priority index, then return to Step 1.

In step 2, the user specified thresholds for mesh properties can be guaranteed. By defining the priority index appropriately so as to evaluate the degree of quality preservation in step 3, high-quality coarse mesh can be obtained through a mesh simplification process. By storing the collapsed edge and neighbour information based on an appropriate data structure, such as progressive mesh structure [8, 9] and multi-triangulations [6], the mesh hierarchy can be constructed, and therefore meshes with different resolutions can be obtained quickly after simplification.

4.2.1 New vertex position calculation

The new vertex positions $\mathbf{p}_k$ resulting from the edge collapse $(i, j) \to k$ are calculated for all edges. In our approach, two positions are adopted as candidates. The first is the midpoint of edge (i, j). This can preserve the mesh quality well in case of high-quality mesh. The second is the vertex position that minimizes the geometric error defined by equation (1). This can be determined by $\mathbf{p}_k = -(\mathbf{A}_i + \mathbf{A}_j)^{-1} (\mathbf{B}_i + \mathbf{B}_j)^T$. Either position is selected from these candidates according to the result of valid edge selection, as described in the next section.

4.2.2 Valid edge extraction

In order to satisfy the user-specified thresholds for tolerance τ_{TL}, stretch τ_{ST}, element size τ_{SZ} and valence τ_{VL}, all edges are checked for whether or not the local mesh near the edge satisfies the thresholds after the edge collapse operation. The edges are identified as valid when they satisfy the following conditions: C1) geometric tolerance: $d_{ij}(k) \leq \tau_{TL}$, C2) lower limit of element shape quality: $\forall f \in f^*(k)$; $St(f) \geq \tau_{ST}$, C3) upper limit of the element size: $\forall f \in f^*(k)$; $Sz(f) \geq \tau_{SZ}$ and C4) upper limit of the valence: $|v^*(i)| + |v^*(j)| - 4 \leq \tau_{VL}$, where, $v^*(i)$ shows a set of neighboring vertices of vertex i. Vertex k is the one generated from edge collapse for the edge $(i,$

j), and evaluation for $f^*(k)$ can be done using the new elements connected to vertex k, which are created temporarily. By evaluating the change of directions of the triangular element normals before and after edge collapse, we avoid mesh foldover [7, 11].

We assume that the input high-density mesh satisfies the upper limit of element size τ_{SZ} and valence τ_{VL}, and lower limit of stretch τ_{ST}. Upper limit conditions for τ_{SZ} (C3) and τ_{VL}(C4) can be mostly satisfied in meshes generated by the high-density mesh generation approaches described in section 2. However, they often do not satisfy condition C2. In such a case, for satisfying τ_{ST}, only edges which do not satisfy C2 are collapsed until all triangular elements satisfy the τ_{ST} in the first step of simplification.

4.2.3 Priority index calculation and edge collapse

The priority index, which indicates the priority of edge collapse applications, is calculated for each valid edge. We define the priority index so that the edge with a larger index has a higher degree of element shape quality preservation and uniformity of element size after collapsing it. The priority index ρ_{ij} for the valid edge (i, j) is defined as $\rho_{ij} = Sz_{DIF} \times St_{MIN} \times St_{AVE}$. The first term Sz_{DIF} denotes the average of the ratios of the size of the triangular elements to the user-specified maximum size in current mesh: $Sz_{DIF} = |f^*(i,j)|^{-1}\sum_{f\in f^*(i,j)}\tau_{SZ}/S_z(f)$, where $f^*(i,j)$ shows the triangles connected to edge (i,j). This makes the element size uniform. The second and third terms, St_{MIN} and St_{AVE}, respectively represent the minimum and the average of stretches of newly generated elements by collapsing the edge (i,j), and are defined by $St_{MIN} = \min_{f\in f^*(k)} St(f)$ and $St_{AVE} = |f^*(k)|^{-1}\sum_{f\in f^*(k)} St(f)$. Applying the edge collapse to the edge (i,j) with larger ρ_{ij} means that the elements with better quality and more uniform size are generated locally in the resulting coarse mesh. Therefore, edge collapse is applied to the edge with the largest priority index.

