IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received August 20, 2020, accepted September 7, 2020, date of publication September 10, 2020,
date of current version September 24, 2020.

*Digital Object Identifier 10.1109/ACCESS.2020.3023281*

# Developing an Image-Based 3D Model Editing Method

**MIN PANG[1], LIGANG HE[2], (Member, IEEE), FENGGUANG XIONG[1], XIAOWEN YANG[1], ZHIYING HE[3], AND XIE HAN[1]**

[1]School of Data Science and Technology, North University of China, Taiyuan 030051, China
[2]Department of Computer Science, The University of Warwick, Coventry CV4 7AL, U.K.
[3]Software School, North University of China, Taiyuan 030051, China

Corresponding author: Xie Han (hanxie@nuc.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61672473, in part by the Key Research and Development Program of Shanxi Province of China under Grant 201803D121081 and Grant 201903D121147, in part by the Natural Science Foundation of Shanxi Province of China under Grant 201901D111150, and in part by the Research Project, Shanxi Scholarship Council of China, under Grant 2020-113.

**ABSTRACT** As 3D technologies advance rapidly, 3D printing, 3D animation, and 3D Movie are springing out in different areas. It becomes a remarkable challenge to generate a large number of 3D models effectively and efficiently. This paper proposes a novel editing method based on the feature lines of images (i.e., image contour and principal axis) for generating new 3D models. Our method takes as input an existing 3D model (as the original model) and an image selected by the user (or a sketch hand-drawn by the users), and performs the model editing to generate a new 3D model. In particular, our method first takes as input an original 3D model and an image selected by the user. Second, the selected image is processed to produce the feature lines, i.e., contour and principal axis of the image. Third, the silhouette of the original model from a given view is acquired, and projected on a projection plane to produce a contour and principal axis of the model, which is the feature lines of the original model. Fourth, by comparing the feature lines of the image and the original model, the constraint conditions are established to control the editing of the 3D model. Finally, 3D model editing is conducted through the as-rigid-as-possible mesh deformation to produce a new 3D model with the appearance resembling the selected image. Furthermore, this paper proposes an energy function to guide the detailed model editing, and measure the similarity between the generated 3D model and the corresponding image. We have conducted extensive experiments to evaluate the proposed method. The results show that comparing with the existing editing methods in literature, the proposed model editing method is able to construct various types of 3D models more effectively and more efficiently.

**INDEX TERMS** 3D original model, energy function, feature lines, image, mesh deformation.

## I. INTRODUCTION

The development of 3D printing technology and the maturity of 3D animation technology and the computer-aided design technology require more 3D models. It imposes a great challenge to acquire a large number of 3D models efficiently and effectively. 3D models can be acquired not only by a 3D scanning equipment such as Kinect, but also by the 3D model editing technique [1], [2]. Model editing is an important research topic in computer graphics. It finds a wide range of applications such as animation, cinematography, games, computer aided design (CAD), 3D printing, medical diagnosis and treatment, virtual reality, mixed reality and so on.

Model editing is most useful at the early stage of modelling in that it can provide basic 3D models in various applications [1]–[4]. Nowadays, most model editing methods are based on some existing 3D models, aiming to achieve the high efficiency of 3D model geometric design. The model editing method works by transforming the existing 3D models to construct novel 3D models with different appearances. The model editing process often welcomes the users' interaction to express their creative ideas. This way, it can achieve high quality and high efficiency of 3D model reuse.

There are abundant digital images on the Internet, which are easily accessible. Contours of images or graphics can be used to sketch the main features of a shape. People often use such a simple and fast sketching method to express their visualized thoughts. This is also why designers often

The associate editor coordinating the review of this manuscript and approving it for publication was Yue Zhang.

implement their design with hand-drawn sketches in the early modeling stage [2], [3]. 3D remodeling based on hand-drawn sketches or images has become an important research topic in computer graphics and computer vision [5]–[10]. Developing more efficient and accurate image-based 3D model editing methods is always a major research direction in the topic. There are some existing works in the literature. Hu *et al.* [5] propose a method to control the surface mesh deformation by drawing the sketches of the orthographic projection of three-dimensional model. This method only supports the deformation guided by the hand-drawn sketches. So it is difficult to obtain the satisfactory deformation results for non-professional designers. Also, because this method uses the Laplace deformation algorithm, it is easy to produce the distortion for large deformation and it is insensitive to rotations. Hou *et al.* [7] propose a learning based image registration method, and utilize a Convolutional Neural Network (CNN) architecture to learn the regression function capable of mapping 2D image slices to a 3D canonical atlas space. The target of their method is mainly Magnetic Resonance Imaging (MRI). Shin *et al.* [10] propose a method that reconstructs the 3D terrain model from the contour lines on the 2D map. However, it is more suitable for generating 3D terrain models, not for other types of 3D model. Wu *et al.* [8] proposed the method of reconstructing a 3D object from a single image. The method, called SliceNet, proposes to generate 2D slices of a 3D shape sequentially with shared 2D deconvolution parameters. Their method is suitable for the reconstruction of 3D objects, similar to our method. However, the object that the algorithm can be applied to is limited to the model type established in the deep learning process. Our method does not have such a limitation, and so has a wider range of applications. Fu *et al.* [26] proposed to realize the mesh deformation by comparing natural lines such as axis, contour line and cross-section line of 3D models with the image contour and skeleton. However, their editing method only considers the distance and does not make use of the angle feature in the process of establishing the feature line comparison. Also, it needs to compare the deformation of cross-section lines several times. Therefore, the effect and efficiency of their method are not as good as our method. In light of the above discussion, this paper proposes a new model editing method that uses the two-dimensional images or hand-drawn sketches to guide the editing of 3D models. Our method can generate 3D models with different geometric shapes that incorporate the users' visualized thoughts in hand-drawn sketches or the visual feature of the 2D images selected by the users.

In this editing process, the user begins by extracting some feature lines of the target object from images containing natural or artificial objects. The feature lines include the *contour* and the *principal axis* of the target object. Next, the viewpoint of the existing original model is selected. According to the relationship between the points on the model and the lines of sight, the contour of the model is acquired. The model contour is composed of the common boundary trajectory of the visible and invisible parts of the model under the view [11].

