Computer-Aided Design and Applications

http://www.cadanda.com

Triangular Mesh Deformation based on Dimensions

Hiroaki Date[1] and Masahiko Onosato[2]

[1] Hokkaido University, hdate@ssi.ist.hokudai.ac.jp
[2] Hokkaido University, onosato@ssi.ist.hokudai.ac.jp

ABSTRACT

In this paper, we propose a triangular mesh deformation method based on dimensions. By specifying the type of dimensions, the regions defining the dimensions, and the target dimensions, the mesh is automatically deformed so that it satisfies the target dimensions. Mesh deformation is based on space-based deformation using barycentric coordinates. The deformation handle, which consists of a set of triangular prisms, is automatically created by mesh segmentation, simplification and offsetting of simplified mesh. In deformation, the deformation and fixed spaces are assigned to each prism in the handle, and the mesh is deformed by applying affine transformation to the control vertices and space-based deformation to the deformation space. Results for the mechanical parts show the effectiveness of our method.

Keywords: mesh deformation, dimensions, barycentric coordinates, CAD/CAE.
DOI: 10.3722/cadaps.2008.287-295

1. INTRODUCTION

Mesh-based geometric modeling is a powerful technology for use in CG character design and product design. Especially in CAE, flexible mesh deformation methods are essential for realizing efficient product development. In order to find a functionally satisfactory shape of the product, the current CAE process includes iterative meshing of the CAD model, analysis, and modification of the product shape using the CAD model according to the results of analysis. Although much research on automatic mesh generation for FEA has been done, the challenge of improving the robustness and efficiency of meshing still remains. Therefore, direct modification of the mesh is highly necessary for avoiding the iterative meshing process.

Dimension-driven mesh deformation is very useful for the efficient modification of the mesh model of the product shape in CAE. Mesh deformation based on dimensions allows us to effectively modify the design of the product shape, as in CAD systems. Moreover, it provides an efficient parametric survey for finding optimal dimensions without the CAD model, and may bridge the gap between the CAD and CAE models.

Much research on mesh deformation has been conducted for mesh-based geometric modeling. Current mesh deformation methods can be classified into space-based, surface-based and simple representation-based methods. Space-based deformation deforms the mesh models by embedding the mesh into a space spanned by some basis functions and deforming the space while manipulating the handle (control points) [5,9,10]. Surface-based deformation describes the mesh geometry by local differential equations and deforms the model by changing the boundary conditions and solving equations [11,14]. An elegant combined method of these methods has also been proposed for assembly model deformation [8]. Simple representation-based deformation makes geometric relationships between mesh vertices and simple representations such as sketches and simplified meshes, and deforms the models by manipulating them [6,15]. These methods achieve flexible and intuitive deformation of mesh models. However, they do not discuss the modification of the dimensions of the model.

This paper proposes a triangular mesh deformation method based on dimensions. Our method is based on space-based deformation using barycentric coordinates. In space-based deformation, creating a deformation handle suitable for the intended deformation is a practical use issue because the degree of freedom for deformation is determined by the number of handle control points. Manipulating the handle appropriately is also important to obtain the intended shapes, because the user does not manipulate the model geometry directly in the basic approach. Therefore, we introduce a method for creating a handle suitable for dimension-based deformation and automatic handle manipulation depending on the target dimensions.

In section 2, the problem settings and basic considerations for dimension-based mesh deformation are described. Then, a mesh deformation method using barycentric coordinates is introduced in section 3. The procedures for dimension-based mesh deformation and the formation of detailed algorithms are proposed in section 4. The results of our method are shown in section 5.

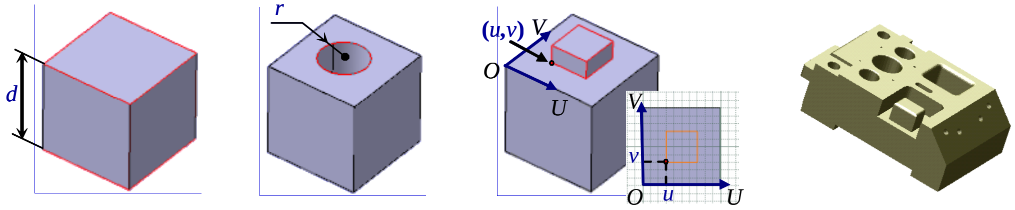
2. DIMENSIONS AND SPACES FOR DEFORMATION

2.1 Dimensions

Many types of dimensions are used in geometric modeling and implemented in CAD systems. In this paper, we address the following three, which are highly useful in the design modification of mechanical parts (Fig. 1):

A) Distance between two parallel planes (Fig. 1(a))
B) Radius of the cylinder (Fig. 1(b))
C) Position of the local object on the plane (Fig. 1(c))

These dimensions are often equivalent to the parameters of major form features in mechanical parts. The distance between two parallel planes (DP) corresponds to the distance of the padding (extrusion) of the sketch and the depth of pockets or blind holes. The radius of the cylinder (RC) can change the radius of the cylindrical holes or bosses. The position of the local object on the plane (PO) can modify the positions of the form features defined on the plane. Fig. 1(d) shows an example of mechanical parts. In this model, it is possible to modify all parameters of the form features used for defining its geometry by these dimensions.



