Proceedings of DETC'04
ASME 2004 Design Engineering Technical Conferences and Computers and Information in Engineering Conference
September 28-October 2, 2004, Salt Lake City, Utah USA

DETC2004/CIE-57676

VOLUME-BASED CUT-AND-PASTE EDITING FOR EARLY DESIGN PHASES

Hiroshi Masuda
Graduate School of Engineering
The University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan
e-mail: masuda@nakl.t.u-tokyo.ac.jp

Yoshiyuki Furukawa
Digital Manufacturing Research Center
National Institute of Advanced Industrial Science and Technology
1-2 Namiki, Tsukuba-shi, Ibaraki 305-8561, Japan
e-mail: y-furukawa@aist.go.jp

Yasuhiro Yoshioka
Graduate School of Engineering
The University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan
e-mail: yeshero@nakl.t.u-tokyo.ac.jp

ABSTRACT

Since product concepts are frequently changed in the early stage of design, the creation of its rough models is useful for communication among the design team. In this paper, we propose modeling operations based on a volume-based cut-and-paste method for generating rough 3D models using existing 3D models. Cut-and-paste editing extracts a characteristic feature from a source model and copies it to a target model. Our method allows pasting a wide variety of features that may have overhangs and handles, while most existing methods cannot manage such shapes. To realize such a volume-based cut-and-paste technique, this paper introduces a new method that consists of mesh segmentation and surface/volume fitting. In our method, a feature region and its surrounding area are separated from the user-specified area, and they are used for generating a parametric volume that involves the feature region. In feature pasting, parametric volumes are deformed and pasted to the target model and then features inside the volume are adaptively deformed and pasted to the target. On the basis of this cut-and-paste method, we propose modeling operations that enable feature registration, feature removal, feature pasting, and texture pasting. These operations were implemented and demonstrated using example models. The result shows our operations can effectively generate new models by removing and reusing partial shapes in existing models.

Keywords: CAD, 3D Modeling, Cut-and-Paste, Conceptual Design, Volume

1. INTRODUCTION

After the product specification is determined based on customer needs, rough models are often created by expert designers in the early stage of product development. Since a lot of uncertainties exist in this stage, the designer creates only rough models and frequently refines them. Such rough models are usually represented in concept sketches or three-dimensional models, and they are used for the design evaluation among the development team. This early evaluation is essentially important to produce competitive products, because most of the values including cost, weight, quality, manufacturability, assemblability, maintenancability, etc., are strongly constrained by decisions in this phase.

The early design review is helpful to prevent inconsistency in design, and to shorten the total development time. Since the design concepts must be understood by a wide variety of specialists from different perspectives, rough models, such as 2D sketches or 3D models, are useful to enrich communication with top management, vendors, team members, customers, and investors [1]. In this phase, however, it may not be reasonable to spend a lot of time for creating detail 3D models, because the design concepts are very often changed or discarded.

One way to rapidly create 3D models is to prepare design templates that produce design variations. This capability is provided by some commercial CAD systems [2,3]. However, these templates cannot deal with variations that are not previously defined. In addition, it is a very expensive process to reveal tacit design knowledge and embed it into a template model.

Another way is to produce product variations on the base of existing 3D models of similar products or 3D models scanned from real products [4-16]. In this paper, we discuss modeling operations for this approach. Since strict surface representation is not necessarily required to convey the design concept in the early stage, we confine our discussion to mesh models, which are suitable for 3D models that are both approximated from precise CAD models and scanned from real products.

Several techniques have been developed to modify existing models [4-16]. Shape deformation [4-6], cut-and-paste editing [7-13], and 3D painting [14-16] seem to be promising techniques for this purpose. While the shape deformation technique modifies coordinates of 3D models without changing the connectivity of meshes, the cut-and-paste editing removes or adds detail features by changing the connectivity. On the other hand, 3D paintings, which draw 2D figures on a 3D model, change colors on the model without changing the coordinates and the mesh connectivity.

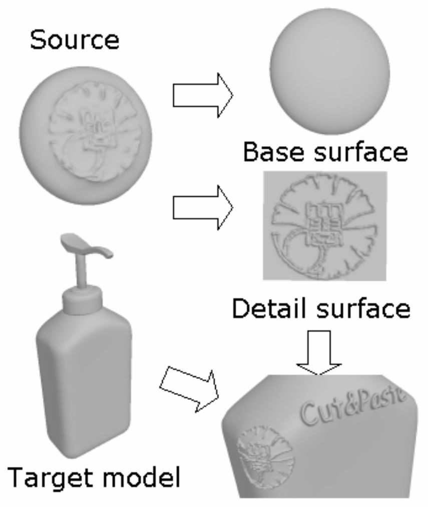


Figure 1. Cut-and-paste editing

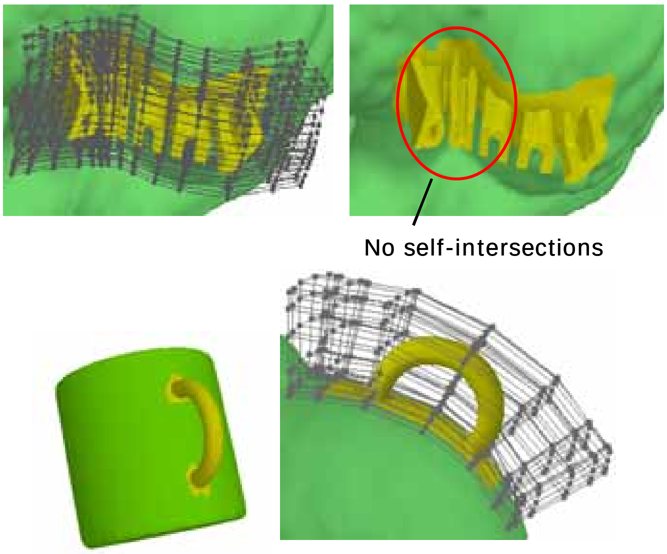


Figure 2. Examples of volume-based pasting.

