

# FEA-Mesh Editing with Feature Constrained

Chuhua Xian, Shuming Gao, Hongwei Lin\*, Yusheng Liu and Dong Xiao  
State Key Lab. of CAD & CG  
Zhejiang University  
310058, Hangzhou, P. R. China  
{xianchuhua, smgao, hwlin, yslu, xiaodong}@cad.zju.edu.cn

## Abstract

*Mesh editing can provide various models for FEA-simulation in industry. This paper proposes a framework for FEA-mesh editing with feature constrained. In the framework, cage-based technique is first used to edit the base-decomposition model. Vertices of the constrained feature are transformed into a local form. Parameters are analyzed before editing operation, and our method permits the user to add constraints on the parameters of the feature. This framework can also keep consistence for the disconnected assembly mesh model. Experimental results show that constrained features are kept precisely after mesh editing. Additionally, experimental data indicate our method is efficient and achieves real-time response.*

## 1. Introduction

In mechanical engineering, products are often evaluated by CAE analysis to check whether they satisfy engineering requirements. Usually, a rough 3D model is first created by CAD software, and then FEA tools are employed for its structure analysis or dynamic analysis. If the analyzing results do not satisfy the engineering requirements, the CAD model has to be modified using CAD software and analyzed again with FEA tools. Thus, the CAD model modification and CAE analysis procedure is repeated till a desirable product is generated. In fact, prior to CAE analysis, the CAD model is required to be *re-meshed* for finite element analysis (CAE analysis). The re-meshing procedure is very time-consuming, and the mesh quality is critical to the CAE analysis. Therefore, in order to avoid totally re-meshing the CAD model in each *modification-and-analysis* repetition, we develop a method to edit the FEA mesh directly while keeping some features [1] and constraints.

## 2. Related Work

In recent decades, a wide variety of mesh-editing techniques have been developed. These techniques can be classified into two categories : cage-based and surface-based.

Cage-based techniques deform shapes by modifying the space in which objects lie [2]. Chen [3] uses free-form deformation(FFD) to generate a series paradigm of FEA-meshes based on a basic design for CAE-based simulation. Inheriting the idea of FFD, cage-based methods construct a cage to envelop a mesh, and the mesh vertices are represented by the affine sums of the cage's vertices and its face normals [4], [5]. Users manipulate the cage to induce a smooth space deformation. The advantages of cage-based techniques are simplicity, flexibility and efficiency. They can also process disconnected models. But it is difficult to use these approaches to keep the specified feature regions without attaching any constraints on the model.

Based on the theory of partial differential equation(PDE), Yu et al. [6] manipulate the gradient field of the coordinate functions of the mesh, and solve the Poisson equation to get the positions of the vertices. Sorkine [7] introduces a similar technique which uses the discrete Laplacian-Beltrami operator to define differential properties at vertices and add positional constraints to form a linear system. In order to make use of surface-based techniques for deforming automobile sheet-metal panels, Masudaa et al. [8] propose the soft and hard constraints on the mesh and develop a framework which can preserve the form features of the sheet-metal panel while deforming the model. In [9], Masudaa defines virtual links between pairs of disconnected vertices, and uses his former work to fix this problem. Since this method needs to select the pairs of vertices from disconnected parts by manual, it is difficult when the assembly model is very complex. On the other hand, this method aims at sheet metal deformation. For deforming other types of models in engineering, it only scales the feature globally and does not permit to modify part of the feature parameters. In FEA models, there may be millions of vertices. Therefore, using the surface-based techniques will lead to solving a very large sparse system, which is a time-consuming task even applying current sparse.

## 3. Framework of Constrained Editing

The full FEA-mesh editing framework proposed in this paper provides four basic operators shown in Figure 1 :

the decomposition operator(*feature separation*), the editing operator(*shape deformation*), the feature constraining operator(*feature analysis*), and the reconstruction operator(*model synthesis*).

