Fast matching, combinations extraction and configuration of mesh models using graph-based feature representation

Kensuke Sato [1], Hiroaki Date [1], Masahiko Onosato [1]

(1) : Hokkaido University
Kita-14, Nishi-9, Kita-ku, Sapporo 060-0814, JAPAN
Phone: +81-11-706-6438
E-mail : satou@dse.ssi.ist.hokudai.ac.jp, { hdate, onosato }@ssi.ist.hokudai.ac.jp

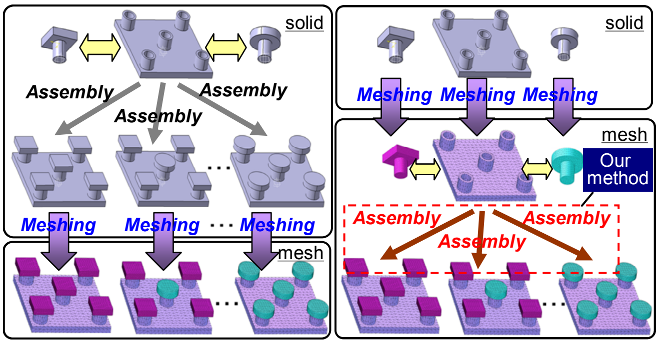
Abstract: In this paper, we propose a method for matching, combinations extraction and configuration of triangular mesh models for assembly and geometric modeling. Given two mesh models, a graph is first generated for each model. This graph, called a "feature-graph", represents abstract geometry information and connections of segments (local regions) for the model. Next, a set of pairs of segments which can be matched are extracted by performing sequential matching tests on the graphs. Finally, using the resulting pairs, the configuration between the models is determined by phased registrations keeping previous contact condition. An example of application to assembly modeling is included.

Key words: Mesh model, Assembly, Feature representation, Matching, Configuration

1- Introduction

The uses of mesh models in engineering have increased in recent years, and more powerful techniques for handling mesh models are now needed. In the future, it will be particularly important to resolve problems involving the matching and configuration of more than one mesh models. For example, the matching and configuration of mesh models will realize efficient CAE because assembly generation using solid models is unnecessary and the number and complexity of meshing of solid model can be reduced (Figure1). They may be able to reconstruct the assembly information defined in a CAD system from the individual mesh models generated by the CAD (STL, etc.). Furthermore, fast matching and configuration are also useful for assembly generation and geometric modeling using database queries.

The matching and configuration of mesh models / point clouds has been researched mainly for registration in reverse engineering, and many methods have been proposed [BR1]. The main aim of these methods is the automatic matching of multiple scanning data. More recently devised methods can accurately register models with high-level noise and outliers. A method for reassembling fractured 3D objects has also been proposed [H1], which is similar to our own work. However, the methods that have been devised to date often require high computational costs in order to deal with noise. Moreover, the extraction of combinations of matching pairs is not one of their functions.



(a) Current process (b) Realized process

Figure 1: Generation of assembly models for CAE.

In this paper, we propose a method for fast matching, combinations extraction and configuration of mesh models using graph-based feature representation. In our current research, we handle triangular mesh models of mechanical parts obtained by finite element meshing of CAD models or quality improvement of arbitrary meshes without scanning noise. And we assume that the mesh models match at planar or cylindrical regions, because planar or cylindrical contacts often appear in assembly models of engineering products.

2- Matching, combinations extraction and configuration

2.1- Overview

Given two mesh models as inputs, our method for matching, combinations extraction and configuration is composed of the following three steps (Figure 2).

1- **Feature-graph generation**: A graph (feature-graph) representing abstract geometry information and connections of segments (local regions) on the model is generated for each model.

2- **Matching and combinations extraction**: A set of matching pairs of segments (a combination) are extracted by sequential matching tests using the graphs. Possible combinations are extracted.
3- **Configuration**: Using the resulting pairs, the configuration between the models is determined by phased registration keeping previous contact condition.