We mainly discuss cut-and-paste editing in this paper because shape deformation is a relatively mature technique. In this paper, we propose a new volume-based cut-and-paste editing method and then apply it to modeling operations for modifying existing models. Our modeling operations include feature removal, feature pasting, feature registration, and 3D painting.

Cut-and-paste editing extracts a characteristic feature from a source model and copies it to a target model. We simply call such a characteristic feature a *feature*. As shown in Figure 1, the region selected by the user is separated into the base surface and the detail surface, and only the detail surface is used as a feature to be pasted. Suppose the detail feature is thin and does not involve overhangs; the base surface can be calculated as the smooth approximation of the original surface by applying surface fitting or polygonal simplification. The detail surface, on the other hand, is obtained as the difference of geometry between the original surface and the base surface. The difference is measured in the normal direction at each vertex and is stored as a height-field, which is a 2.5D shape. To paste the detail surface to a target model, the corresponding vertices of the target model are moved to the normal directions using the height values of the detail surface.

However, it is difficult to apply these cut-and-paste operations to general features, which may be thick or have overhangs and handles, as shown in Figure 2, because a feature cannot be easily separated by the approximation of the original surface; a thick feature may cause self-intersections; and a feature that produces handles or holes cannot be mapped on a planer surface.

To paste features with overhangs, Kanai et al. [11] proposed a mesh fusion technique based on the morphing of mesh models. This approach is very interesting, but it may cause unnatural distortions because of the parameterization based on harmonic mapping. Forsey and Bartels [8] proposed hierarchical B-spline surfaces that can be used to paste features in parametric space, but their method restricts the pasting within surface patches.

Our method is based on a volume-pasting approach, which pastes a parametric volume instead of a height-field. Since we paste a volumetric region on the target model, we can deform and paste a feature in the volume even if the feature has overhangs or handles. In addition, our optimization algorithm for volume pasting avoids self-intersections of thick features, as shown in Figure 2.

A parametric volume maps coordinates from parametric space to Euclidean space using a function $\mathbf{V}:(\mathrm{u},\mathrm{v},\mathrm{w})\rightarrow(\mathrm{x},\mathrm{y},\mathrm{z})$. A typical mapping $\mathbf{V}(u,v,w)$ is defined by the tensor product of three B-spline curves [17]. A regular three-dimensional lattice of control points $\{\mathbf{P}_{ijk}\}$, each of which has a $\{x, y, z\}$ coordinate, determines such a parametric volume $\mathbf{V}(u, v, w)$.

To realize a volume-based cut-and-paste technique, we propose the following methods:
1. To calculate the base and the detail surfaces, we divide the user-specified region into a pure feature region and its surrounding region. The base surface is calculated using only the surrounding region. We introduce a local mesh segmentation method to extract pure feature regions automatically.
2. Parametric volumes that fit to the source or target models need to be calculated. We solve this problem using B-spline volume fitting. In this fitting method, the parameterization, which assigns a (u, v) parameter to each vertex in a mesh, is gradually updated to improve the fitting errors. In the objective function for volume fitting, the weights of the fitting error metrics and the smoothness metrics are controlled to avoid self-intersections.
3. On the basis of our cut-and-paste method, we introduce modeling operations that allow feature registration, feature removal, feature pasting, and texture pasting.

The rest of the paper proceeds as follows. The next section provides the overview of our method. In Section 3, we describe how to separate a region into the base surface and the detail surface. In Section 4, our volume fitting technique is explained, and in Section 5, modeling operations based on volume fitting are described. In Section 6, we demonstrate our modeling operations, and finally, Section 7 concludes the paper.

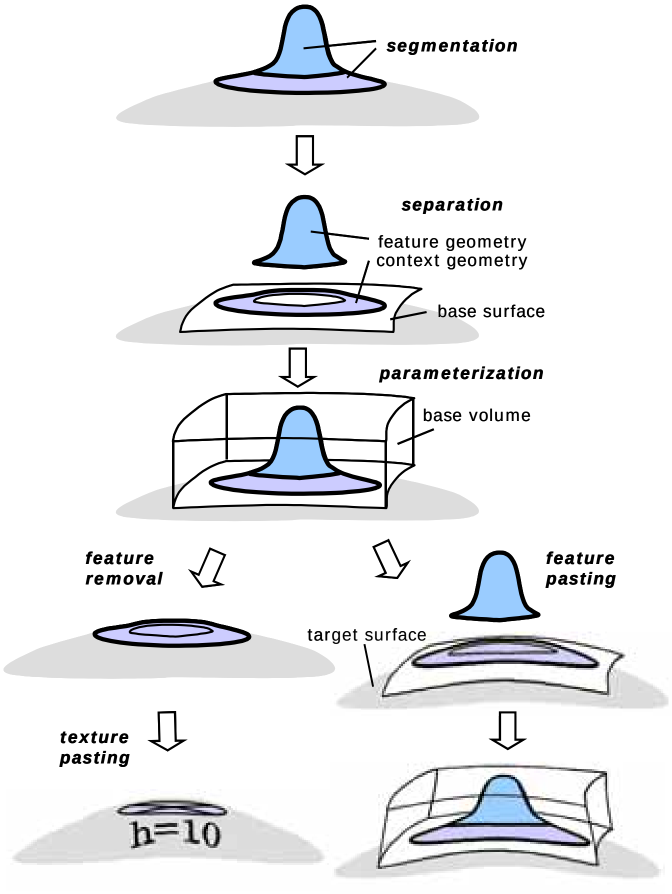


Figure 3. Modeling process.