5. Adaptive LOD for FE Mesh Generation

Locally detailed mesh is needed for more accurate analysis after the rough analysis. In this section, adaptive LOD methods for FE mesh generation are described to obtain the locally detailed mesh. The adaptive methods are done after global, uniform density control using the mesh simplification method described above.

5.1 Local LOD

In order to achieve efficient local LOD, the data structures based on vertex hierarchy [9] and directed graph [6] are proposed. We adopt vertex hierarchy structure based on binary tree [9]. In this structure, parent node corresponds to a vertex generated by edge collapse, and its two children are the collapsed vertices.

All triangular elements in the meshes at different resolution levels generated by the method in section 3 satisfy the user-specified threshold for stretch. Therefore, in order to avoid expensive quality recalculations during changing the LOD, we adopt the triangular elements that appeared in the simplification process for the ones in every adaptive LOD. To achieve this, at every node, we make the pointers to the neighboring vertices (*vstars*, Fig. 3(a)) when edge collapse is applied during simplification. For the criterion of density management, the average size of *vstars* (*e_size*) is also assigned to each node at the same time. Finally, each node has a front flag, which represents whether it is on the active mesh or not. The node with true front flag (front node) indicates that it is on the active mesh.

Vertex split is the operation where the front moves down from a node to their children in the tree, as shown in Fig. 3(b). This can be applied when all *vstars* of the node to be processed are included in front nodes. If it is not applicable, by traversing the tree from the *vstars* to their parents, and by applying vertex split iteratively to their parents, all *vstars* can be added to the front nodes and vertex split of the target node can be done. On the other hand, edge collapse moves up the front from collapsed nodes to their parent. This is done when all the *vstars* of collapsed nodes are included in the front ones. If they are not, similarly to the vertex split, by traversing *vstars* to the children and applying edge collapse repeatedly to them, edge collapse for the target vertex pair in the current front nodes can be done.

5.2 Index of mesh density and execution of local LOD

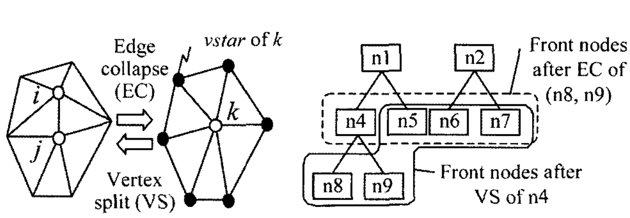
Mesh density can be managed by the threshold T_S for element size. This threshold is assigned to the vertices. Vertex split is done when the value T_S is smaller than the *e_size* of the node. As a result, the elements whose sizes are smaller than threshold T_S can be reconstructed locally. Also, using the T_S, we can specify the mesh density by the ratio between element sizes. When the ratio between the size of arbitrary element and target element sizes is defined, we first calculate the T_S using each element size, then the resulting T_S can be used for mesh density control.

5.3 Adaptive LOD

5.3.1 Local density control

a) Curvature based LOD

Local mesh is detailed according to the curvature of shape. First, mesh curvature at each vertex of input high-



(a) EC, VS and *vstar* (b) An example of vertex hierarchy

Fig. 3. Edge collapse and vertex hierarchy.

density mesh is calculated using the curvature evaluation method (In our implementation, we adopt the method of Alliez [1]), and maximum curvature κ_i is stored at each vertex. Then, in the edge collapse at the stage of mesh simplification, the new vertex k inherits the largest curvature from the collapsed vertices ($\kappa_k \leftarrow \max_{m\in(i,j)}\kappa_m$). Curvature based adaptive LOD is done by defining the T_S according to κ_i at each front node, and by executing the local LOD method described in section 4.2. In our approach, the maximum curvature value of the input mesh corresponds to $T_S = 0$ (i.e. original small elements are generated near the high curvature region, such as sharp edges and corners), and T_S gradually becomes larger depending on the value of κ_i.