2.2- Feature-graph generation

To enable efficient matching, combinations extraction, and configuration, the feature-graph is introduced. In this graph, a node indicates a segment, and an arc indicates a neighboring relationship between two segments (Figure 2). Node n has the attributes ($\mathbf{g}^n$, σ^n, $\mathbf{e}^n$) and surface type (*plane / cylinder / others*) of the corresponding segment r^n, where $\mathbf{g}^n = (\bar{\kappa}^n_{\max}, \bar{\kappa}^n_{\min})$ and σ^n are the averages and variance of the principle curvatures of vertices in r^n, respectively. $\mathbf{e}^n$ is the representative vector of r^n. An arc a connecting nodes m and n has an attribute θ_a which is the average dihedral angle of the edges between r^m and r^n. This graph is generated independently for each of the two given mesh models (Figure 2).

Although many mesh segmentation methods have been proposed [KT1] [RB1], we used dihedral angle-based segmentation for the sake of simplicity. In this method, sharp edges become segment boundaries. The method proposed in [A1] was used to calculate curvature. Surface type is determined by thresholding of node's attribute. If $|\bar{\kappa}^n_{\max}| < \tau_{\kappa\max}$, the surface type of the node is *plane*, if $|\bar{\kappa}^n_{\max}| \geq \tau_{\kappa\max} \wedge |\bar{\kappa}^n_{\min}| < \tau_{\kappa\min} \wedge \sigma^n < \tau_\sigma$, it is *cylinder*, and otherwise, it is *others*. If the type of n is plane, $\mathbf{e}^n$ is the normal vector of the plane, if it is cylinder, $\mathbf{e}^n$ is the axis vector of the cylinder. These are calculated from principal direction vectors which are the byproducts of the curvature calculation.

2.3- Matching and combinations extraction

From the resulting two graphs, a set of pairs of matching nodes are extracted using sequential tests. The tests are composed basically of *node matching* and *arc matching*.

Node matching tests whether or not the segments corresponding to two given nodes ($n1$, $n2$) match geometrically, using the following conditional expression:

$$|\bar{\kappa}^{n1}_{\max} + \bar{\kappa}^{n2}_{\max}| < \varepsilon_{\kappa\max} \wedge |\bar{\kappa}^{n1}_{\min} + \bar{\kappa}^{n2}_{\min}| < \varepsilon_{\kappa\min} \wedge |\sigma^{n1} - \sigma^{n2}| < \varepsilon_\sigma \quad (1)$$

Here, $\varepsilon_{\kappa\max}$, $\varepsilon_{\kappa\min}$ and ε_σ are thresholds defined by the user. Arc matching tests whether or not the segment boundaries corresponding to two given arcs ($a1$, $a2$) match geometrically using the conditional expression $|\theta_{a1} + \theta_{a2}| < \varepsilon_\theta$, where ε_θ are thresholds defined by the user.

The tests start with a pair of the nodes from each feature-graph. This pair can be specified by the user; otherwise, all pairs are used in sequence. First, node matching is applied to the two given nodes. If the pair of nodes passes the test, then arc matchings are applied to the pairs of incident arcs for each node. If the pair of arcs passes the test, node matching is applied to a pair of untested nodes for each arc. This process is repeated until no more pairs of nodes or arcs which passes matching are found (Figure 3). Finally, passing pairs of the nodes are extracted as a combination. If multiple initial pairs are used, some combinations are obtained.