Based on the feature points on the image contour, the constraint condition of the mesh deformation is determined from the correspondence between the image contour and the plane projection of the model contour. Finally, the improved as-rigid-as possible mesh deformation algorithm is performed to generate a model with a different appearance. Figure 1 shows the results from an editing process that uses the proposed

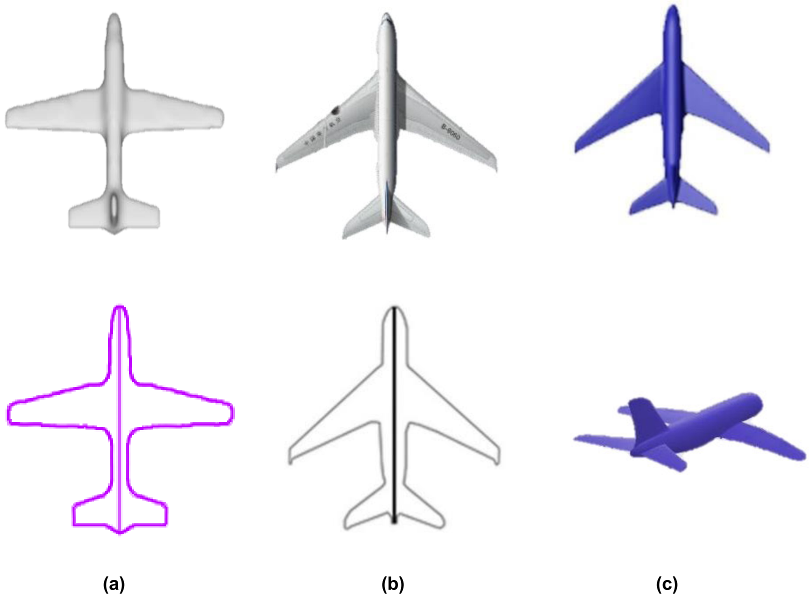


**FIGURE 1.** 3D model editing using the proposed method (a) the contour projection of the original 3D aircraft model; (b) the outer contour of the target image selected by the user; (c) the deformation 3D model based on the original model and the target image.

method, where Fig. 1a shows a 3D original model (top figure) and contour projection of model (bottom figure).

Figure 1b is the target object acquired from the image and the outer contour of the object. Figure 1c is the resulting 3D models after editing the original model in Fig. 1a using the proposed method under the guidance of the image in Fig. 1b. The generated model is a 3D model and can be viewed from different angles. The two figures in Fig. 1c are the projections from two perspectives of the generated 3Dmodel. The 3D editing method proposed in this work is able to generate the 3D models which are more similar to the target objects in the selected 2D images than those generated by the existing methods in the literature. At the same time, our method is more efficient and can generate new models in much less time than the existing methods.

The rest of this paper is organized as follows. In Section II, the related work is reviewed. In Section III, an overview of our image-based mesh editing technique is presented firstly, and then our image-based mesh editing technology is presented in detail. The experimental results are presented and discussed in Section IV. Section V concludes this paper and also discusses our research plan in future.

## II. RELATED WORK

In this section, we review the related research studies from the following three aspects: 1) the image processing techniques regarding image optimization, segmentation and contour extraction of the objects in images; 2) the differential-based mesh deformation algorithms; and 3) the image-based (or sketch-based) 3D model editing techniques.

### A. IMAGE PROCESSING

In computer image processing, many mature and well-established algorithms in image filtering, sharpening, target recognition, and segmentation of 2D images have been developed [12]–[18].

The image segmentation technology, which is still evolving, acquires the target objects based on image texture and edge information [12], and then uses an outer boundary tracking algorithm based on the topological analysis to detect object contours [14].

Arbeláez *et al.* [15] propose a new idea for contour detection and image segmentation, which makes use of a spectral clustering method and local clues to generate a global framework of a 2D image. Martin *et al.* [17] propose a method that constructs a linear model based on the color, light and texture of an image, applies the edge detection operator to extract edges, removes stray redundant edges depending on the contour feature of the target object. Finally the edge repairing is performed to produce the final contour of the target object.

### B. DIFFERENTIAL MESH DEFORMATION ALGORITHM

Differential mesh deformation algorithm, which is an important research subject in 3D model editing, marks a breakthrough in the development of 3D model editing.

The differential-based mesh deformation technique originated from the Laplacian differential coordinate-based mesh deformation algorithm proposed in 2004 by Sorkine *et al.* [19], Lipman *et al.* [20]. It works by mapping the Cartesian coordinate representation of the mesh vertices in the input model to the Laplacian coordinate representation. The latter represents the vertices of the mesh as an approximation of the average curvature of the vertices. In the process of mesh deformation, the local features of the model surface are preserved because of the constancy of the Laplacian coordinates, thereby achieving the constancy of local details on the surface. However, this method has a drawback in that it does not provide rotational invariance. If rotation is involved in mesh deformation, shape distortion will occur.

Sorkine and Alexa [21] propose an as-rigid-as-possible mesh deformation algorithm, which achieves the improvement by introducing the parameter $R_i$ into the energy function of the Laplacian differential mesh deformation, aiming to rectify the inadequacy of rotation invariance. Tang *et al.* [22] propose to couple the local and non-local factors for guiding the mesh deformation to the entire surface while preserving the global shape and local attributes of the surface to a maximum degree. Yu *et al.* propose the Poisson energy equation editing technique [23], which works by changing the vector field to the scalar field of the original mesh. This technique, as a mesh editing framework, not only achieves the mesh deformation, but finds satisfactory applications in mesh splicing and denoising. However, with this technique the deformation distortion is owing to its translation invariance. The translation process does not change the gradient field of the model. So the Poisson method is not sensitive to translation. More recently, Chen *et al.* [24] propose a rigidity-preserving shape deformation algorithm, which is capable of controlling the rigidity and allows the resizing of the neighborhood by means of automatic learning of objects in the deformation database. This method enhances the consistency of adjacent rigid transformation.

### C. IMAGE-BASED 3D MODEL EDITING METHOD

The 3D model editing method based on images or hand-drawn sketches provides a powerful paradigm for model editing. A few contours can often sketch the main features of an object. Designers have been accustomed to such a simple and fast sketching method to express their design ideas. Therefore, it would be useful to use the designers' hand-drawn sketches (or the images selected by the designers) to guide the model editing. In this subsection, we review the related research in this direction.

Zimmermann *et al.* [3] propose a sketch-based surface mesh editing method that is capable of preserving features of the mesh. First, the user determines the appropriate view and sketches around the silhouette. Second, the system segments the silhouette area of the projected surface, identifies the best matching part among all silhouette segments, derives the vertices in the surface mesh corresponding to the silhouette part, and selects a sub-region of the mesh to be modified.