b) Region specification based LOD

High-density mesh can also be generated in the user-specified region of interest. This is done by directly specifying the thresholds T_S of certain elements or vertices in the interactively-selected region, and then by executing local LOD. In this LOD, we have to inherit the T_S during vertex split operations. To preserve the position of user-specified region on the mesh, only a nearest child vertex from original T_S-specified position inherit T_S from its parent.

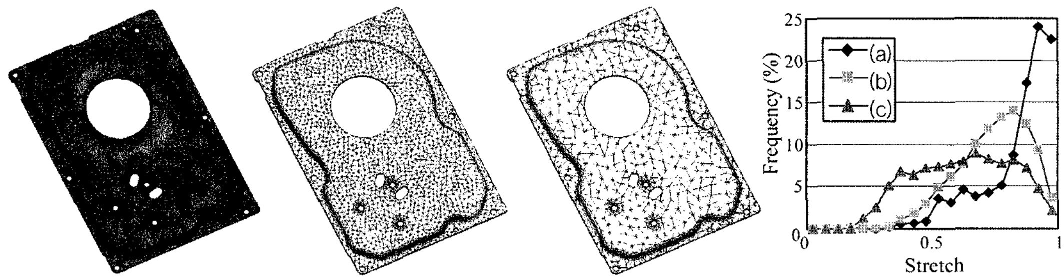
5.3.2 Global LOD with mesh density preservation

In some situations, users want to increase or decrease the total number of elements without loss of original density distributions. To meet this requirement, a global LOD method which can preserve mesh density distribution is proposed. First, a new variable S_{REF} is introduced for each vertex in the tree. For each front node i, e_size is substituted for its S_{REF} ($S_{REF}(i) \leftarrow e\_size(i)$). Children of front nodes inherit the S_{REF} directly from the front nodes ($S_{REF}(i)$, $S_{REF}(j) \leftarrow S_{REF}(k)$), and for each parent k of front nodes, we assign the average of two S_{REF} of its children i and j to it ($S_{REF}(k) \leftarrow (S_{REF}(i) + S_{REF}(j))/2$). According to this rule, we can assign S_{REF} to all nodes in the tree. Then, we calculate the ratio r of element size from front for all nodes in the tree as $r = e\_size/S_{REF}$. Making r uniform in the mesh means that the density distribution of the initial mesh is preserved in the different meshes in LOD. Therefore, for reducing the number of elements, edge collapse is applied to the vertex pair which has the parent with the smallest r. On the other hand, the vertex split is applied to the vertex with the largest r for increasing the number of elements.

6. Results and Evaluations

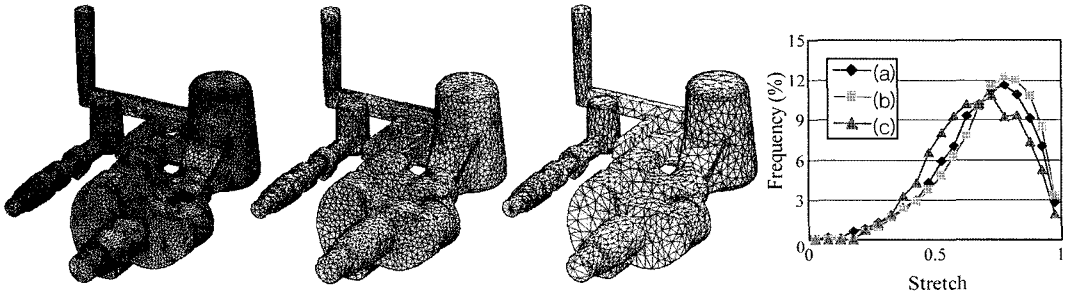
The meshes obtained by the proposed mesh simplification methods from high-quality dense meshes are shown in Figs. 4-6. The dimensions and the number of triangular elements of each model are shown in Table 1.