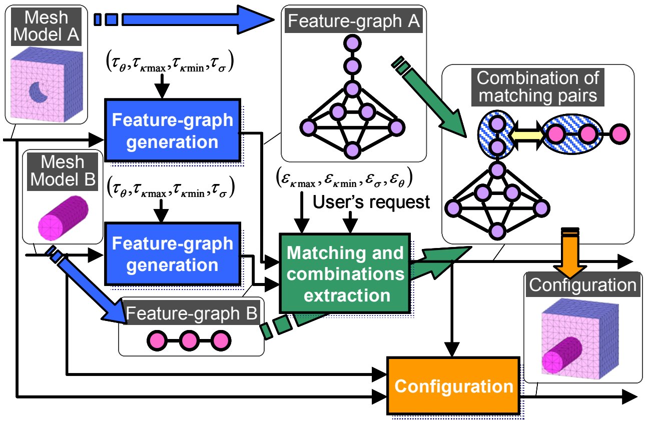


Figure 2: Algorithm for our method.

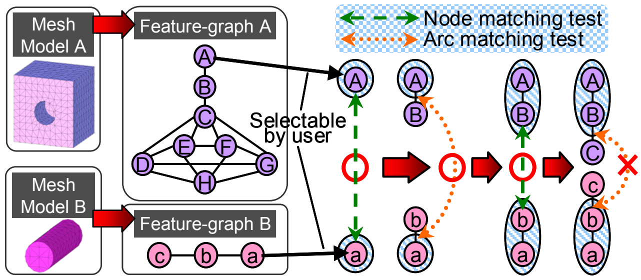


Figure 3: Matching process.

2.4- Configuration

Finally, using the resulting pairs, the configuration between the models is determined by phased registrations keeping previous contact condition. The registrations are implemented by contacting pairs of segments on two given models in passing order of matching tests.

For planar contact between two given nodes ($n1$, $n2$), the rotation and translation are applied so as to satisfy $\mathbf{e}^{n1} = -\mathbf{e}^{n2}$ and make the distance between the planes zero. For cylindrical contact, they are applied so as to satisfy $\mathbf{e}^{n1} = \pm\mathbf{e}^{n2}$ and make the cylindrical segments coaxial.

The configuration process is shown in Figure 4. Registration is first performed without constraints, and the following registrations are done with constraints for keeping previous contact condition. Once registration is completed, a possible rotational axis (directional vector **a**, based point **b**) and translational vectors ($\mathbf{m}_1$, $\mathbf{m}_2$) for keeping current contact condition are generated. In the following registration, rotation using the possible rotational axis and translation along possible translational vectors are performed, and then they are updated to keep new contact condition using contact state transition similar to [MI1] and an update table. Figure 5 shows contact state transition in the case of planar or cylindrical contact, and Table 1 shows the transition conditions for contact state and the update rules for the axis and vectors.

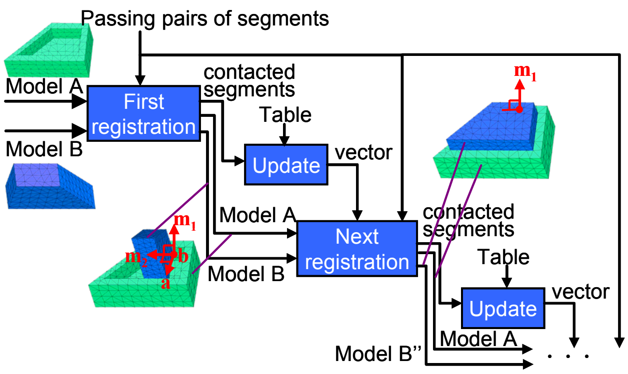


Figure 4: Configuration process.

3- Results and discussions

Figure 6 is an example of extracted combinations which can be matched. In this example, all expected combinations were completely extracted. To evaluate the significance of each combination, for example, the number of contact segments and their surface area, and to show it to the user will be required for using the resulting combinations in some applications. Figure 7 is an example of assembled engine parts. In this example, matching and configuration were performed successively by two parts, and the user selected the initial pairs interactively. The order of configuration is shown in the figure. The positions and orientations for the remaining DOF in a certain contact state were interactively determined by the user. An appropriate configuration was thus obtained. In all examples, the processing time was less than 2.0 sec. (P4-3.4GHz). Collisions are not considered in the current method, therefore, collisions between parts occurred in some configurations. Adopting the collision test will realize more accurate combinations extraction and configuration. In addition, although we handled the meshes obtained by meshing of CAD models, it will be possible to apply our approach to the meshes obtained by 3D scan using robust curvature estimation and robust segmentation methods. These extensions will be included in future works.