Finally, the system uses the sketch and the vertex positions that are appropriately modified, together with the submeshes, to achieve the mesh deformation. This mesh editing method will produce distortion when large-scale deformation is carried out. Our proposed method strives to reduce the distortion in the process.

Hu *et al.* [5] propose a method for model designers to sketch on the orthographic projection and control the deformation of surface mesh. First, they obtain the contour of the orthogonal view projection of the model and sketch the desired contour on the orthographic projection plane. Based on the relationship between the original projected contour and the hand-drawn contour by the designer, the model is edited with the Laplacian deformation method. The Laplacian deformation algorithm for the surface mesh is insufficient in rotational sensitivity. In addition, the feature extraction method used in their paper for the hand-drawn contour considers the distance only. We believe that there is still room to extract the feature information more comprehensively.

Tan *et al.* [25] propose a method to use an image to drive the stylized deformation of a 3D mesh. This method represents an image as a planar mesh and establishes a correspondence between the planar mesh and the original mesh model, and then completes the style transfer for the original model by learning the image style, which results in a model that is the same style as the image. Their method is more suitable for the deformation of the same type of objects. In addition, in general the model editing methods proposed in the existing work consider either the image-guided or the sketch-guided model editing. The method proposed in this paper works with both guiding mediums for model editing.

Fu *et al.* [26] put forward the parametric mesh deformation of the model by comparing the natural lines such as axis, contour and cross-section in the Cartesian coordinate system. Their editing method misses the angle feature in the process of model deformation, and has to compare natural lines many times. The mesh editing method proposed in this paper exploits the contour comparison to construct the constraint conditions, and then uses the Laplace coordinates to maintain the local stability of the model. At the same time, the energy equation is used as the objective function to constrain the mesh deformation and limit the number of deformation iterations. In the process of deformation, our method not only maintains the angles well, but also provides higher deformation efficiency and accuracy.

Shin *et al.* [10] propose a method that reconstructs geometric models from the contour lines on 2D map. They consider the reconstruction for simple regions without branches, where only one tiling operation is needed to generate the triangular strips. If there are some branches in the contours, it partitions the contour lines into several sub-contours according to the number of vertices and their spatial distribution. However, their method is more suitable for the generation of 3D terrain model, not suitable for other types of 3D model.

Hou *et al.* [7] propose a learning based image registration method, and utilize a Convolutional Neural Network (CNN) architecture to learn the regression function capable of mapping 2D image slices to a 3D canonical atlas space. Their method is used for quantitative analysis of simulated Magnetic Resonance Imaging (MRI) and fetal brain imagery with synthetic motion. Wu *et al.* [8] propose an algorithm to reconstruct a 3D object from a single image. Their algorithm depends on much prior knowledge of 3D shapes and uses the deep learning method. They propose a method called SliceNet, which sequentially generates 2D slices of 3D shapes with shared 2D deconvolution parameters. Their method is suitable for the reconstruction of 3D objects, similar to our method. However, SliceNet needs to use CNN to learn prior knowledge before 3D reconstruction. Moreover, the reconstructed object is of the same type as the model of prior knowledge. In other words, the object that the algorithm can be applied to is limited by the model type established in the deep learning process. However, our method does not have such restrictions and can be applied to any type of models. Therefore, our method has a wider scope of application. In our experiments, we can use different types of 3D models, such as human body, teapot, airplane and so on.

Based on the discussions in previous sections, we summarize the differences of the related works including our method in Table 1.

## III. OUR METHOD

In this section, we first give an overview of our method, and then present the details in subsequent subsections.

Huge amounts of image data are available on the Internet, which can be used by designers to guide the creation of 3D models. Designers creating a 3D model from scratch are faced with problems such as poor precision, low efficiency, and considerable computational complexity. For this reason, new models are mostly acquired by means of editing existing models. The proposed method obtains feature lines of the target object by processing the hand-drawn sketches or the selected image containing the target object. It then identifies feature points on the contour line by analyzing local extremum nature and connectivity. From the contour of the object, its features, such as size, shape, and structure of the target object, can be estimated.

When processing a 3D original model, we first select the view and the region of interest. Then, the contour of the region from this view as well as the projection and the feature lines of the contour are acquired. Note that in our method, we have to select a standard front view of a 3D model, which affects the result of model editing. In the method, we first establish the triangle through the feature points on the contour and the principal axis of the target object. Then the contour of the 3D model is projected on the selected view to obtain the projected contour line. According to the length proportion principle, the corresponding image feature points are found on the projected model contour line, which are used as the control points. The triangle similar to the triangle in the image is established with the projection axis of the model contour being an edge, so as to locate the target position of the control

**TABLE 1.** Comparison of image-based 3D model reconstruction methods.

| References | Function | Principle | Applicable model |
|---|---|---|---|
| Fast Reconstruction of 3D Terrain Model from Contour Lines on 2D Maps [10] | Using contour lines on 2D map to create 3D terrain models. | Generating a triangle strip with single tiling operation for simple matching region that does not contain branch structures in the contour of 2D map | large-scale virtual environment rendering |
| 2D and 3D Deformable Models with Narrow Band Region Energy [16] | A narrow band region approach in explicit deformable models for 2D and 3D image segmentation. | Proposing a new narrow band region term for explicit deformable models driven by energy minimization. | real medical Data. |
| 3D Reconstruction in Canonical Co-ordinate Space from Arbitrarily Oriented 2D Images [7] | 3D reconstruction from arbitrarily oriented 2D Images | By utilizing a Convolutional Neural Network (CNN) architecture to learn the regression function capable of mapping 2D image slices to a 3D canonical atlas space. | Magnetic Resonance Imaging (MRI) |
| SLICENET: Slice-Wise 3D Shapes Reconstruction from Single Image [8] | 3D shape reconstruction from a single image | Using the deep learning method to capture prior knowledge of 3D shapes; using the shared 2D deconvolution parameters to generate 2D slices of 3D shapes. | ordinary 3D models; but the models have to be of the same type as those learned in the CNN training |
| Our proposed method | 3D model reconstruction from a single image | A similar triangle relationship is established between the feature line of image contour and the feature line of model contour projection; The changes of the control points during model deformation are obtained to constrain the mesh deformation. | ordinary 3D models |

point and consequently obtain its position change. However, if we cannot obtain the standard front view of the model but have to use another view, the position of the main axis will change in the contour projection, and the correct control point and the target position of the control point cannot be found.

The next step is the core step of the proposed method: comparing the feature lines of the image with those of the projection of the model's contour. The correspondence between the two curves is used to establish the constraint condition, which is used to guide the deformation (editing) of the model. This results in a new 3D model different from the original one.