The input mesh of hard disk cover in Fig. 4(a) was generated by applying advancing front method [15] to the solid model. From (b) and (c), it was confirmed that high-quality coarse mesh could be obtained by our method. Fig. 5 shows the resulting casting part model meshes generated from the mesh quality improvement method [4] of the STL model. Our approach can also be applied to the mesh obtained from 3D scanning. The results for the crankshaft mesh scanned by industrial X-



(a) Original mesh (b) Simplified mesh 1 (c) Simplified mesh 2 (d) Stretch distribution

Fig. 4. Simplification results of hard disk cover model.



(a) Original mesh (b) Simplified mesh1 (c) Simplified mesh2 (d) Stretch distribution

Fig. 5. Simplification results of casting part model.

Table 1. Mesh properties.

Model (Dim : H × W × D)	State	Number of elements	Thresholds (τ_{TL}, τ_{ST}, τ_{SZ}, τ_{VL})	Stretch (ave, min)	Size (max)	Valence (max)
Hard disk cover : Fig. 4. (101.2 × 145.6 × 4.9)	(a) Original	78,488	---	0.84, 0.03	2.14	9
	(b) Simplified 1	12,000	(0.15, 0.2, 15.0, 12)	0.74, 0.25	5.22	10
	(c) Simplified 2	6,000	(0.15, 0.2, 15.0, 12)	0.62, 0.20	10.75	12
Casting part : Fig. 5. (516.5 × 398 × 277)	(a) Original	30,000	---	0.69, 0.02	15.93	10
	(b) Simplified 1	14,000	(1.0, 0.2, 50.0, 10)	0.71, 0.20	19.36	10
	(c) Simplified 2	7,000	(1.0, 0.2, 50.0, 10)	0.66, 0.20	31.95	10
Crankshaft : Fig. 6. (198.6 × 155.4 × 648.8)	(a) Original	974,754	---	0.63, 0.03	1.73	12
	(b) Simplified 1	50,000	(5.0, 0.2, 30.0, 15)	0.67, 0.20	15.4	13
	(c) Simplified 2	20,000	(5.0, 0.2, 30.0, 15)	0.62, 0.20	25.32	10
Hard disk cover : Fig. 8. (101.2 × 145.6 × 4.9)	(a) Curvature	27,630	---	0.76, 0.21	8.09	10
	(b) Region	8,782	---	0.67, 0.20	10.75	12
	(c) Global	13,550	---	0.74, 0.25	5.38	10

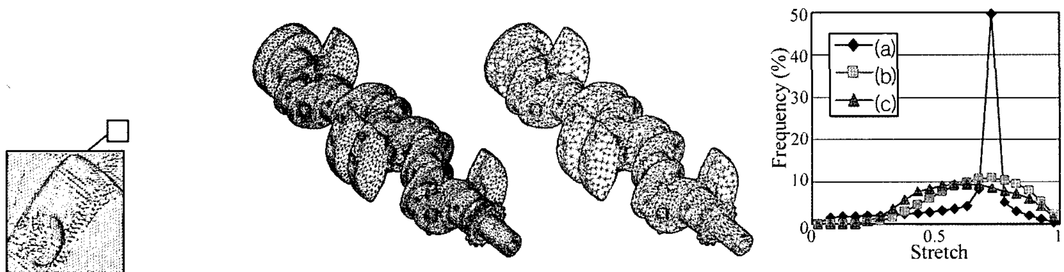
ray computed tomography device are shown in Fig. 6. Similar to the previous results, high quality coarse mesh could be obtained from dense meshes using our quality preserving mesh simplification. Used simplification parameters (thresholds for properties), the evaluations of minimum stretch, maximum size and valence are summarized in Table 1. The stretch distributions of each mesh are also shown in Figs. 4-6(d). From these results, it was shown that the user-specified thresholds for element shape quality and size are satisfied.