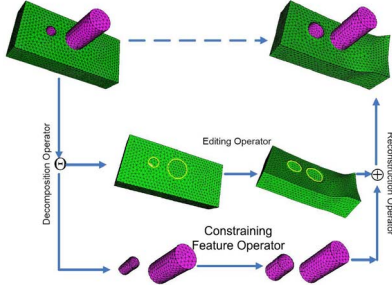


Figure 1. A FEA-mesh editing framework with feature constrained

When the users specify features and add constraints, decomposition operator is performed. It can be given by,

$$B_i = S_i \ominus F_i, \quad (1)$$

and

$$D = \bigcup_i B_i \oplus R, \quad (2)$$

where  $B_i$  is the surface that has subtracted the constrained feature  $F_i$ ,  $S_i$  is the surface with specified feature,  $R$  is the remainder part of the FEA-mesh,  $\ominus$  is a minus operator,  $D$  is the base decomposition model(**BDM**), and  $\oplus$  is a summation operator. The reconstruction is done by combining the modified  $B_i \rightarrow B'_i$ , the constrained features  $F_i \rightarrow F'_i$ , and  $R \rightarrow R'$ , that is,

$$M = \bigoplus_i (B'_i \oplus F'_i) \oplus R' \quad (3)$$

#### 4. Base-decomposition Model Editing

As presented in Eqs. (1) and (2), the base-decomposition model is the model getting rid of the constrained features. Since the cage-based techniques are able to handle the disconnected assembly model and perform very efficiently for large mesh with lots of vertices, in this paper, a recent developed cage-based technique, Green Coordinates(GC) [5], is employed to deform the models. Given a BDM  $D$ , we first construct a cage  $C$  that envelops the edited part, and represent each vertex on the part as a weighted combination of cage vertices and face normals. As the cage changes,  $D$  is deformed in turn by applying the GC weights to the deformed cage.

A good cage is important to the cage-based mesh editing, which should loosely encloses the mesh. The cage generation algorithm mainly includes the following steps:

- 1) Compute the bounding box and voxelize the mesh model;
  - 2) Extract and triangulate the outer faces of the feature voxels;
  - 3) Smooth the cage by an improved mean curvature flow method, and decimate the cage.
  - 4) If the cage is not suitable for editing, goto step 3.
- More details about this method can be found in [10].

#### 5. Constraints on Features

Feature is a partial part that has special meaning in engineering such as holes or protrusions. For a FEA-mesh model, feature is a set of discrete vertices connected by some relations on the specified region.

##### 5.1. Local Coordinates Transformation

Suppose a constraining feature  $F(V, T)$  is a subset of the FEA-mesh, where  $V$  is vertices of  $F$  and  $T$  is the face set of  $F$ . We denote the index set of  $F$  as  $\Gamma_F$ , and the intersected boundary as  $\Omega_F$ . Then a local transformation is defined on the vertex  $\eta_i (i \in \Gamma_F - \Omega_F)$  of the feature as

$$\Lambda(\eta_i) = \frac{1}{d_{\Omega_F}} \sum_{j \in \Omega_F} (\eta_i - v_j), \quad (4)$$

where  $d_{\Omega_F} = |\Omega_F|$  is the number of vertices of the intersected boundary. Intuitively, the local transformation encapsulates a local translation of the barycenter of the intersected boundary and the vertex  $v_i$  (see *left* in Figure 2)

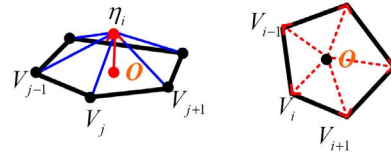


Figure 2. (*left*). Local coordinates translation from boundary vertices. (*right*). Intersected boundary constraints

To preserve the form of the feature precisely, we add the constraint

$$\Lambda(\eta'_i) = f(R_F \Lambda(\eta_i)), \quad (5)$$

where  $R_F$  is the rotation matrix which will be introduced in the following, and  $f(\bullet)$  is a constraining function about parameters.

The transformation of the vector of absolute Cartesian coordinates to the vector of  $\Lambda(\eta)$  in Eq. (4) can be regarded as a local coordinate defined by the boundary.

In order to reconstruct the feature after performing editing operator, we need to determine  $\Omega_F$ . Let  $O = \frac{1}{d_{\Omega_F}} \sum_{i \in \Omega_F} v_i$

is the barycenter of the intersected boundary, which is regarded as a virtual vertex of **DBM** and treated in the same manner as other vertices. For each vertex  $v_i (i \in \Omega_F)$ , a scalar function  $\Xi$  is defined as (right in Figure 2),

$$\Xi(v_i) = \|v_i - O\|. \quad (6)$$

We have therefore additional constraints of the form:

$$v_i'' - O' = f(\Xi(v_i)(v_i' - O')), i \in \Omega_F, \quad (7)$$

where  $v_i'$  and  $O'$  denotes the vertices after editing, and  $f(\bullet)$  is the same as in Eq. (5). After performing reconstruction operator, Eq.(7) will be satisfied, and feature  $F$  can be reconstructed from Eqs.(4) and (5).