(a) Distance between two parallel planes (b) Radius of the cylinder (c) Position of the local object on the plane (d) An example of mechanical part

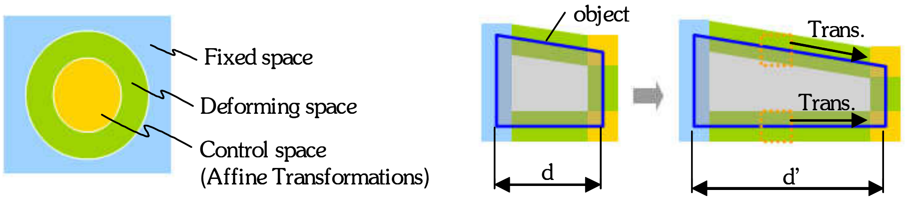
Fig. 1: Dimensions.

2.2 Spaces for Dimension-based Mesh Deformation

Changing the above dimensions is achieved using affine transformations: translation of a planar face or local object for the DP and PO, and scaling of cylindrical faces for the RC. Here, the face and local object indicate a meaningful set of triangles in the mesh, such as a planar region, cylindrical region or the form features. Due to the transformation of the faces or objects, their neighboring faces must be deformed in order to preserve geometrical consistency. Therefore, we define three spaces for dimension-based deformation: the *control space*, where affine transformations are applied, the *fixed space*, which does not changed during deformation, and the *deformation space*, which is deformed depending on changes in the control space. The deformation space exists between the control and fixed spaces, as shown in Fig. 2(a), because it absorbs the effects of changes in the control space and prevents these effects from reaching the fixed space.

A simple arrangement of spaces is achieved by assigning the control space to the face or local object to be transformed, the deformation space to its adjacent faces, and the fixed space to the others. However, the transformed face or local object must often be deformed in order to preserve the geometries of the adjacent faces. For example, in the case of changing the DP, a planar face adjacent to some planar faces must be deformed in order to preserve the planes of these adjacent faces, as shown in Fig. 2(b). To address the deformation of the transformed face, representative vertices are adopted at the points where some faces meet. By assigning the control spaces only to the representative vertices and the deformation spaces to the faces among them, the deformation of the transformed face can be achieved in order to preserve the geometries of the adjacent faces. Here, the representative vertices are moved along the planes of the adjacent faces.

In our method, the deformation handle consists of a set of convex triangular prisms. Each triangular prism is created by connecting representative vertices, including mesh vertices. In mesh deformation, the affine transformation with appropriate parameters is applied to the representative vertices of the transformed faces, and the positions of the mesh vertices in the triangular prisms deformed by affine transformation are updated by space-based deformation using barycentric coordinates as described in the next section.



(a) Spaces for deformation (b) An example of space arrangement

Fig. 2: Spaces for deformation.

3. MESH DEFORMATION USING BARYCENTRIC COORDINATES

In this section, we first give the definition of the barycentric coordinates for a convex polytope proposed by Warren et al. [12], and we then introduce mesh deformation using these coordinates [9]. Another method for space-based deformation can be useful. For example, the mean value coordinates proposed by Ju et al. [5] allow us to use a non-convex polyhedron for the deformation handle. However, our method creates the deformation handle, which consists of a set of convex triangular prisms (*simple* convex polytopes) as described in section 4.2, so barycentric coordinates were used for simplicity and efficiency.

First, we consider a *simple* polytope, which is constructed only of *simple* vertices. Given a point $\mathbf{x}$ in the polytope *P*, the weight $w_i(\mathbf{x})$ for a vertex *i* of the polytope is defined as the ratio of the parallelepiped volume made from the unit normal vectors $\mathbf{n}^i_{\sigma 1}, \mathbf{n}^i_{\sigma 2}, \mathbf{n}^i_{\sigma 3}$ of the adjacent three faces of *i*, and the product of the distances between each of the neighboring faces and $\mathbf{x}$.