The effectiveness of the feature edge preservation measure described in section 3 is shown in Fig. 7. Fig. 7(b) shows the result of mesh simplification without feature edge preservation, and (c) shows the one with the feature edge preservation. The resulting meshes have similar number of elements (about 10k triangles). From the figures, it was confirmed the shape of the feature edges could be preserved well using our approximation error metric for feature edges.

Adaptive LOD was applied to the mesh in Fig. 4(c) and the results are shown in Fig. 8. Fig. 8(a) and (b) show the results of curvature based and region specification based LOD. The region was specified around the large hole. Fig. 8(c) is the result of density preserving global LOD of (b). From these figures, it was confirmed that local density control and density-preserving global LOD were executed appropriately while quality thresholds were satisfied in the resulting meshes from Fig. 8(d) and Table 1.

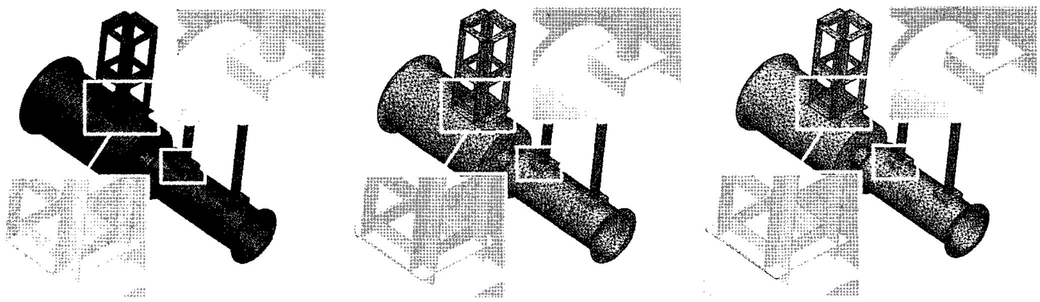
7. Conclusions

In this paper, a new FE mesh generation method from triangular mesh using mesh simplification and adaptive LOD was proposed. By introducing valid edge selection and new criterion for edge collapse in the simplification



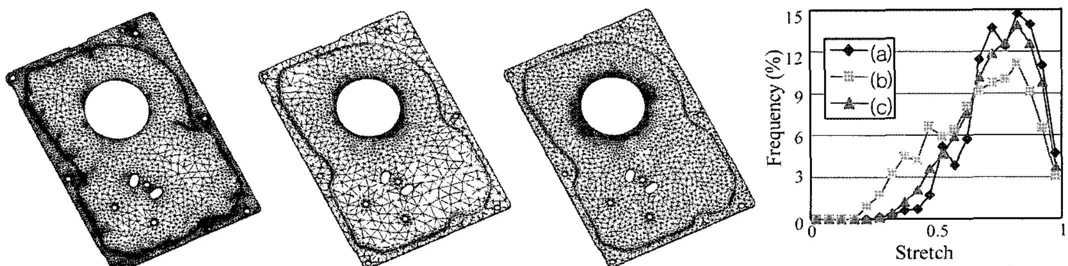
(a) Original mesh (b) Simplified mesh1 (c) Simplified mesh2 (d) Stretch distribution

Fig. 6. Simplification results of crankshaft model.



(a) Original Mesh (b) Mesh without feature edge preservation (c) Mesh with feature edge preservation

Fig. 7. Effect of feature edge preservation.



(a) Curvature based (b) Region based (c) Global LOD (d) Stretch distribution

Fig. 8. Adaptive LOD for hard disk cover model.

algorithm, FE meshes that satisfy user-specified thresholds for tolerance, element quality, size and valence, could be obtained from dense mesh. Adaptive LOD methods based on curvature and region specification, and global LOD method preserving mesh density distributions were also proposed using vertex hierarchy and additional attributes. Finally, the effectiveness of these methods was proved by the mesh generation of actual product shapes.