4- Conclusions

This paper proposed a method for the matching, combinations extraction, and configuration between mesh models using graph-based feature representation. In simple cases, all expected combinations were successfully extracted, and our method was useful for assembly modeling using mesh models. Further works will focus on collision-free configuration, application to 3D scanned meshes, and the simultaneous processing of multiple models.

5- References

[A1] Pierre Alliez, et.al. Anisotropic Polygonal Remeshing, Proc. SIGGRAPH2003: 485-493, 2003

[BR1] Benedict Brown and Szymon Rusinkiewicz. Global Non-Rigid Alignment of 3-D Scans, ACM Transactions on Graphics. 26(3) : 21:1-21:9, 2007

[H1] Qi-Xing Huang, et.al. Reassembling Fractured Objects by Geometric Matching, Proc. SIGGRAPH2006: 569-578, 2006

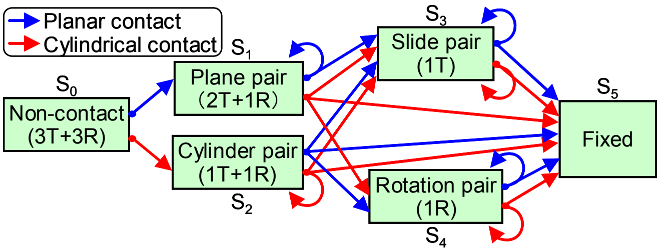


Figure 5: Transition of contact state.

Current State	Contact type	Transition Condition	Next State	Update a	Update b	Update m_1	Update m_2
S_0	P	——	S_1	$\mathbf{e}$	$\mathbf{p}_1$	$\mathbf{e}_{\perp 1}$	$\mathbf{e}_{\perp 2}$
	C	——	S_2	$\mathbf{e}$	$\mathbf{p}_2$	$\mathbf{e}$	ϕ
S_1	P	$\mathbf{a}\times\mathbf{e}=\mathbf{0}$	S_1	$\mathbf{a}$	$\mathbf{b}$	$\mathbf{m}_1$	$\mathbf{m}_2$
		$\mathbf{a}\times\mathbf{e}\neq\mathbf{0}$	S_3	ϕ	ϕ	$\mathbf{a}\times\mathbf{e}$	ϕ
	C	$\mathbf{a}\cdot\mathbf{e}=0$	S_3	ϕ	ϕ	$\mathbf{e}$	ϕ
		$\mathbf{a}\times\mathbf{e}=\mathbf{0}$	S_4	$\mathbf{e}$	$\mathbf{p}_2$	ϕ	ϕ
		$\mathbf{a}\cdot\mathbf{e}\neq0$, $\mathbf{a}\times\mathbf{e}\neq\mathbf{0}$	S_5	ϕ	ϕ	ϕ	ϕ
S_2	P	$\mathbf{a}\cdot\mathbf{e}=0$	S_3	ϕ	ϕ	$\mathbf{m}_1$	ϕ
		$\mathbf{a}\times\mathbf{e}=\mathbf{0}$	S_4	$\mathbf{a}$	$\mathbf{b}$	ϕ	ϕ
		$\mathbf{a}\cdot\mathbf{e}\neq0$, $\mathbf{a}\times\mathbf{e}\neq\mathbf{0}$	S_5	ϕ	ϕ	ϕ	ϕ
	C	$\mathbf{a}\times\mathbf{e}=\mathbf{0}$, $\mathbf{p}_2+t\mathbf{e}=\mathbf{b}$	S_2	$\mathbf{a}$	$\mathbf{b}$	$\mathbf{m}_1$	ϕ
		$\mathbf{a}\times\mathbf{e}=\mathbf{0}$, $\mathbf{p}_2+t\mathbf{e}\neq\mathbf{b}$	S_3	ϕ	ϕ	$\mathbf{m}_1$	ϕ
		$\mathbf{a}\times\mathbf{e}\neq\mathbf{0}$	S_5	ϕ	ϕ	ϕ	ϕ
S_3	P	$\mathbf{m}_1\cdot\mathbf{e}=0$	S_3	ϕ	ϕ	$\mathbf{m}_1$	ϕ
		$\mathbf{m}_1\cdot\mathbf{e}\neq0$	S_5	ϕ	ϕ	ϕ	ϕ
	C	$\mathbf{m}_1\times\mathbf{e}=\mathbf{0}$	S_3	ϕ	ϕ	$\mathbf{m}_1$	ϕ
		$\mathbf{m}_1\times\mathbf{e}\neq\mathbf{0}$	S_5	ϕ	ϕ	ϕ	ϕ
S_4	P	$\mathbf{a}\times\mathbf{e}=\mathbf{0}$	S_4	$\mathbf{a}$	$\mathbf{b}$	ϕ	ϕ
		$\mathbf{a}\times\mathbf{e}\neq\mathbf{0}$	S_5	ϕ	ϕ	ϕ	ϕ
	C	$\mathbf{p}_2+t\mathbf{e}=\mathbf{b}$	S_4	$\mathbf{a}$	$\mathbf{b}$	ϕ	ϕ
		$\mathbf{p}_2+t\mathbf{e}\neq\mathbf{b}$	S_5	ϕ	ϕ	ϕ	ϕ