$$w_i(\mathbf{x}) = \frac{\left(\mathbf{n}^i_{\sigma 1} \times \mathbf{n}^i_{\sigma 2}\right) \cdot \mathbf{n}^i_{\sigma 3}}{\prod_{k=1}^{3} (\mathbf{p}_i - \mathbf{x}) \cdot \mathbf{n}^i_{\sigma k}} \tag{2.1}$$

where $\mathbf{p}_i$ is the position of vertex *i* of the polytope. The barycentric coordinates $b_i(\mathbf{x})$ are defined by normalizing weight $w_i(\mathbf{x})$ for all vertices of the polytope as Eq. (2.2).

$$b_i(\mathbf{x}) = \frac{w_i(\mathbf{x})}{\sum_{j \in V_P} w_j(\mathbf{x})} \tag{2.2}$$

where V_P is a set of the vertices of polytope *P*. If the polytope is not simple, it has non-simple vertices and we cannot calculate the weights using Eq. (2.1). Then, each non-simple vertex is split into simple vertices by perturbing its adjacent faces, and their barycentric coordinates are calculated. Finally, by adding the barycentric coordinates of the simple vertices, the barycentric coordinates for a non-simple vertex can be obtained.

In mesh deformation using barycentric coordinates [9], a handle is first defined by the convex polytope so that it includes the local/global mesh to be deformed, and the barycentric coordinates of the mesh vertices in the handle are calculated using Eq. (2.2). Then, the positions of the handle vertices are changed by the user. Finally, the positions of the mesh vertices are updated using Eq. (2.3).

$$\mathbf{p}^M_i \leftarrow \sum_{j \in V_H} \mathbf{p}^H_j b_j(\mathbf{p}^M_i) \tag{2.3}$$

where $\mathbf{p}_i^M$ denotes the position of mesh vertex i, $\mathbf{p}_j^H$ the position of handle vertex j, and V_H a set of the vertices of handle H.

In our approach described below, the handle consists of a set of convex triangular prisms, which are not connected. Each prism includes the local vertices of the given mesh. In the deformation, some prisms become the boundary of the deforming space based on the user-selected dimension types and regions which define the dimension, and then some of their vertices move by affine transformations. Finally, the positions of inner vertices of the deformed prisms are modified using Eq. (2.3).

4. MESH DEFORMATION BASED ON DIMENSIONS

4.1 Deformation Procedure

The proposed mesh deformation method is shown in Fig. 3. By following these three steps, the user can change the arbitrary dimensions of a given mesh model.

(1) Selecting the type of dimension.
(2) Selecting the regions which define the dimension.
(3) Specifying the target value of the dimension.

The system first creates a deformation handle (section 4.2). Then, according to the type of dimension and selected regions, the dimensions of the model are calculated (section 4.3). Finally, the mesh model is deformed so as to satisfy the target dimension by manipulating the handle using the user-specified target value of the dimension (section 4.4).