## 5.2. Rotation Constraint

When we edit the mesh, the constrained features need to be rotated to fit the normal of the base surface. After the boundary has been reconstructed, rotation matrix in Eq. (5) can be determined from the normal of the intersected boundary. As illustrated in Figure ??,  $n$ ,  $n'$  and  $O$  are the original normal, new normal and the barycenter of the intersected boundary, respectively. The rotation matrix  $R_F$  in Eq. (5) is obtained by rotating  $n$  to  $n'$  around barycenter  $O$ . Figure 3 shows some examples deformed with rotation and without rotation. The constrained features in Figure 3(c) are more reasonable in practice.



Figure 3. Edited model. (a)Original model (b)Deformed model without feature rotation (c)Deformed model with feature rotation

## 5.3. Constraint on Parameters

Suppose a feature has  $m$  parameters, and  $\vec{g}_i(\vec{c}_i, \bullet)$  ( $i = 0, 1, \dots, m-1$ ) is a constraining function on parameter  $\vec{c}_i$ . Then  $f$  in Eq. (5) is rewritten as

$$f(\bullet) = \sum_{i=0}^{m-1} \vec{g}_i(\vec{c}_i, \bullet). \quad (8)$$

Different types of features have different sketches and parameters. In this section, we mainly analyze the parameters of through holes and protrusions with circular and rectangle shape. Other types can be analyzed in a similar way.

**Circular hole and protrusion:** In traditional design procedure, users first create a circular sketch on a reference plane, and then extrude it to get a cylindrical feature. Radius  $\vec{r}$  is defined in sketch plane, and height  $\vec{h}$  is defined

by extrusion, which are shown in Figure 4(a)(b). Then, constraint  $\vec{g}(\vec{r}, \bullet)$  on  $r$  and  $\vec{g}(\vec{h}, \bullet)$  on  $h$  are

$$\vec{g}(\vec{r}, \vec{h}) = s_r \vec{h} \sin \alpha, \quad (9)$$

and,

$$\vec{g}(\vec{h}, \vec{h}) = s_h \vec{h} \cos \beta, \quad (10)$$

where  $s_r$  and  $s_h$  are scalar factors,  $\alpha$  is the angle between the normal of sketch plane  $n$  and vector variable  $\vec{h}(R_F \Lambda(\eta_i))$  in Eq. (5) or  $\Xi v_i(v_i' - O')$  in Eq. (7)), and  $\beta$  is the angle between the extrusion direction  $\vec{e}$  and  $\vec{h}$  (see Figure 5).

**Rectangular hole and protrusion:** Figure 4(c) and (d) show three parameters, length  $l$ , width  $w$  and height  $h$ . We define  $\vec{l}$ ,  $\vec{w}$  as the directions of length and width, respectively. Similar with above, we have

$$\vec{g}(\vec{l}, \vec{h}) = s_l \vec{h} \cos \gamma, \quad (11)$$

and,

$$\vec{g}(\vec{w}, \vec{h}) = s_w \vec{h} \cos \vartheta, \quad (12)$$

where  $\gamma$  is the angle between  $\vec{l}$  and  $\vec{h}$  (Figure 5(c)), and  $\vartheta$  is the angle between  $\vec{w}$  and  $\vec{h}$ . Constraint on  $h$  can be treated using Eq. (10).

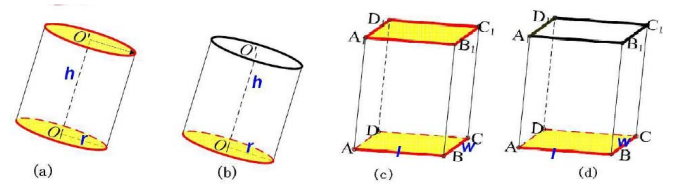


Figure 4. Parameters of circular and rectangular features.

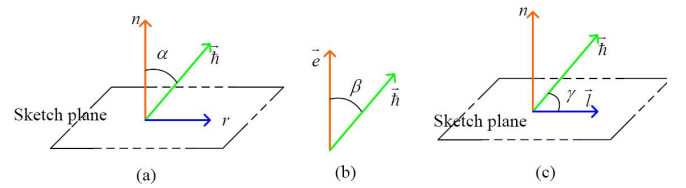


Figure 5. Constraints on parameters

## 6. Implementation and Results

The framework of FEA-mesh editing with feature constrained developed in this paper is implemented with VC++ 2005 and OpenGL, and runs on the PC with Core 4<sup>TM</sup> 2.4GHZ and 4GB RAM in a single thread. In our experiments, all mesh models are generated by FEA tools.