※ $\mathbf{e}={}^{A}\mathbf{e}^{r}$. P and C stand for planar and cylindrical. $\mathbf{p}_1$ is an arbitrary vertex on the contact planar segment of the model A. $\mathbf{p}_2$ is a point on the axis of contact cylindrical segment of the model A.

Table 1: Transition rule and vector update.

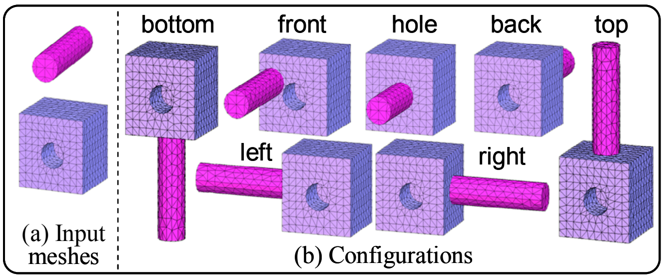


Figure 6: Extracted combinations.

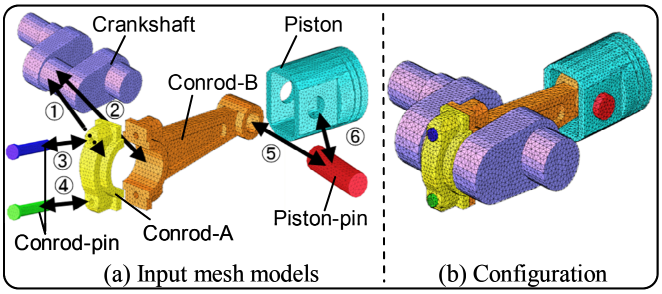


Figure 7: Assembled engine parts.

[KT1] Sagi Katz and Ayellet Tal. Hierarchical Mesh Decomposition using Fuzzy Clustering and Cuts, Proc. SIGGRAPH2003: 955-961, 2003

[MI1] Jun Miura and Katsushi Ikeuchi. Task-Oriented Generation of Visual Sensing Strategies in Assembly Tasks, IEEE Trans. on Pattern Analysis and Machine Intelligence, 20(2): 126–137, 1998

[RB1] Anshuman Razdan and MyungSoo Bae. A hybrid approach to feature segmentation of triangle meshes, Computer-Aided Design, 35: 783-789, 2003