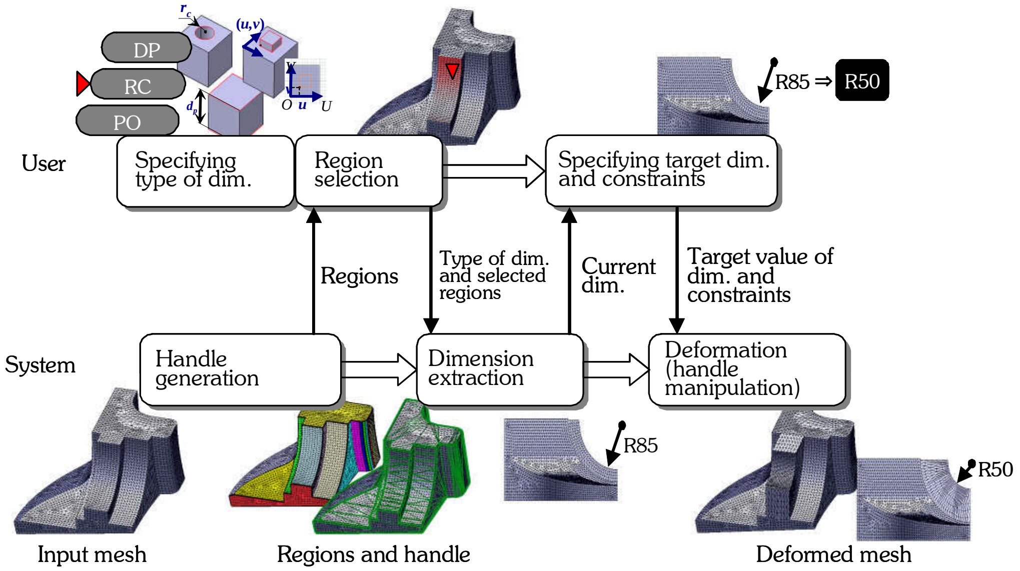


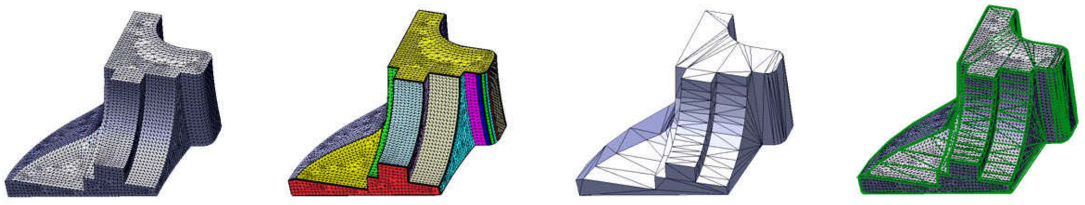
Fig. 3: Procedure for dimension-based mesh deformation.

4.2 Handle Generation

In this section, a method for creating the global handle of a given mesh is described. Fig.4 shows the procedure for the handle generation. First, a given mesh model is segmented into regions (Fig. 4(b)) to find the planar and cylindrical regions, which are selected by the user for determining the dimensions to be modified, and then, the surface parameters of each region are calculated in order to extract the dimensions. Then, a simplified version of the given mesh is created to find the base triangles and edges of the triangular prisms, which are the components of the global handle (Fig. 4(c)). Finally, by offsetting the triangles and edges of the simplified mesh, a set of triangular prisms is generated as the deformation handle (Fig. 4(d)).

In this procedure, many existing methods for mesh segmentation [2,3], surface fitting for parameter extraction [1,13], and mesh simplification [4,7] are useful, and we used some of them in our implementation (The implementation details are described in section 5.1). In this section, only the method concerning handle construction is described (from

Fig. 4(c) to Fig. 4(d)). Therefore, we assume that the mesh is segmented into the regions and each region is classified as a plane, a cylinder with surface parameters or other, and a simplified version of the given mesh is obtained.

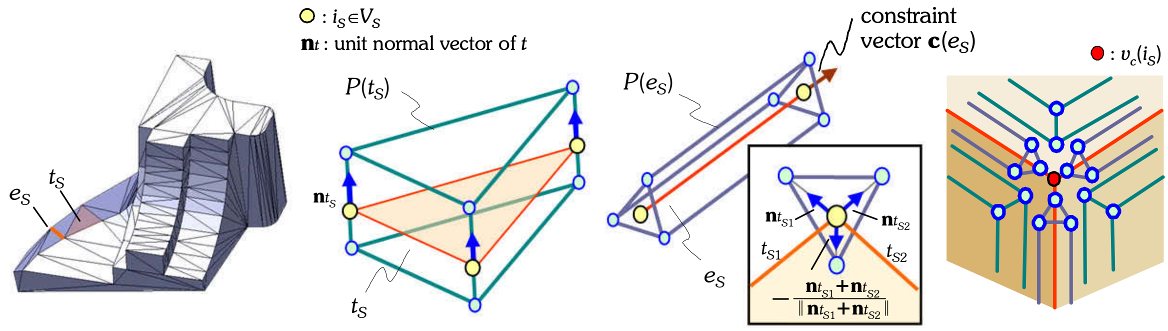


(a) Original mesh (b) Segmentation (regions) (c) Simplification (d) Offset (handle and original mesh)

Fig. 4: Deformation handle generation.

First, we would like to present some of the notations used in the following algorithm. We denote the given mesh as M_O=<V_O, E_O, T_O>, and the simplified mesh as M_S=<V_S, E_S, T_S>, where $V_g=\{i_g\}$, $E_g=\{e_g=(i_g, j_g)\}$, and $T_g=\{t_g=(i_g, j_g, k_g)\}$ (g=O or S) are sets of vertices, edges, and triangles of each mesh. We assume that each vertex i_S in V_S has a corresponding vertex $v_c(i_S)$ in V_O. Such a vertex can be found directly, when we use certain mesh simplification operators, such as edge collapse and vertex removal [4,7]. Each planar region has four parameters (a, b, c, d) of the fitted plane ($a^2+b^2+c^2=1$). Each cylindrical region has axis direction vector $\mathbf{d}$, a point on the axis $\mathbf{a}$, and radius r of the fitted cylinder.

The deformation handle, which consists of a set of triangular prisms, including mesh vertices in V_O, is created by splitting and offsetting the vertices of the edges and triangles of the simplified mesh M_S (Fig. 5(a)). The details of handle generation procedure are as follows: each vertex of the triangle t_S in the T_S are split into the outside and inside of the surface, as shown in Fig. 5(b). As a result, a triangular prism $P(t_S)$ is created for each triangle. For each edge e_S, except for the one on the same planar region, each endpoint is divided into three. If e_S is convex, two vertices are generated outside of the surface, and one is generated inside, as shown in Fig. 5(c), and vice versa if e_S is concave. As a result, as in the case of the triangle, the triangular prism $P(e_S)$ is created for e_S. The unit direction vector of e_S is assigned to each $P(e_S)$ as *constraint vector* $\mathbf{c}(e_S)$.