Figure 6b shows a deformed part without feature constraints. We can see that three circular features have transformed into elliptical shapes. However, with feature constraints in Figure 6c, circular features preserve the shape

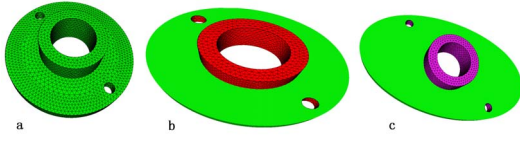


Figure 6. (a). Original model (b). Deformed model without feature constraints. (c). Deformed model with feature constraints.

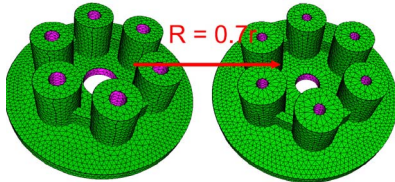


Figure 7. (a). Original model (b). Deformed model with scaling radius.

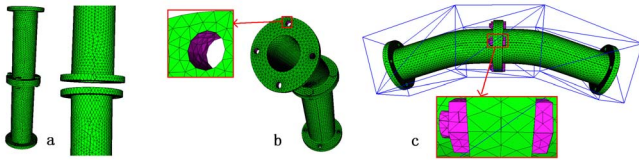


Figure 8. Assembly model

during deforming the model. Figure 7 shows parameter constraints on features. The radii of circular holes in Figure 7 shrink. The scalar factor  $s_r$  is 0.7.

The assembly model in Figure 8 consists of two pipes and four bolts. There is a gap between the two pipes. Figure 8 shows the deformed shape with four hole constraints. In Figure 8c, four bolts are constrained and assembly information keeps consistent after editing the model.

## 7. Conclusion

This paper focuses on developing a FEA-mesh framework with feature constrained. The whole framework consists of two major processes: editing the base-decomposition model and constraints on features. We use the cage-based technique to edit the base-decomposition model. Vertex of feature is transformed into a local form, which makes it only relate with the intersected boundary. Furthermore, rotation matrix of the feature can be obtained by the normal rotation of the intersected boundary. We analyze constraints on feature parameters. Our method can preserve the feature shape and permit user to add parameter constraints. Experiment results show that assembly mesh models can be kept consistent when performing editing operation. And experimental data indicate our method is efficient and achieves real-time responds.

## Acknowledgment

The authors would like to thank Zhu Hua for providing the model in Figure 8. This paper is supported by NSF of China (No. 60736019).

## References

- [1] M. Fontana, F. Giannini, and M. Meirana, "A free-form feature taxonomy," *Computer Graphics Forum*, vol. 18, no. 3, pp. 107–118, 1999.
- [2] W. S. Thomas and R. P. Scott, "Free-form deformation of solid geometric models," in *Proceedings of SIGGRAPH 1986*, 1986, pp. 151–160.
- [3] Y. Chen, P. Stewart, and A. Marsan, "A mesh feature paradigm for rapid generation of CAE-based design of experiments data," in *Proceedings of ASME DETC 2002*, Montreal, Canada, September 2002.
- [4] T. Ju, S. Schaefer, and J. Warren, "Mean value coordinates for closed triangular meshes," in *Proceedings of SIGGRAPH 2005*, 2005, pp. 561–566.
- [5] Y. Lipman, D. Levin, and D. Cohen-Or, "Green coordinates," *ACM Transaction on Graphics*, vol. 27, no. 3, August 2008.
- [6] Y. Yu, K. Zhou, X. Shi, H. Bao, B. Guo, and H. Shum, "Mesh editing with poisson-based gradient field manipulation," *ACM Transactions on Graphics*, vol. 23, no. 3, pp. 644–651, 2004.
- [7] O. Sorkine, D. Cohen-Or, Y. Lipman, C. Ross, and H. Seidel, "Laplacian surface editing," in *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, 2004, pp. 175–184.
- [8] H. Masudaa, Y. Yoshiokaa, and Y. Furukawab, "Preserving form features in interactive mesh deformation," *Computer-Aided Design*, vol. 39, pp. 361–368, 2007.
- [9] H. Masudaa and K. Ogawa, "Application of interactive deformation to assembled mesh models for CAE analysis," in *Proceedings of the ASME 2007*, Las Vegas, Nevada, USA, September 4–7 2007.
- [10] C. Xian, H. Lin, and S. Gao, "Automatic generation of coarse bounding cages from dense meshes," in *IEEE International Conference on Shape Modeling and Applications (SMI) 2009*, Beijing, P. R. China, June 2009, p. Accepted.