2. OVERVIEW

Our objective is to develop modeling operations suitable for the reuse of existing 3D models to support communication in the early stage of design. We suppose mesh models generated by scanning real products or tessellating CAD models.

On the basis of our volume fitting method, we introduce the following operations:
Feature registration: Features are extracted from mesh models and stored in a feature library. Coordinates of a feature are normalized in a parametric volume for reuse.
Feature removal: Unnecessary features are removed, and replaced by their base surfaces.
Feature pasting: An applicable feature is adapted to and pasted on the target model.
Texture pasting: If users cannot find appropriate features, they can interactively sketch on the target model.

Figure 3 shows the proposed modeling process:
(a) First, the user selects a region including a feature and its surrounding area. Then, the system calculates the border of these two regions. Here, we call the pure feature area a *feature region*, and the surrounding area a *context region*.
(b) The region is separated into two regions at the border. Then, the base surface is calculated by approximating the context region.
(c) The initial parametric volume is defined so that it involves the entire feature region. Then, the control points of the volume are optimized so that the bottom surface corresponds to the base surface and the feature region is not largely modified. We call this parametric volume the *base volume*.
(d) The feature region is parameterized in the base volume.
(e) To remove a feature, the feature region is replaced by the base surface.
(e) To paste a feature, the base volume of a feature is deformed to fit to the target model. Then, the feature region is deformed**.**
(f) When the user cannot find suitable features in a feature library, he/she can draw strokes on the target model after unnecessary features are removed. The drawings are represented in images and pasted on the model using texture mapping.

3. SEPARATION INTO FEATURE AND CONTEXT

To calculate a base surface for a cut-and-paste operation, we have to separate the user-specified region into a feature region and a context region.

For the segmentation of regions, several methods have been reported [18-22]. They are roughly categorized into local search methods [18,19] and global optimization methods [20-22].

In local search methods, a region is typically separated into several small regions by investigating the tangency or the curvature of adjacent faces. However, these methods may not separate a region into exactly two regions and may not manage nearly flat areas correctly. In addition, local searches are not robust for noisy data, such as scanned models.

Therefore, we will use a global optimization method. Although several algorithms have been reported so far, we use the formalism proposed by Kats et al. [22], which reduces the segmentation problem to a well-known maximum flow, minimum cut problem [23]. This method is stable and divides a region into exactly two regions. In our experiments, it produces correct and non-jaggy boundaries for most 3D models. However, this method is very time-consuming for large models. Therefore, we use the method only for the interactive selection of features.

Figure 4 shows a segmentation process in our system. First, the user selects a region that includes a feature region and a context region, as shown in Figure 4(b). The region must be selected so that the mesh model is separated into exactly two regions by eliminating faces in the region; in other words, the region includes cut-set edges of the mesh. Then, the optimal cut-set is calculated and the feature region is correctly separated from the user-specified region, as shown in Figure 4(c). Since the selected region contains the relatively small number of faces, it can be segmented in a very short calculation time.

The segmentation algorithm consists of the following steps:
(1) In Figure 5, the region roughly specified by the user is shown as medium region **C**, and faces in the region are shown in dashed lines.
(2) The mesh in the selected region is converted to its dual graph, which is generated by mapping every face in the mesh to a node, and by connecting two dual nodes by an edge if the corresponding faces are adjacent, as shown in solid lines in Figure 5.
(3) Source **S** and sink **T** are located in each of two regions **A** and **B**. Then, **S** and **T** are connected to nodes of the dual graph.
(4) A capacity is assigned to each dual edge so that the capacity becomes smaller when the angle of two adjacent faces is larger. We use the capacity function proposed by Kats et al. [22], as follows. Given two adjacent faces, f_i and f_j, which correspond to nodes v_i and v_j in a dual graph, the angle between their normals is referred to as $\alpha_{i,j}$. Then, the capacity *C(i,j)* of a dual edge between nodes v_i and v_j is defined as:

$$d(i,j) = \eta(1-\cos\alpha_{i,j}),$$

$$C(i,j) = \begin{cases} \dfrac{1}{1+\dfrac{d(i,j)}{average(d(R))}} & (\mathrm{i, j} \neq \mathrm{s, t}) \\ \infty & (\mathrm{i, j} = \mathrm{s, t}) \end{cases} \qquad (1),$$

where η is a constant value. The appropriate value of η varies depending on whether a feature is convex or concave. We used $\eta = 0.1$ for convex features, and $\eta = 1$ for concave ones. One of these values is selected after a convexity check is applied to a feature.
(5) The minimum cut of this undirected flow network graph is calculated. We solve this maximum flow, minimum cut problem using the Ford-Fulkerson algorithm [23]. Since this minimum cut tends to flow through edges with small capacities, this method correctly detects feature boundaries even when flat faces exist on the boundary.

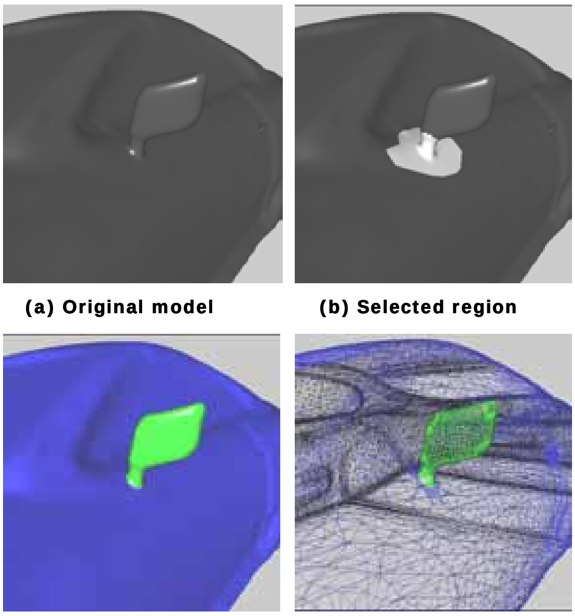

(a) Original model
(b) Selected region
(c) Extracted feature geometry.