(a) Simplified mesh (b) A prism for a triangle (c) A prism for an edge (d) Representative vertices

Fig. 5: Triangular prisms of the deformation handle.

For triangles, the direction of the offset of its three vertices is that of its normal. For the edges, two vertices are generated along the normal directions of the two incident triangles, and one vertex is created along their average normal direction. These directions are able to include all vertices in V_O except for $v_c(i_S)$ using certain offset distances, and they create a set of convex prisms (simple convex polytope) as shown in Fig. 5. Therefore, the barycentric coordinates of each vertex in V_O can be calculated for the corresponding prism. In the final step of the handle construction, the vertices in V_O included in $P(t_S)$ and $P(e_S)$ except for $v_c(i_S)$, are found successively, and the barycentric coordinates of each vertex are calculated for the corresponding prism using Eq. (2.2). Here, we define *representative* *vertices* for each vertex $v_c(i_S)$, which consist of the vertices of the prisms generated by splitting i_S, and $v_c(i_S)$ as shown in Fig. 5(d). As a result, both a set of triangular prisms and a set of representative vertices are obtained.

4.3 Region Selection and Dimension Extraction

As described in section 4.1, the user first selects the dimension type from among the DP, RC and PO. Then, depending on the dimension type chosen, some regions obtained by mesh segmentation and reference points are selected by the user. For each dimension type, the user selects following regions and points.

DP: two parallel planar regions as the region to be transformed r_A and the reference region r_B
RC: one cylindrical region r_C
PO: a base planar region r_b where the target object exists, two points $\mathbf{p}_o$ and $\mathbf{p}_a$ on the base region for defining local coordinate system (origin and a point on the local axis), regions r_{obj} defining the object, and a reference point $\mathbf{p}_{obj}$ of the object

In PO modification, local coordinate system is defined by the origin $\mathbf{p}_o$, a axis direction vector $\mathbf{u}=(\mathbf{p}_a-\mathbf{p}_o) / \| \mathbf{p}_a-\mathbf{p}_o \|$, and its orthogonal vector on the base planar region, $\mathbf{v}=(\mathbf{n}_{rb}\times\mathbf{u}) / \| \mathbf{n}_{rb}\times\mathbf{u} \|$, where $\mathbf{n}_{rb}$ is the normal vector of r_b.

For the DP, the current dimension d_{crr} is the determined by $d_{rA} - d_{rB}$, where d_{rA} and d_{rB} are the fourth coefficients of the planes of the regions r_A and r_B respectively. For the RC, the current dimension r_{crr} is the radius of the cylinder of the region r_C. In the PO, the current object position on the plane is given by $(u_{crr}, v_{crr})=((\mathbf{p}_{obj}-\mathbf{p}_o)\bullet\mathbf{u}, (\mathbf{p}_{obj}-\mathbf{p}_o)\bullet\mathbf{v})$.

4.4 Deformation

Using the extracted current dimensions and target dimensions given by the user, the mesh is deformed so that it satisfies the user-specified dimensions. In this step, the deformation, control, and fixed spaces described in section 2.2 are first assigned. The deformation spaces are assigned to the inside of the triangular prisms, which include vertex in $V_R(r_t)$, where $V_R(r)$ is the representative vertices including the vertex in region r of M_O, and r_t is the region to be transformed, i.e. $r_A/r_C/r_{obj}$ in DP/RC/PO. The control spaces include $V_R(r_t)$. The control spaces are not explicitly defined because only affine transformation is applied to the vertices in the control space. The remaining representative vertices and prisms are included in the fixed spaces.

Given the target dimension d_{trg}, DP modification is done by applying $Trans(\mathbf{n}_{rA}, d_{trg}-d_{crr})$ to each vertex in $V_R(r_A)$, where $Trans(\mathbf{n}, d)$ means the translation by d along vector $\mathbf{n}$, $\mathbf{n}_{rA}$ is the unit normal vector of the plane of r_A. Constraints for preserving the planes of the neighboring regions of the transformed region r_A can be easily imposed using constraint vector $\mathbf{c}(e_S)$ of $P(e_S)$. If only one of the endpoints $v_c(i_S)$ of the edge e_S is included in the control space, the corresponding representative vertices of it are moved along $\mathbf{c}(e_S)$ (In DP modification, they are translated by $Trans(\mathbf{c}, (d_{trg}-d_{crr})/\mathbf{c}\bullet\mathbf{n}_{rA})$). For the RC, given the target radius r_{trg}, scaling by r_{trg}/r_{crr} is applied to the $V_R(r_C)$ on the plane perpendicular to the axis of the cylinder. In the PO, given target dimension (u_{trg}, v_{trg}), the positions of the vertices in $V_R(r_C)$ are obtained by $Trans(\mathbf{u}, u_{trg}-u_{crr})Trans(\mathbf{v}, v_{trg}-v_{crr})$.