In this paper, if the original model is rotationally symmetric, then the image-guided model editing can be performed from multiple views each differing by 90 degrees. Otherwise, only one view of the original model is selected before carrying on the model editing. Because the model editing algorithm proposed in this paper uses a single image to guide the model editing. Comparing the contour of the image with the contour of a single view of the 3D model is sufficient to determine the change of control points. Multiple views of the model will not improve the accuracy of the model deformation results.

Figure 2 illustrates the main procedure of our method. Our method begins with dividing the target object $O$ shown in Fig. 2a and extracting the target object's feature lines, i.e., the contour and the principal axis, as shown in Fig. 2b. Next, a suitable view is selected for the original 3D model as shown in Fig. 2e; the region of interest $P$ is determined, and the contour of the region is acquired and projected onto the projection plane to produce the feature lines of the 3D model, i.e. the contour and the principal axis of the 3D model, as shown in Fig. 2f. According to the length ratio constraint between the image contour and the model contour projection, which will be presented in detail in subsection III.C, the vertex correspondence is established, as shown in Fig. 2c and Fig. 2g. Then, using the similar triangle relationship proposed in this paper (presented in subsection III.C), the constraints of model editing are established, as shown in Fig. 2d and Fig. 2h. Finally, with the assistance of the as-rigid-as possible mesh deformation algorithm, the mesh deformation is converted to the optimization process of an energy function (presented in subsection III.D), which completes the deformation of the interested region. As the result, a model with a different appearance is generated, as shown in Fig. 2i.

In the following subsections, we will detail how to use the target image to guide the 3D model editing process. The user begins with inputting a 2D image containing the target object $O$, and then selects the original model from the 3D model database. The region of interest $P = (V, E, F)$ in the original model is chosen, where $P$ represents a triangular mesh of $n$ vertices; $V = \{v_0, v_1, \ldots v_{n-1}\}$ is the set of $n$ vertices; $E$ is the set of edges between the vertices in $V$; $F$ is the set of triangles formed by the edges in $E$. The Cartesian coordinates of a vertex are denoted by $p_i = (x_i, y_i, z_i)$.

The proposed method aims to use the feature lines of the target object $O$ in the image to guide the deformation of the region of interest $P$ and obtain a different surface mesh $P'$.

### A. EXTRACTION OF IMAGE CONTOUR

Nowadays, a large number of pictures containing various natural and artificial objects are available on the Internet. It is also very easy for users to take pictures of real-world objects. These pictures provide the abundant resources for the research in this paper.

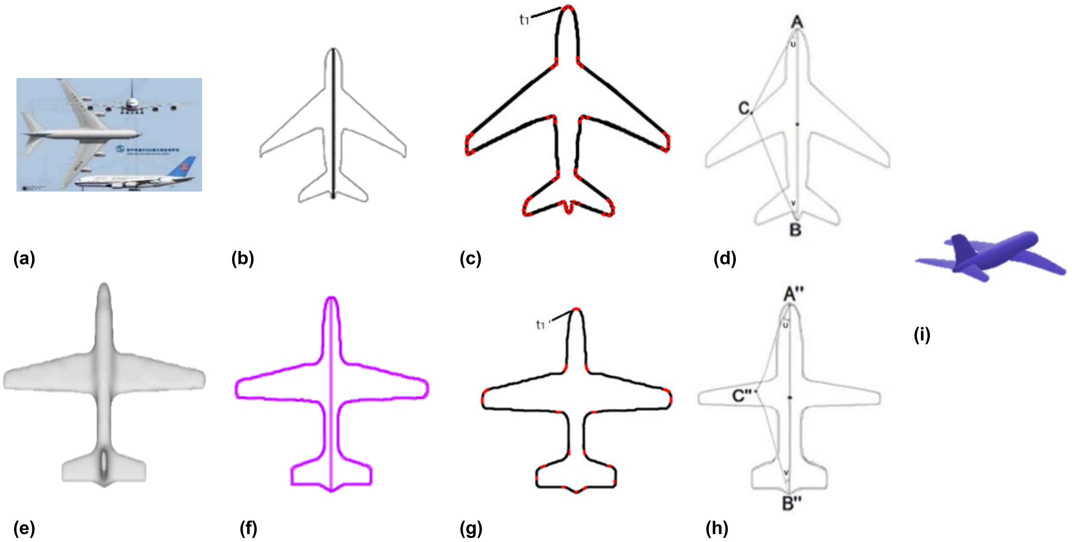


**FIGURE 2.** The review of the method (a) Original 2D image; (b) Acquired feature line of O; (c) Set T of feature points of the target reference line; (d) Triangle $\Delta ABC$ on the target reference line; (e) The 3D original model; (f) contour projection of this model; (g) Set $T'$ of corresponding points of the editable source line; (h) Similar triangle $\Delta A''B''C''$ on the editable source line; (i) the deformation result based on the original model and the target image.

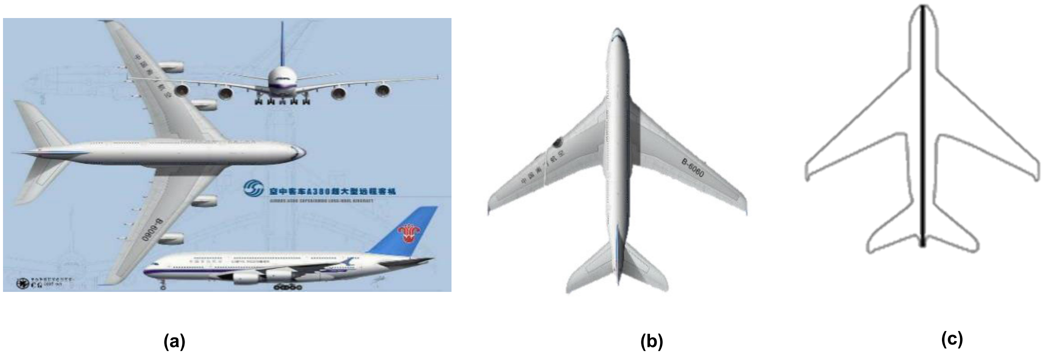


**FIGURE 3.** Contour extracted from a 2D image (a) Original 2D image; (b) Target object O; (c) Acquired feature line of O.