Figure 4. Segmentation of a selected region.

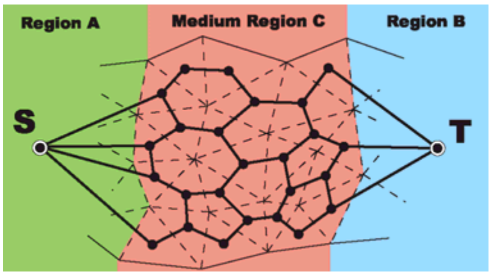


Figure 5. Flow network of a dual graph.

4. VOLUME FITTING

The user-specified region is divided into a feature region and a context region by the segmentation algorithm described in the previous section. Then, we calculate the base surface using the context region, and the base volume using the feature region.

4.1. Notations

Let **S** denote a B-spline surface and **V** a B-spline volume. They are defined as follows:

$$\mathbf{S}(u,v) = \sum_{i=0}^{n-1}\sum_{j=0}^{m-1} N_i^p(u) N_j^q(v) \mathbf{P}_{ij} \qquad (2)$$

$$\mathbf{V}(u,v,w) = \sum_{i=0}^{n-1}\sum_{j=0}^{m-1}\sum_{k=0}^{l-1} N_i^p(u) N_j^q(v) N_k^r(w) \mathbf{P}_{ijk} \qquad (3)$$

where $\mathbf{P_{ij}}$ and $\mathbf{P_{ijk}}$ are control points; N_i^p is a B-spline basis function; *p*, *q*, and *r* are orders; *l*, *m*, and *n* are the numbers of control points.

4.2. Surface Fitting

We denote points of vertices in the context region by $\{\mathbf{Q_t}|\ t = 0, \ldots, \text{s-1}\}$, and their parameters in the fitted surface by $\{(u_t, v_t)\}$. Then, we define the fitting error metrics as:

$$F_p^S = \sum_{t=0}^{s-1} \left|\mathbf{Q}_t - \mathbf{S}(u_t, v_t)\right|^2 \qquad (4)$$

We also introduce the smoothness metrics:

$$F_s^S = \iint \left(|\mathbf{S}_{uu}|^2 + 2|\mathbf{S}_{uv}|^2 + |\mathbf{S}_{vv}|^2 \right) dudv\,. \qquad (5)$$

These metrics are important for smoothly extrapolating holes in the context region as well as penalizing the distortion and the folding of faces.

Then, the surface-fitting problem is formalized as the optimization of:

$$F_p^S(\mathbf{P}^{(S)}) + \beta\, F_s^S(\mathbf{P}^{(S)}) \rightarrow \min \qquad (6)$$

where $\mathbf{P}^{(S)}$ is control points of the base surface, and β is a non-negative weight.

To calculate control points of surface $\mathbf{S}(u,v)$ by optimizing Eq. (6), we have to assign a (u, v) parameter to each point in $\{\mathbf{Q_t}\}$. In our implementation, the initial parameter settings are roughly determined by projecting points $\{\mathbf{Q}_t\}$ on a plane that minimizes the sum of the least square distance errors. Then, control points that minimize Eq. (6) are calculated by solving a sparse linear system. When the calculated surface causes self-intersections or the folding of faces, the value of β is changed to be large so that the distortion is more largely penalized than the fitting error.

In cut-and-paste editing, the parameters must be assigned very carefully, because the poor parameterization causes distorted unnatural feature pasting [13,24-26]. To improve the initial settings of parameters, we use a similar approach to the one used by Weiss et al. [27], which was originally developed for reverse engineering. In their method, (u, v) parameters are gradually refined by calculating control points and parameterizing $\{\mathbf{Q_t}\}$ alternately. Suppose parameter (u_t, v_t) is assigned to $\mathbf{Q}_t$, and it is refined to $(\hat{u}_t, \hat{v}_t)$, which is defined as the parameter of the nearest point to $\mathbf{Q}_t$ in the fitted B-spline surface. The parameter $(\hat{u}_t, \hat{v}_t)$ can be calculated by solving the following equations [28]:

$$\begin{cases} \mathbf{S}_u{}^T \cdot (\mathbf{S}(\hat{u}_t, \hat{v}_t) - \mathbf{Q}_t) = 0 \\ \mathbf{S}_v{}^T \cdot (\mathbf{S}(\hat{u}_t, \hat{v}_t) - \mathbf{Q}_t) = 0 \end{cases}, \qquad (7)$$

where $\mathbf{S}_u$ and $\mathbf{S}_v$ are partial differentials of $\mathbf{S}$.

When the surface has large fitting-errors even after parameters are refined, the system automatically inserts knots to increase the degrees of freedom for fitting. See [27] for the detail knot-insertion algorithm.

The above steps, which consist of the optimization, parameterization, and knot-insertion, are iterated until the fitting error satisfies a predefined threshold value. In our experience, this method successfully generates natural and smooth parameterization by a few iterations.

4.3. Volume Fitting

We extend the surface fitting method to B-spline volume fitting. In volume fitting, parameters and control points are refined alternately in the same way as surface fitting.