After changing the positions of the representative vertices in control spaces, the positions of vertices in V_O are updated using barycentric coordinates by Eq. (2.3).

5. IMPLEMENTATION AND RESULTS

5.1 Used Algorithms and Implementation

For mesh segmentation and surface parameter extraction, we indicated in section 4.2 that we have the option of using the existing methods. In our implementation, we used dihedral angle-based segmentation for simplicity. First, dihedral angles between two faces sharing an edge are evaluated, and then the edges are identified as feature edges, if the angle is smaller than the user-specified threshold. The region surrounded by the feature edges is identified as a region. After segmentation, planes and cylinders are fitted to each region respectively. We used the method described in [1] for the plane and cylinder fitting.

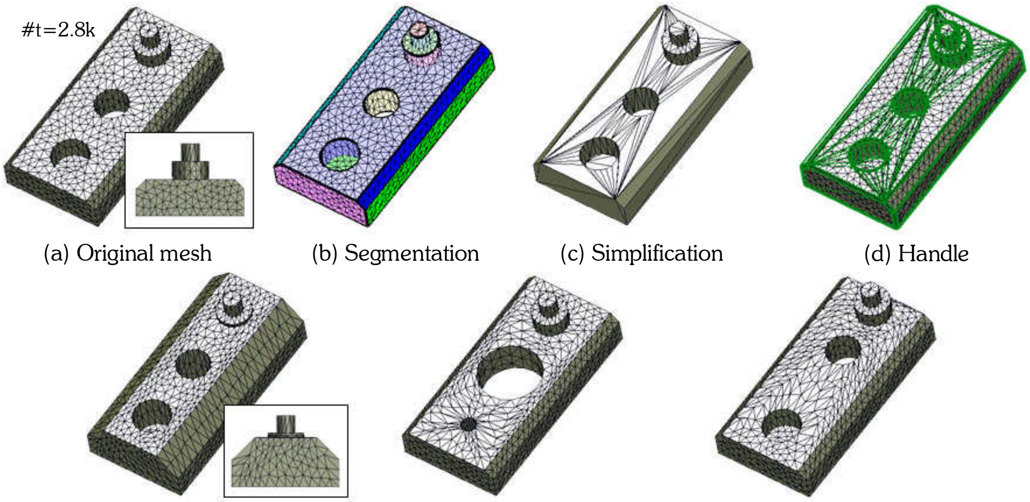
Additionally, we leave a choice in the method to be used for mesh simplification in handle creation. Variational shape approximation (VSA) method with $L^{2,1}$ norm [3] was used for obtaining simplified mesh in our implementation, because it provides good approximation and straightforward correspondences between the mesh vertices in the simplified and original meshes. The VSA was applied so that each triangle cluster does not go over the region resulting from segmentation. This was achieved by inserting the adjacent triangles on same region in the queue in the flooding step. Simplified triangular mesh was generated by triangulation method using the flooding described in [3].

In our system, the user interactively selects the regions resulting of segmentation. However, in the case of PO modification, the user often has to select many regions, which define the local object to be moved. Therefore, we designed the software so that the user can select the regions of the object on the base plane at a time (for example, a boss in Fig. 6(a)) or only regions connecting to the base plane (for example, a region for through hole in Fig. 6(a)). These selection approaches allowed the user to efficiently determine the regions for the deformation.

5.2 Results

Fig. 6 shows the results of dimension-based mesh deformations for a simple shape. Each figure shows the original mesh (a), results of segmentation (b), simplified mesh using VSA (c), handle (d), and the results of changing DP (e), RC (f), and PO (g). Figs. 7 and 8 show other examples of the design modification of a mechanical part, and connecting rod consisting of some parts. In these examples, we generate the original meshes by FE meshing of solid models. The numbers of the triangles of original meshes are shown in each figure as #t. The handle generation took less than few seconds (within 0.8 sec for the models in Figs. 6-8, and 2.1 sec for the fandisk model with 12.9k triangles in Figs. 3-4), and all deformations was achieved instantaneously (In our implementation, vertex position modification using barycentric coordinates in Eq.(2.3) took less than 30 msec for three hundred thousand mesh vertices and ten handle vertices) using a PC (P4-3.8GHz, RAM 2GB).