It is very important that the target image should contain a target object $O$ that is completely unmasked. The image meeting this requirement is firstly sharpened using the Laplacian operator. Then, the detection of a significant target is performed based on the difference between the foreground area and the background area of the target [27]. Next, the target object $O$ is cut out from the image by the grab cut [12] method. The contour line of the object $O$ is extracted [15] and subsequently normalized. The centroid and the principal axis of the contour are obtained [28]. The principal axis is the axis corresponding to the largest eigenvalue of the autocorrelation matrix obtained from the contour vertices. Figure 3 shows the result of the pre-processing, and the result of cutting and extracting of feature lines from the image, where Fig. 3a shows the original 2D image, Fig. 3b shows the target object extracted and cut out from the image, and Fig. 3c shows the feature lines, including the contour and the principal axis of the target object extracted from Fig. 3b.

As shown in Fig. 3, the proposed model editing method defines the outer contour of the target object as the target reference line.

### *B. EXTRACTING FEATURE LINES OF THE ORIGINAL MODEL*

The user selects an original model and chooses the region $P$ to be edited. Typically, the region should be a simple, component-level unit suitable for editing. For a simple model such as the one shown in Fig. 1, the entire model can be chosen as the region of interest. Then, we select a viewpoint, and the trajectory of the common boundary of the visible and invisible parts of the model is taken as the contour of this original model [11]. The contour of the original model obtained through Equation (1) is a sequence of discontinuous depth. As shown in Fig. 4, $s_0$ is the coordinate of the midpoint on the shared edge of the adjacent patch on the model surface.

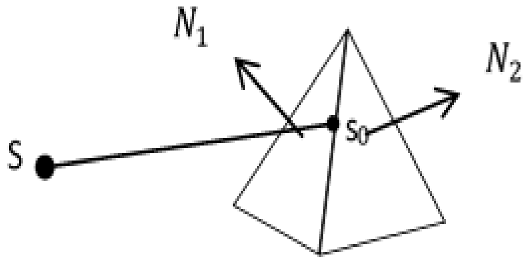


**FIGURE 4. Relation of the view vector and surface.**

$$((s_0 - S) \cdot N_1) * ((s_0 - S) \cdot N_2) \leq 0 \tag{1}$$

$S$ is the position of the current viewpoint, $s_0 - S$ is the view vector. $N_1$ and $N_2$ are the normal vectors of the adjacent faces. If the dot product of the view vector and the normal vector of the face is positive, this face faces the line of sight; otherwise it faces away from the line of sight. The contour of the model consists of the shared edges of visible and invisible surfaces.

The contour obtained here is projected onto the selected projection plane. The projection contour is normalized to obtain its centroid and principal axis. Figure 5 shows the feature lines extracted from a 3D original model by using the above steps, where (a) shows the original 3D model and (b) shows the projected contour and the principal axis of the model.

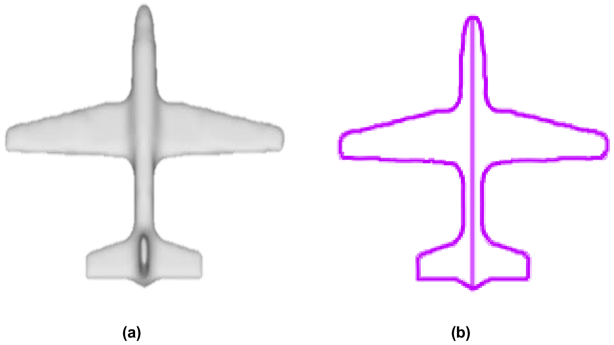


**FIGURE 5. Outer contour extracted from the 3D model (a) The 3D original model; (b) contour projection of this model.**

As shown in Fig. 5, the proposed model editing method defines the contour projection of the region as the editable source line.

### C. CONSTRAINT FOR MODEL EDITING

The model editing method proposed in this paper firstly defines the target reference line in the image or the sketch, and then defines the editable source line of the interested region in the model. The constraint condition for model editing is then established by comparing the above two curves.

The first step of this method is to build up the set $T$ of feature points on the target reference line [29]. Then, based on the length proportion between the curves, the points of the editable source line corresponding to all the points in the set $T$ are identified one by one and saved to the point set $T'$. Suppose the full length of the target reference line is $H$ and the full length of the editable source line is $H'$.

Let point $A$ be the intersection point between the principal axis and the upper segment of the target reference line. We select a point $\mathrm{t_i}$ from the set $T$. The length from $A$ to $\mathrm{t_i}$ towards the clockwise direction on the target reference line is denoted by $\mathrm{d}(\mathrm{t_i})$. Assume that point $A'$ is the intersection point between the principal axis and upper segment of the editable source line (on the original model).

$\mathrm{d}\left(\mathrm{t_i}'\right)$ denotes the length from $A'$ to a point $\mathrm{t_i}'$ towards the clockwise direction on the editable source line. $\mathrm{d}\left(\mathrm{t_i}'\right)$ can be estimated by Equation (2) based on the principle that the length is proportional, namely, the ratio of $\mathrm{d}(\mathrm{t_i})$ to length H is the same as the ratio of $\mathrm{d}\left(\mathrm{t_i}'\right)$ to length H′. With the value of $\mathrm{d}\left(\mathrm{t_i}'\right)$, the position of point $\mathrm{t_i}'$ can be determined and saved to the set $T'$.

$$\underset{t_i'}{\operatorname{argmin}} \left\| \frac{d\,(t_i)}{H} - \frac{d\,(t_i')}{H'} \right\|_2^2 \tag{2}$$

The above steps define the point $\mathrm{t_i}'$ on the editable line (saved in the set $T'$), which corresponds to the feature point $\mathrm{t_i}$ in the set $T$ of the target reference line. This correspondence is part of the constraint condition for the subsequent model editing. The correspondence between the two sets is as shown in Fig. 6, where Fig. 6a shows the set $T$ of feature points obtained on the target reference line, and Fig. 6b shows the set $T'$ of corresponding points on the editable source. Let point $B$ (or $B'$) be the intersection point of the principal axis of the target reference line (or the editable source line) and the lower segment of the target reference line (or the editable source line). $C$ is a point (assume it is $\mathrm{t_i}$) in the set $T$. Points $A$, $B$ and $C$ form a triangle $\Delta ABC$. We then find such a point $C''$ for the source model that the triangle $\Delta A''B''C''$ is similar to $\Delta ABC$. $C''$ is saved to the set $T''$ as the i-th vertex $\mathrm{t_i}''$ in set $T''$. So $\mathrm{t_i}''$ in the set $T''$ corresponds to $\mathrm{t_i}$ in the set $T$.