Let $\{\mathbf{Q}_f \,|\, f = 0, \cdots, r-1\}$ be the coordinates of vertices in a feature region. We generate a rectangular bounding box that involves these points and roughly assign their parameters (u_t, v_t, w_t) in the bounding box.

Then, we define the fitting error metrics $F_d{}^V$ and the smoothness metrics $F_s{}^V$, as follows:

$$F_p^V = \sum_{t=0}^{v-1} \left|\mathbf{Q}_t - \mathbf{S}(u_t, v_t, w_t)\right|^2 \qquad (8)$$

$$\begin{aligned} F_s^V = \iiint \Big(& |\mathbf{V}_{uu}|^2 + |\mathbf{V}_{vv}|^2 + |\mathbf{V}_{ww}|^2 \\ & + 2|\mathbf{V}_{uv}|^2 + 2|\mathbf{V}_{vw}|^2 + 2|\mathbf{V}_{wu}|^2 \Big) dudvdw \end{aligned}\,. \qquad (9)$$

Since the bottom surface $\mathbf{V}(u,v,0)$ corresponds with the base surface $\mathbf{S}(u,v)$, control points of the base volume can be calculated by minimizing the following objective function:

$$F_p^V(\mathbf{P}^{(V)}) + \beta^V F_s^V(\mathbf{P}^{(V)}) \rightarrow \min \qquad (10)$$

$$\text{subject to} \quad \mathbf{P}_{ij0}^{(V)} = \mathbf{P}_{ij}^{(S)}$$

where $\mathbf{P}^{(V)}$ is control points of the base volume.

The parameter (u_t, v_t, w_t) of point $\mathbf{Q}_t$ is refined by solving the following equation:

$$\begin{cases} \mathbf{V}_u{}^T \cdot (\mathbf{V}(\hat{u}_t, \hat{v}_t, \hat{w}_t) - \mathbf{Q}_t) = 0 \\ \mathbf{V}_v{}^T \cdot (\mathbf{V}(\hat{u}_t, \hat{v}_t, \hat{w}_t) - \mathbf{Q}_t) = 0 \\ \mathbf{V}_w{}^T \cdot (\mathbf{V}(\hat{u}_t, \hat{v}_t, \hat{w}_t) - \mathbf{Q}_t) = 0 \end{cases} \qquad (11)$$

Then, a B-spline volume is calculated again using Eq. (10) with updated parameters.

In a pasting operation, a volume is deformed on the target model using this volume fitting method and then the feature region inside the volume is also adaptively deformed.

5. MODELING OPERATIONS

5.1. Registration to Feature Library

After the base surface and volume are calculated using the fitting algorithm, the feature region is refined for registrations to a feature library.

First, the border of the feature region is recalculated so that it approximately rides on the base surface. It is refined by remeshing triangles near the border and investigating differences between the mesh and the base surface. If a vertex is connected to the vertices on the border and its difference is large, the vertex is added to the feature region.

Second, coordinates in a feature region are normalized as (u,v,w) $(0 \le u, v, w \le 1)$ in parameter space of the parametric volume. The normalized mesh models of features are stored in a feature library. In the current implementation, the feature data are stored in the STL format or the VRML format.

The size of the bounding box of a feature is also stored. It is used to determine the initial default size of a volume to be pasted on a target model. The height of a volume, which is not specified by the user in a pasting operation, is set to have a similar ratio to the one of the stored size.

5.2. Feature removal

A feature removal operation removes a feature region from a mesh model and fills the region with a smooth surface. Figure 6 shows a process of feature removal:

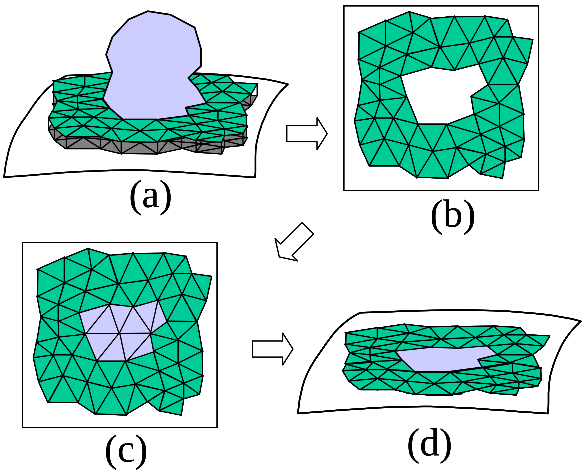


Figure 6. Feature removal.

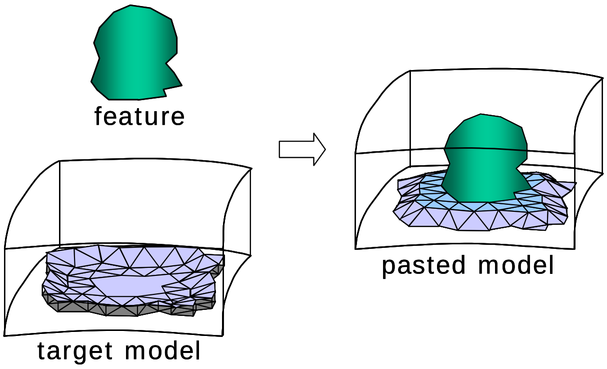


Figure 7. Feature pasting.

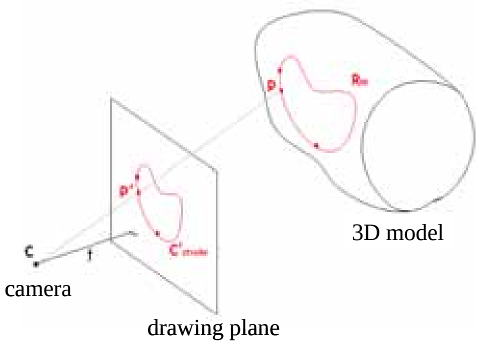


Figure 8. Drawing on a 3D model.