The results of deformations show that the proposed method is useful for modifying the design of mechanical parts. Additionally, we confirmed that the constraints for preserving the planes of the neighbors of the transformed region worked well, as in Fig. 6 and 7. In the case shown in Fig. 8, dimension-based deformation easily provided consistent deformation in assembly model using the same target dimensions for corresponding regions. Moreover, using resulting dimensions, the CAD model could be easily modified and dimension-based mesh deformation demonstrated an ability to bridge the CAD and CAE models.

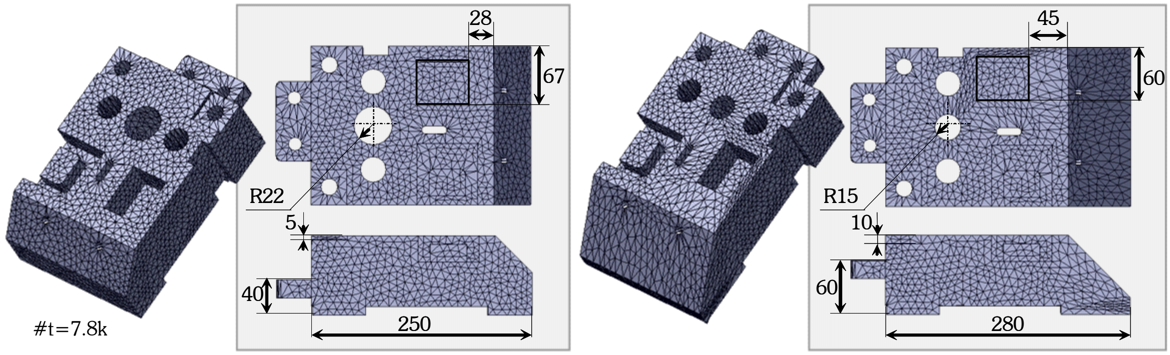


(a) Original mesh (b) Segmentation (c) Simplification (d) Handle

(e) Changing DP (f) Changing RC (g) Changing PO

Fig. 6: Simple examples of dimension-based deformation.

In current approach, large deformations cause large distortion of the triangles resulting in the frequent occurrence of face flipping due to the deformation. In our method, the deformable range, given consistent geometry without flipping, is determined by the connectivity of the simplified mesh, which defines the structure of the handle (The flipping of the prisms of the handle by affine transformation always causes the face flipping of the mesh). For large and flexible dimension-based deformation, the optimal simplified mesh generation and dynamic modification of the handle structure considering the target dimension and original geometry, and also connectivity modification of the mesh is required. Moreover, volume mesh must be handled for application of the method to efficient CAE. These are our main works for the future.



(a) Original mesh (b) Deformed mesh

Fig. 7: The deformation of a mechanical part model.

6. CONCLUSIONS

In this paper, a triangular mesh deformation method based on dimensions was proposed. First, we defined the spaces for dimension-based deformation and stated that the control, deformation and fixed spaces are useful for dimension-based deformation. Then, we showed a procedure for mesh deformation based on dimensions, along with detailed algorithms for creating and manipulating deformation handle. Our handle creation method used segmentation, surface fitting, mesh simplification and offsetting. The resulting handle was constructed of only triangular prisms. We also described in detail the handle manipulation method for deformation with constraints. Our results for mechanical parts showed that our approach was able to modify mesh shapes by dimensions in real time. The results also indicate that dimension based deformation had ability to provide consistent deformation of assembly model and to bridge CAD and CAE models using dimensions. Future works include introducing dynamic handle modification for more flexible deformation without face flipping, and extending the dimension-based deformation to volume meshes for efficient CAE.

7. REFERENCES

[1] Attene, M.; Falcidieno, B.; Spagnuolo, M.: Hierarchical mesh segmentation based on fitting primitives, The Visual Computer, 22(3), 2006, 181-193.

[2] Benkő, P.; Várady, T.: Segmentation methods for smooth point regions of conventional engineering objects, Computer-Aided Design, 36(6), 2004, 511-523.

[3] Cohen-Steiner, D.; Alliez, P.; Desbrun, M.: Variational Shape Approximations, ACM Transactions on Graphics, 23(3), 2004, 905-914.

[4] Hoppe, H.: Progressive meshes, Proc. of SIGGRAPH'96, 1996, 99-108.

[5] Ju, T.; Schaefer, S.; Warren, J.: Mean value coordinates for closed triangular meshes, ACM Transactions on Graphics, 24(3), 2005, 561-566.

[6] Kho, Y.; Garland, M.: Sketching mesh deformations, Proc. of the 2005 symposium on interactive 3D graphics and games, 2005, 147-154.

[7] Luebke, D.; Reddy, M.; Cohen, J. D.; Varshney, A.: Level of Detail for 3D Graphics, Morgan Kaufmann, 2002.