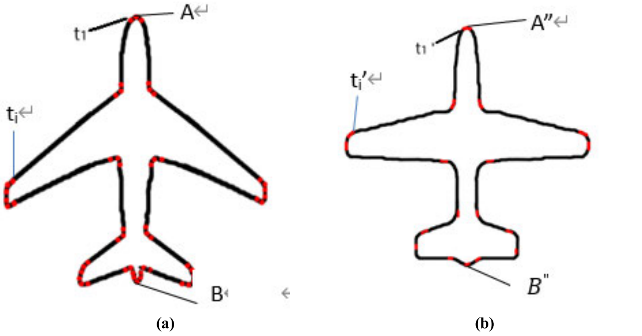


**FIGURE 6. Deformation constraint condition for the target and editable source lines a. Set T of feature points of the target reference line b. Set *T′* of corresponding points of the editable source line.**

Now we introduce three sets of points: i) $T$: a set of feature points on the outer contour of the object in the selected 2D image (i.e., target reference line); ii) $T'$: a set of points on the contour of the 3D model (i.e., the source reference line), each of which corresponds to a point in the set $T$ according to the length proportion principle; iii) $T''$: a set of points found on the 2D projection of the input 3D model. Each point in $T''$corresponds to a point in $T$ based on the principle of similar triangles.

The aim of the mesh deformation (i.e., model editing) is to move the point $t_i'$ in the set $T'$to $t_i''$ in the set $T''$. Namely, $t_i'$ is the control point for mesh deformation while $t_i''$ is the target position that $t_i'$ should be moved to. In addition to moving the points in T, all points in the mesh of the 3D model are moved to the appropriate positions based on the deformation rules presented in subsection III.D. The points in the sets $T'$ and $T''$ form the constraint of other points' movement (i.e., the constraint of model editing).

Figure 7 illustrates how to use the principle of similar triangles to determine $C''$ based on a point $C$ on the target reference line. Since triangles $\Delta ABC$ and $\Delta A''B''C''$ are similar, the corresponding angles of the two triangles are identical, and the length of the corresponding sides of the two triangles are proportional.

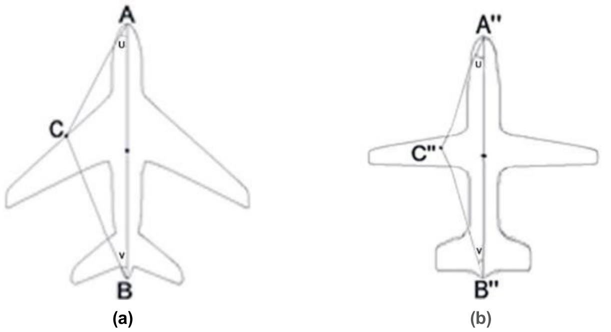


**FIGURE 7. Correspondence between similar triangles on two contours (a) Triangle $\Delta ABC$ on the target reference line (b) Similar triangle $\Delta A''B''C''$ on the editable source line.**

The position of the point $C''$ identified by the principle of similar triangles shows not only the change in the distance of the vertices (points) in the set $T'$ as in the previous research, but also the change in the direction of these vertices.

When the positions of three points of a triangle $\Delta ABC$ are known, the angles $U$ and $V$ of the triangle (shown in Fig. 7a) can be determined by the cosine theorem. Assume the lengths of the side $BC$, $AC$ and $AB$ are $a$, $b$ and $c$, respectively. Angles $U$ and $V$ can be calculated by Equations (3) and (4), respectively:

$$\mathrm{U} = \cos^{-1}\left(\frac{\mathrm{b}^2 + \mathrm{c}^2 - \mathrm{a}^2}{2\mathrm{bc}}\right) \tag{3}$$

$$\mathrm{V} = \cos^{-1}\left(\frac{\mathrm{c}^2 + \mathrm{a}^2 - \mathrm{b}^2}{2\mathrm{ca}}\right) \tag{4}$$

If $\Delta A''B''C''$ is similar to $\Delta ABC$, they will have the same arrangement relationship. Namely, when the three vertices of $\Delta ABC$ are arranged clockwise, the three vertices of $\Delta A''B''C''$ are also arranged clockwise. The Cartesian coordinates of points $A''$ and $B''$ are $(\mathrm{x''}_\mathrm{A}, \mathrm{y''}_A)$ and $(\mathrm{x''}_\mathrm{B}, \mathrm{y''}_\mathrm{B})$, respectively. The Cartesian coordinates of points $A$, $B$ and $C$ are $(\mathrm{x}_\mathrm{A}, \mathrm{y}_A)$, $(\mathrm{x}_B, \mathrm{y}_B)$ and $(\mathrm{x}_C, \mathrm{y}_C)$, respectively. After angles $U$ and $V$ are calculated by Equations (3) and (4), Equations (5), (6) and (7) can be applied to determine the coordinates $(\mathrm{x''}_\mathrm{C}, \mathrm{y''}_C)$ of point $C''$,

$$sign(x) := \begin{cases} -1, & if\ x < 0 \\ 1, & if\ x > 0 \end{cases} \tag{5}$$

where $x = AB \times AC = |\mathrm{x}_B - \mathrm{x}_A, \mathrm{y}_B - \mathrm{y}_A| \times |\mathrm{x}_C - \mathrm{x}_A, \mathrm{y}_C - \mathrm{y}_A|$. Note that if A, B and C are in an anti-clockwise order in terms of their coordinates, then $x > 0$ If they are clockwise, $x < 0$.

$$\mathrm{x''}_\mathrm{C} = \frac{\mathrm{x''}_\mathrm{A} \cot \mathrm{V} + \mathrm{x''}_\mathrm{B} \cot \mathrm{U} + \left(\mathrm{y''}_A - \mathrm{y''}_\mathrm{B}\right) sign(x)}{\cot \mathrm{U} + \cot \mathrm{V}} \tag{6}$$

$$\mathrm{y''}_\mathrm{C} = \frac{\mathrm{y''}_A \cot \mathrm{V} + \mathrm{y''}_\mathrm{B} \cot \mathrm{U} + \left(\mathrm{x''}_\mathrm{B} - \mathrm{x''}_\mathrm{A}\right) sign(x)}{\cot \mathrm{U} + \cot \mathrm{V}} \tag{7}$$

The existing methods in literature [25], [26] use the length proportion principle to locate the deformed position of the points in the input 3D model. Our similar triangle-based method can capture not only the distance similarity, but also the angle similarity between the generated model and the target object in the selected 2D image. The similarity between the projected contour of the generated 3D model and that of the target object (in the selected 2D image) can be quantified by the metric of Laplacian distance between the two curves defined in [30]. We have conducted the experiments using the similarity metric (presented in Section IV); the results verify that the 3D models generated based on the principle of similar triangles are more similar to the target 2D objects than those generated by the existing editing method. Moreover, since our method can exploit more info (angle), our method is faster than the existing methods too. The experimental results regarding this are also presented in Section IV.