(a) The user roughly specifies a region by drawing strokes on a 3D model and the system calculates the border of the feature region using the segmentation method described in Section 3.

(b) Faces in the feature region are removed from the mesh model. Then, the system applies the B-spline surface fitting to the remaining vertices, as described in Section 4.2. Since a UV parameter of each vertex is also calculated in this fitting process, the border of holes can be mapped in UV parameter space.

(c) The holes are tessellated in UV parameter space and filled by triangles.

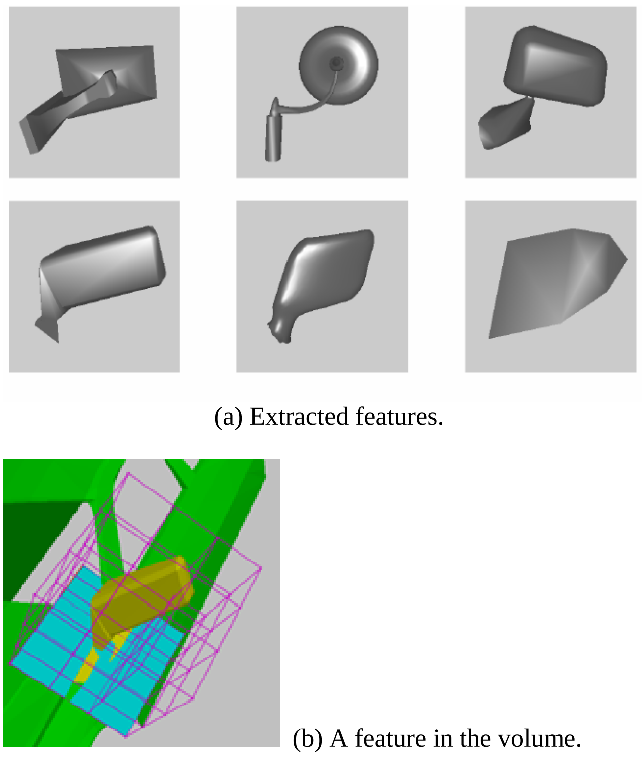
(d) Coordinates in UV space are mapped in 3D Euclidean space. Finally, the feature region is replaced by the smooth surface that extrapolates the context region.

5.3. Feature Pasting

In feature pasting, the user specifies a region by drawing a rectangular region on a target model. Strokes specified on a screen are projected onto a 3D model, as shown in Figure 8, and the specified region is regarded as the boundary of the base surface. Then, the base surface and volume are fitted to this region using the surface and volume fitting methods described in Section 4. The default size of the volume is determined by values stored in the feature library.

Vertices $\{(u_i,v_i,w_i)\}$ in the feature are embedded in parameter space of the base volume, and the shape of the feature is deformed. In this phase, the user can edit the feature by changing the height or each control point of the B-spline volume, or by adding new constraints to the parametric volume.

Then, the system remeshes the target model so that the border of the feature region is included in the mesh. Then, the



(a) Extracted features.

(b) A feature in the volume.

Figure 9. Extracted features.

system removes faces on which the feature is pasted, as shown in Figure 7. Finally, the feature is connected to the target model.

5.4. Texture pasting

In new product development, the designer may not be able to find appropriate features in the feature library. To support such cases, we define operations that paste 2D drawings on 3D models.

This operation is mainly used in combination with feature removal operations. The user first removes features on a 3D model to create a drawing region on the model. Then, the user draws figures on a drawing plane, as shown in Figure 8. In texture pasting, strokes on the drawing plane are stored as a texture image, and are mapped on a 3D model. Each vertex in the 3D model is related to a texture coordinate (i, j), which represents a position in the image. For real-time mapping, we only calculate texture coordinates near strokes drawn on the 3D model.

This operation is very simple and cannot define a correct shape, but it will be complementarily used with feature pasting. The careful use of texture pasting can support the communication in the initial design phase.

6. RESULTS

We implemented our operations and applied to several models. We used a PC with a Pentium 4.15GHz CPU and 1.0GB memory.

Figure 9 shows extracted fender mirrors to store in a feature library. Their feature regions were interactively specified by strokes. The calculation time for each segmentation process was less than 0.02sec.

Calculation time of volume-fitting using various numbers of control points is shown in Figure 10. In our experiments, volumes with less than 500 control points can be calculated interactively. The number of control points is determined by the fitting error, which can be controlled by the user in our implementation. We observe that this calculation time is

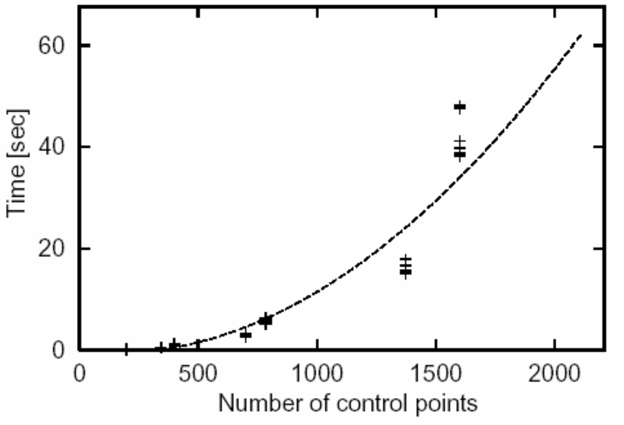


Figure 10. Elapsed time of volume-fitting.