[8] Masuda, H.: Feature-preserving Deformation for Assembly Models, Computer-Aided Design & Applications, 4, 2007, 311-320.

[9] Mizuno G.; Date, H.; Kanai, S.; Kishinami, T.: A Parametric Deformation Method for Mesh Models based on Barycentric Coordinates, Proc. of Visualization, Imaging and Image Processing 2006, 2006, 319-324.

[10] Sederberg, T. W.; Parry, S. R.: Free-Form Deformation of Solid Geometric Models. Proc. of SIGGRAPH' 86, 1986, 151-160.

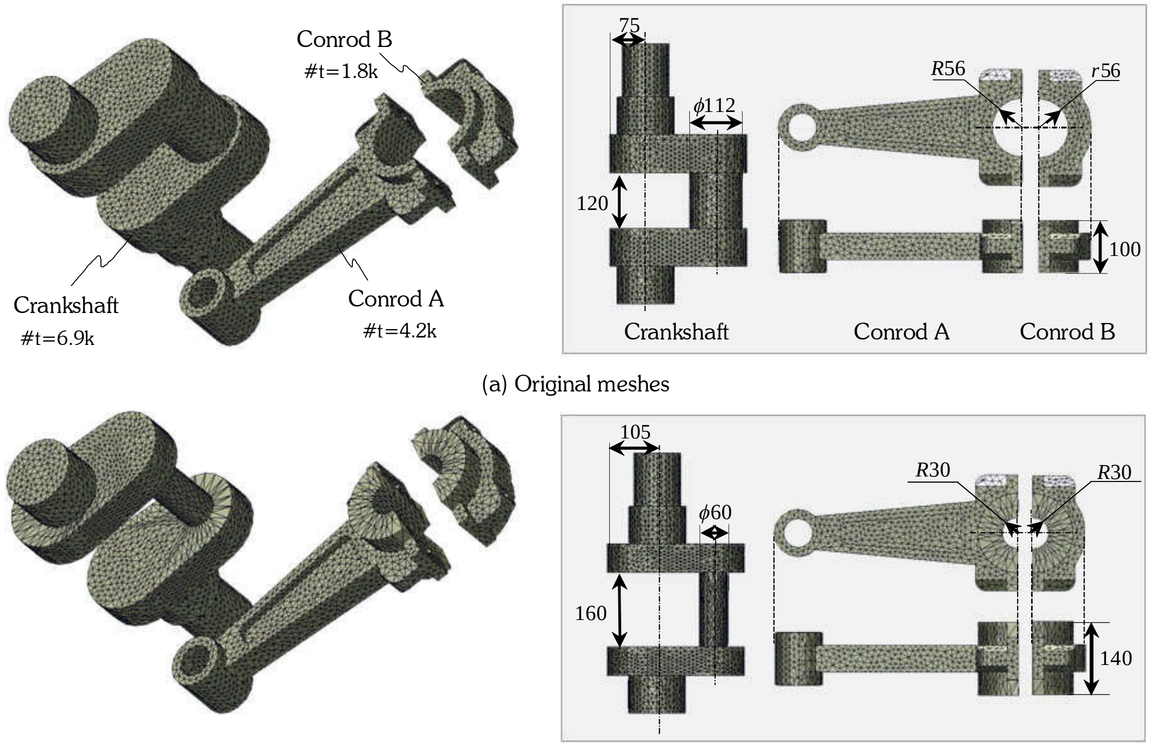
[11] Sorkine, O.; Cohen-Or, D.; Lipman, Y.; Alexa, M.; Rössl, C.; Seidel, H.-P.: Laplacian Surface Editing, Proc. of the 2004 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing, 2004, 175–184.

[12] Warren, J.; Schaefer, S.; Hirani, A. N., Desbrun, M.: Barycentric Coordinates for Convex Sets, Advanced in Computational Mathematics, 27(3), 2007, 319-338.

[13] Yan, D.-M.; Liu, Y.; Wang, W.: Quadric Surface Extraction by Variational Shape Approximation, Lecture Notes in Computer Science, 4077/2006, 2006, 73-86.

[14] Yu, Y.; Zhou, K.; Xu, D.; Shi, X.; Bao, H.; Guo, B.; Shum, H.-Y.: Mesh Editing with Poisson-Based Gradient Field Manipulation, ACM Transactions on Graphics, 23(3), 2004, 644–651.

[15] Zorin, D.; Schröder, P.; Sweldens, W.: Interactive Multiresolution Mesh Editing, Proc. of SIGGRAPH'97, 1997, 259-268.



(b) Deformed meshes

Fig. 8: The deformation of connecting rod parts and crankshaft.