### D. DEFORMATION

The model editing method proposed in this paper falls into the methodology of As-Rigid-As-Possible (ARAP) mesh deformation [21]. The as-rigid-as-possible mesh deformation improves the Laplacian differential mesh deformation by introducing a parameter $R_i$ into the energy function of the Laplacian differential mesh deformation. Local rigidity is the basic principle of mesh deformation. The as-rigid-as-possible mesh deformation generates the models with different appearances by using the position change of control points defined on the mesh to generate the position change of other vertices of the mesh.

The proposed model editing method first uses the points, which are 2D points, in the set $T'$ to find the corresponding 3D vertices in the original model, which are used as a set of control points in the model editing. Because all the points in the set $T'$ come from the plane projection of the model contour, each feature point $t_i'$ contained in the set $T'$ corresponds to a point in the model contour, that is, a vertex in the model. According to the coordinates of $t_i'$, we find the corresponding

3D vertices on the original model as the control points of the later model deformation. These 3D points are saved in the set $\mathrm{D}^{\mathrm{s}}$ (the superscript "s" represents source). In the method presented in section III.C, we have obtained the set $T''$. Similarly, we use the 2D points in $T''$ to find the corresponding 3D points in the space of the 3D model, which are saved in the set $\mathrm{D}^{\mathrm{d}}$ ("d" means deformation). Note that the depth of the vertices contained in set $\mathrm{D}^{\mathrm{d}}$ is the same as the depth of the corresponding points in set $\mathrm{D}^{\mathrm{s}}$. The points in the set $\mathrm{D}^{\mathrm{d}}$ are the points that the corresponding points in the set $\mathrm{D}^{\mathrm{s}}$ should be deformed into. The position changes from the points in $\mathrm{D}^{\mathrm{s}}$ to the corresponding point in $\mathrm{D}^{\mathrm{d}}$ are used to control the mesh deformation of an interested region $P$ on the input 3D model.

The topology of $P$, which is a triangle mesh, is determined by $n$ vertices $\{v_i\}$ and a set of edges $\{e_{ij}\}$ between the vertices. Assume $P$ is deformed into $P'$. The deformed mesh $\mathrm{P}'$ is defined by $\{v_i{}'\}$. We denote the set of vertices connected to vertex $v_i$ by $N(i)$, which we call the one-ring neighbors of $v_i$. Each vertex $v_i$ and its one-ring neighborhood form a unit, the ARAP energy of the unit is generated by the non-rigid transformation between $v_i$ and $v_i{}'$ in the deformation.

The deformed energy function $\mathrm{E}\left(P, P'\right)$ of the entire mesh P is obtained by summing up the ARAP energy of each vertex unit, which is formulated as Equation (8), where $n$ is the number of vertices contained in $P$, $\mathrm{v_i}$ is a vertex in the region $P$, $N_i$ is a set of adjacent vertices of $v_i$, $v_i{}'$ is the vertex position after deformation, and $R_i$ is a rotation matrix that describes the local rotation of each vertex and its one-ring neighborhood. $\omega_{ij}$ is a cotangent weight, which prevents the surface from discrete deviation. $\omega_{ij}$ can be calculated by Equation (9). In Equation (9), $\alpha_{ij}$ and $\beta_{ij}$ are the angles opposite of the mesh edge $e_{ij}$, where $e_{ij}$ is determined by Equation (10).

$$E\left(P, P'\right) = \sum\nolimits_{i=1}^{n} \sum\nolimits_{j\in N_i} \omega_{ij} \left\| (v_i{}' - v_j{}') - R_i(v_i - v_j) \right\|^2 \quad (8)$$

$$\omega_{ij} = \frac{1}{2}\left(\cot\alpha_{ij} + \cot\beta_{ij}\right) \quad (9)$$

$$e_{ij} = v_i - v_j \quad (10)$$

We aim to find such positions $\{v_i{}'\}$ of $\mathrm{P}'$ that minimize $\mathrm{E}\left(P, P'\right)$ in (8), subject to the deformation constraints defined in $\mathrm{D}^{\mathrm{s}}$ and $\mathrm{D}^{\mathrm{d}}$. Since $v_j$ (or $v_j{}'$) is the one-ring neighbour of $v_i$ (or $v_i{}'$), the local features of the surface can be preserved as much as possible after the deformation.

However, since the energy function $\mathrm{E}\left(P, P'\right)$ is obtained by summing up the energy of each individual vertex unit (i.e., a vertex and its one-ring neighborhood), we find that the deformation can be trapped into the local optimum, i.e., reducing the energy of each individual vertex unit as much as possible, which may generate the distortion in the deformed model. We show this phenomenon in our experiments presented in Section IV.

In order to overcome this problem, we introduce a global energy function in this work, denoted by $E_c\left(P, P'\right)$, to measure the energy difference of the entire region $P$ before and after the deformation of the mesh in the model editing process, which acts as another (soft) constraint to control the structural stability of the model.

$E_c\left(P, P'\right)$ is calculated by Equation (11), where $v_m$ and $v_m{}'$ are the mean values of the coordinates before and after the deformation of the region $P$. $v_m = \frac{1}{n}\sum_{i=1}^{n} v_i$, and $v_m{}' = \frac{1}{n}\sum_{i=1}^{n} v_i{}'$. The stability of the whole model structure is ensured by minimizing the value of $E_c$.

$$E_c\left(P, P'\right) = \left\| v_m - v_m{}' \right\|^2 \quad (11)$$

In this paper, we combines Equations (8) and (11) and propose an energy optimization framework as in (12) to transform the mesh deformation into a quadratic minimization problem:

$$E_{IARAP}\left(P, P'\right) = \mathrm{E}\left(P, P'\right) + \gamma E_c\left(P, P'\right) \quad (12)$$

In Equation (12), the coefficient $\gamma$ is used to balance the influence of the global and the local constraints in the mesh deformation. We have carried out the experiments to investigate the impact of $\gamma$, and identify the optimal empirical value of $\gamma$ that can generate the best deformation result. The experimental results are presented in section IV.A. We also carried out the experiments to evaluate the effectiveness of incorporating the global energy function defined in (11). The experimental results are presented in Section IV.C.