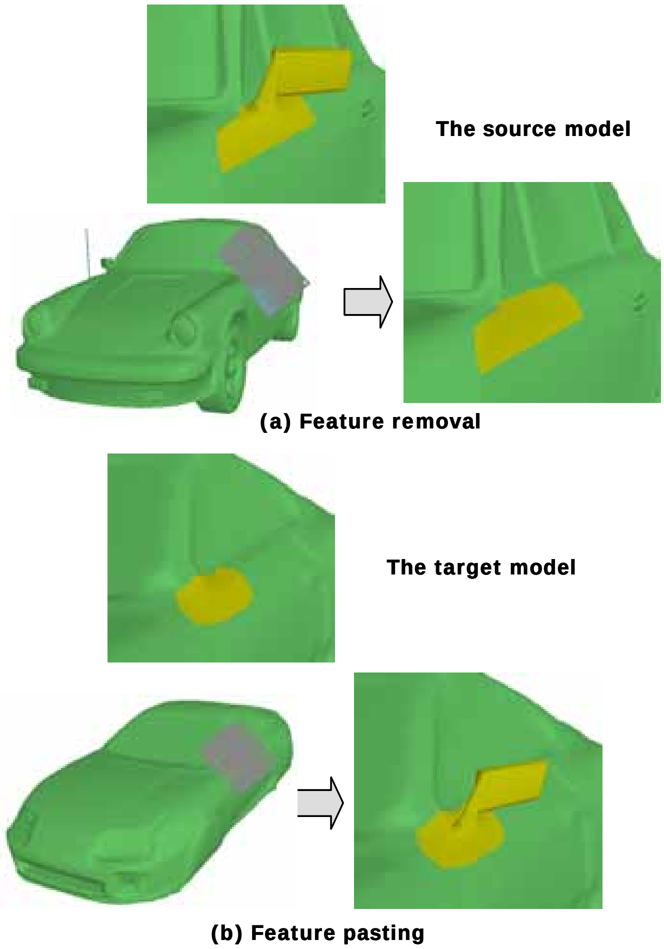


Figure 11. Examples of feature removal and pasting.

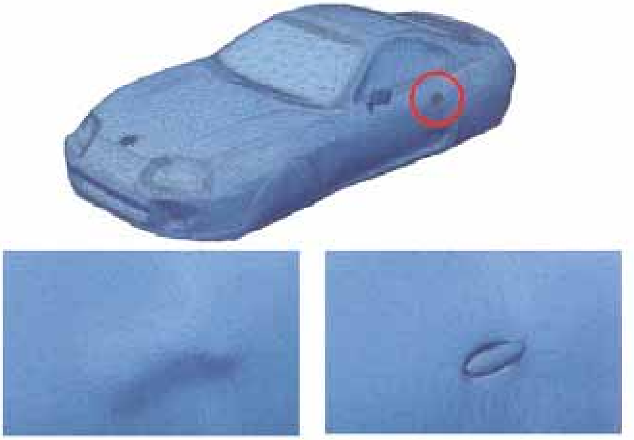


Figure 12. A combination of feature removal and pasting.

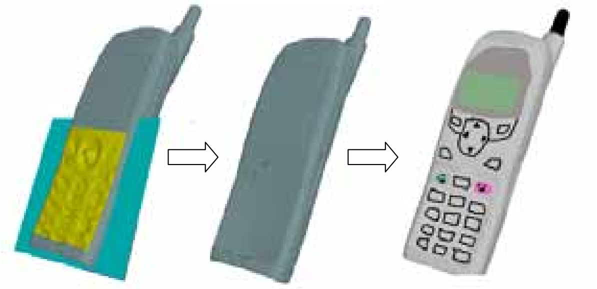


Figure 13. An example of texture pasting.

sufficient for our modeling operations, because precise surface fitting, such as a 10^{-5} order, is not required in feature pasting. In addition, volumes need a small number of control points in the height direction.

Figure 11 shows feature removal and pasting operations. In the feature removal, the specified feature could be smoothly replaced by the base surface. In feature pasting, the fender mirror in Figure 11(a) could be also smoothly pasted on another model in Figure 11(b).

Figure 12 shows a combination of feature removal and pasting operations. Since both operations use the common volume fitting process, they can be applied simultaneously.

Figure 13 shows an example of texture pasting. In the left figure, the user specifies unnecessary features in the part of buttons, and the system generates the base surface that approximates the specified region. After the feature removal in the middle figure, the user can draw a new concept on the region.

7. CONCLUSIONS AND FUTURE WORK

Our research aims to develop a modeling environment that supports the early phases of product development. In the early stage of design, rough models are useful for representing, evaluating, and narrowing down a number of design concepts. We discussed modeling operations suitable for creating rough 3D models by modifying existing 3D models and models scanned from real products. We do not intend to create precise detail 3D models by our modeling operations, because such models should be created using existing commercial CAD systems after concepts are narrowed down to a few candidates.

In this paper, we proposed a volume-based cut-and-paste method. Our method allows pasting shapes that may be thick or have overhangs and handles. A B-spline volume is fitted to a region in a 3D model by the optimization of the fitting error metrics and the smoothness metrics. This method enables users to smoothly paste a wide variety of features and to efficiently avoid self-intersections. In addition, mesh segmentations based on a maximum-flow minimum-cut algorithm can correctly separate feature regions and context regions. These regions are used to calculate the base surfaces and base volumes for cut-and-paste editing. Our volume-based approach can successfully enhance the capability of cut-and-paste methods.

On the basis of the method, we developed modeling operations, which involve feature registration, feature removal, feature pasting, and texture pasting. These operations were implemented and evaluated using sample models. The result shows they can effectively generate new models by registering, removing, and reusing partial shapes in existing models.