Our future work is to extend our algorithm to tetrahedral mesh generation based on volume mesh simplification. In current approach, selective refinement or coarsening according to the analysis results can be done using region specification LOD, however re-volume-meshing is required for analysis of the volume mesh. The extension of our approach to the volume mesh allows us to achieve more smooth sequential analysis operations in several phases of CAE.

8. Acknowledgements

This work was financially supported by the grant-in-aid of Intelligent Cluster Project (Sapporo IT Carrozzeria) founded by MEXT.

References

[1] Alliez, P., Cohen-Steiner, D., Devillers, O., Lévy, B. and Desbrun, M. (2003), Anisotropic Polygonal Remeshing, Proceedings of SIGGRAPH2003, 485-493.

[2] Bechet, E., Cuilliere, J.C. and Trochu, F. (2002), Generation of a finite element MESH from stereolithography (STL) files, *Computer-Aided-Design*, 34, 1-17.

[3] Bianconi, F. (2002), Bridging the gap between CAD and CAE using STL files, *International Journal of CAD/CAM*, 2(1), 55-67.

[4] Date, H., Kanai, S. and Kishinami, T. (2004), Mesh Quality Improvement using Mesh Subdivision and Simplification, Proceedings of 3D Modelling 2004 (NUM3D).

[5] Floater, M.S. (1997), Parameterization and Smooth Approxi-mation of Surface Triangulations, *Computer Aided Geometric Design*, 14(3), 231-250.

[6] Floriani, L.D., Magillo, P., Morando, F. and Puppo, E. (2000), Dynamic View-Dependent Multiresolution on a Client-Server Architecture, *Computer-Aided-Design*, 32, 805-823.

[7] Garland, M. and Heckbert, P.S. (1997), Surface Simplification using Quadric Error Metrics, Proceedings of SIGGRAPH 97, 209-216.

[8] Hoppe, H. (1996), Progressive meshes, Proceedings of SIGGRAPH 96, 99-108.

[9] Hoppe, H. (1997), View-dependent refinement of progressive meshes, Proceedings of SIGGRAPH 97, 189-198.

[10] Karabassi, E.A., Papaioannou, G., Fretzagias, C. and Theoharis, T. (2003), Exploiting multiresolution models to accelerate ray tracing, *Computer & Graphics*, 27, 91-98.

[11] Luebke, D.,, Reddy, M., Cohen, J.D., Varshney, A., Watson, B. and Huebner, R. (2003), *LEVEL of DETAIL FOR 3D GRAPHICS*, Morgan Kaufmann.

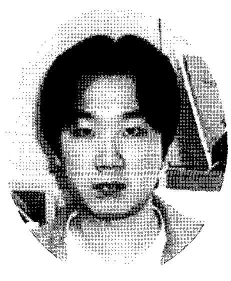
[12] Sheffer, A. and Sturler, E.D. (2001), Parameterization of faceted surfaces for meshing using angle-based flattening, *Engineering with Computers*, 17(3), 326-337.

[13] Sifri, O., Sheffer, A. and Gotsman, C. (2003), Geodesic-based Surface Remeshing, Proceedings of the 12th International Meshing Roundtable, 189-199.

[14] Taubin, G., Gueziec, A., Hom, W. and Lazarus, F. (1998), Progressive Forest Split Compression, Proceedings of the 25th annual conference on computer graphics and interactive techniques, 123-132.

[15] Topping, B.H.V., Muylle, J., Iványi, P., Putanowicz, R. and Cheng, B. (2004), *Finite Element Mesh Generation*, SAXE-COUBURG PUBLICATIONS.

Hiroaki Date is an instructor of graduate school of information science and technology, Hokkaido University. He received his PhD in Engineering from Hokkaido University in 2003. His research interest includes multiresolution modeling and its application to product development.



Hiroaki Date