In future work, we need to develop easy-to-use interface for 3D modeling, because our modeling operations would be used in an interactive environment. We need to study 3D sketches and other interactive modeling techniques. In addition, for practical use, our modeling operations will be used in combination with deformable modeling, which allows locally deforming shapes. Our current system is just a prototype, but we would like to add more modeling operations for realizing a practical system for the early design phases.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the support provided by Satoshi Kaneda of the University of Tokyo.

REFERENCES

1. Ulrich, K. T. and Eppinger, S. D. (1995). “Product design and development,” McGraw-Hill, New York, NY.
2. http://www.catia.com
3. http://www.unigraphics.com
4. Sederberg, T. and Parry, S. R. (1986). “Free-form deformation of solid geometric models”, *Proceedings of SIGGRAPH 86*, pp. 151-160.
5. Coquillart, S. (1990). “Extended free-form deformation: a sculpturing tool for 3D geometric modeling,” *Proceedings of SIGGRAPH 90*, pp. 187-196.
6. MacCracken, R. and Joy, K. I. (1996). “Free-Form Deformation with Lattices of Arbitrary Topology”, *Proceedings of SIGGRAPH 96*, pp.181-188
7. Eck, M. DeRose, T., Duchamp, T., Hoppe, H., Lounsbery, M. and Stuetzle, W. (1995). “Multiresolution analysis of arbitrary meshes,” *Proceedings of SIGGRAPH 95*, pp. 173–182.
8. Forsey, D. R. and Bartels, R. H. (1995), “Surface fitting with hierarchical splines,” *ACM Transactions on Graphics*, 14(2), pp.134–161.
9. Barghiel, C., Bartels, R., and Forsey, D. (1994). “Pasting spline surfaces,” Mathematical Methods for Curves and Surfaces, edited by M. Daehlen, T. Lyche, L. L. Schumaker, Vanderbilt University Press, Nashville, TN., pp. 31-40.
10. Chan, L.K.Y., Mann, S., and Barrels, R. (1997). “World space surface pasting,” *Proceedings of Graphics Interface*, pp. 146-154.
11. Kanai, T., Suzuki, H., Mitani, J., and Kimura, F (1999). “Interactive mesh fusion based on local 3D metamorphosis,” *Proceedings of Graphics Interface*, pp. 148-156.
12. Suzuki, H., Sakurai, Y., Kanai, T., and Kimura, F (2000). “Interactive mesh dragging with an adaptive remeshing technique,” *The Visual Computer*, 16(3-4), pp. 159-176.

13. Biermann, H., Martin, I., Bernardini, F., and Zorin, D. (2002).“Cut-and-paste editing of multiresolution surfaces,” *ACM Transaction on Graphics*, 21(3), pp. 312-321.
14. Hanrahan, P., Haeberli, P. (1990). “Direct WYSIWYG painting and texturing on 3D shapes”, *Proceedings of SIGGRAPH 90*, pp.215-223.
15. Igarashi, T. and Cosgrove, D. (2001). “Adaptive Unwrapping for Interactive Texture Painting,” *Proceedings of the 2001 Symposium on Interactive 3D graphics*, pp. 209 - 216.
16. DeBry, D., Gibbs, J., Petty, D. D., Robins, N. (2002). “Painting and rendering textures on unparameterized models”, *ACM Transactions on Graphics*, 21(3), pp. 763-768.
17. Farin, G. (1993), Curves and surfaces for computer aided geometric design (3rd edition), Academic Press, San Diego, CA.
    4th ed., Academic Press, Boston, 1997
18. Mangan, A. P. and Whitaker, R. T. (1999). “Partitioning 3D surface meshes using watershed segmentation,” *IEEE Transactions on Visualization and Computer Graphics*, 5(4), pp. 308-321.
19. Chazelle, B., Dobkin, D. P., Shouraboura, N., and Tal, A. (1995). “Strategies for polyhedral surface decomposition: An experimental study,” *Proceedings of Symposium on Computational Geometry*, pp. 297-305.
20. Li, X., Toon, T. W., and Huang, Z. (2001). “Decomposing polygon meshes for interactive applications,” *Proceedings of the 2001 Symposium on Interactive 3D Graphics*, pp. 35-42.
21. Shalafman, S., Tal, A., and Katz, S. (2002). “Metamorphosis of polyhedral surfaces using decomposition,” *Proceedings of Eurographics*, pp. 219–228.
22. Katz, S. and Tal, A (2003). “Hierarchical mesh decomposition using fuzzy clustering and cuts,” *ACM Transactions on Graphics*, 22(3), pp. 954-961.
23. Fulkerson, D. R., and Ford, L. R. (1962). “Flows in Networks,**”** Princeton Univ. Press.
24. Floater, M. S. (1997). “Parametrization and smooth approximation of surface triangulations,” *Computer Aided Geometric Design*, 14(3), pp. 231-250.
25. Krishnamurthy, V., and Levoy, M. (1996). ”Fitting smooth surfaces to dense polygon meshes,” *ACM SIGGRAPH 96*, pp. 313-324.
26. Sheffer, A., de Sturler, E. (2000). “Surface parameterization for meshing by triangulation flattening,” *Proceedings of 9th International Meshing Roundtable*, pp.161-172.
27. Weiss, V., Andor, L., Renner, G., Varady, T. (2002). “Advanced surface fitting techniques,” *Computer Aided Geometric Design*, 19(1), pp. 19-42.
28. Piegl, L. A. and Tiller, W. (1998) “The Nurbs Book,” Springer Verlag, New York.