Solving the unknown vertex $V' = \left\{\mathrm{v_i}{}'\right\}$ in Equations (11) and (12) is a non-linear optimization problem for solving unknown $\{\mathrm{R_i}\}$ and $\{\mathrm{v_i}{}'\}$. Inspired by the method proposed by Sorkine O [21], we adopt an iterative flip-flop optimization in this paper to solve the combined energy function $E_{IARAP}\left(P, P'\right)$. First, the vertices of mesh remain fixed and the unknown rotation matrix $\{\mathrm{R_i}\}$ is solved using Singular Value Decomposition (SVD).

The covariance matrix $\mathrm{S_i}$ is established as follows.

$$\mathrm{S_i} = \sum\nolimits_{\mathrm{j\in N(i)}} \omega_{\mathrm{ij}} \mathrm{e_{ij}} \mathrm{e_{ij}}{}'^{\mathrm{T}} = \mathrm{P_i D_i P_i}{}'^{\mathrm{T}} \quad (13)$$

where $D_i$ is a diagonal matrix containing the cotangent weight $\omega_{\mathrm{ij}}$, $P_i$ and $P_i{}'$ are $3 \times |\mathrm{N}(\mathrm{v_i})|$ matrices with edges $e_{ij}$ and $e_{ij}{}'$ as their columns, respectively. Using the singular value decomposition of $\mathrm{S_i} = \mathrm{U_i \Sigma_i V_i^T}$, $R_i$ can be obtained by Equation (14).

$$\mathrm{R_i} = \mathrm{V_i U_i^T} \quad (14)$$

We find positions $\mathrm{V}'$ that minimize $E_{IARAP}\left(P, P'\right)$ for a fixed set of rigid transformations. Then for the previously solved $V'$, we continue to obtain the rotation matrices $\{\mathrm{R_i}\}$, as described in Equations (13) and (14). We repeat the process of solving $\{\mathrm{R}\}$ and $\mathrm{V}'$ iteratively until the minimum of $E_{IARAP}\left(P, P'\right)$ is reached.

When the rotation matrices $R_i$ are fixed, the energy function $E_{IARAP}\left(P, P'\right)$ is a quadratic equation of $\left\{v_i{}'\right\}$. Therefore, in order to obtain the minimum value of the energy function $E_{IARAP}\left(P, P'\right)$, we set the partial derivatives of $E_{IARAP}\left(P, P'\right)$ to zero w.r.t. each $v_i{}'$ i.e., $\frac{\partial E_{IARAP}}{\partial \mathrm{v_i}{}'} = 0$. This way, we build a set of linear equations to solve the optimal grid points ($\omega_{ij} = \omega_{ji}$).

The partial derivative of $E_{IARAP}\left(P, P'\right)$ can be derived as follows.

$$\begin{aligned}
&\frac{\partial E_{IARAP}}{\partial {v_i}'} \\
&= \frac{\partial}{\partial {v_i}'}\left(\mathrm{E}\left(P, P'\right) + \gamma E_c\left(P, P'\right)\right) \\
&= \frac{\partial}{\partial {v_i}'}\left(\sum_{i=1}^{n}\sum_{j\in N_i}\omega_{ij}\left\|({v_i}'-{v_j}')-R_i(v_i-v_j)\right\|^2+\gamma\left\|v_m-{v_m}'\right\|^2\right) \\
&= \sum_{j\in N(i)} 2\omega_{ij}\left(\left({v_i}'-{v_j}'\right)-R_i\left(v_i-v_j\right)\right) \\
&\quad - \sum_{j\in N(i)} 2\omega_{ji}\left(\left({v_j}'-{v_i}'\right)-R_j\left(v_j-v_i\right)\right) - \frac{2\gamma}{n^2}\left(v_i-{v_i}'\right) \\
&= \sum_{j\in N(i)} 2\omega_{ij}\left(\left({v_i}'-{v_j}'\right)-R_i\left(v_i-v_j\right)\right) \\
&\quad + \sum_{j\in N(i)} 2\omega_{ij}\left(\left({v_i}'-{v_j}'\right)-R_j\left(v_i-v_j\right)\right) + \frac{2\gamma}{n^2}\left({v_i}'-v_i\right) \\
&= 4\omega_{ij}\sum_{j\in N(i)}\left({v_i}'-{v_j}'\right) - 2\omega_{ij}\sum_{j\in N(i)}\left(R_i+R_j\right)\left(v_i-v_j\right) \\
&\quad + \frac{2\gamma}{n^2}{v_i}' - \frac{2\gamma}{n^2}v_i
\end{aligned}$$

Since $\frac{\partial E_{IARAP}}{\partial {v_i}'}$=0, we can derive the following sparse linear equation set:

$$\begin{aligned}
&\omega_{ij}\sum_{j\in N(i)}\left({v_i}'-{v_j}'\right)+\frac{\gamma}{2n^2}{v_i}' \\
&\qquad = \frac{\omega_{ij}}{2}\sum\nolimits_{j\in N(i)}\left(R_i+R_j\right)\left(v_i-v_j\right)+\frac{\gamma}{2n^2}v_i \qquad (15)
\end{aligned}$$

Equation (15) can be simplified to Equation (16), which can then be solved to obtain the mesh vertex set $\left\{{v_i}'\right\}$ such that the energy function $E_{IARAP}\left(P, P'\right)$ is minimized.

$$\mathrm{L}V' + \frac{\gamma}{\mathbf{2n^2}}IV' = b + \frac{\gamma}{\mathbf{2n^2}}IV \qquad (16)$$

where $L$ is the discrete Laplace-Beltrami operator applied to $V'$; $I$ is an $n$-order unit matrix; $b$ is an $n$-vector whose i-th row is $\frac{\omega_{ij}}{2}\sum_{j\in N(i)}\left(R_i+R_j\right)\left(v_i-v_j\right)$.

According to the method described above, we iteratively solve $R$ and $V$ using Equations (14) and (16), and finally obtain the mesh vertices that minimize the deformation energy in Equation (12). For non-rotationally symmetric models, such as aircraft or dolls, deformation can be performed directly at a specified view. For rotationally symmetric models such as vases or cups, the contour of the model is projected and deformed with multiple views. In our experiment, two adjacent views have the difference of 90 degrees. An object such as mug and teapot in Fig. 6 can be segmented first so that the handle and the main body are separated using the method in [31]. Then the main body is independently deformed using the method presented above. Finally, the deformed body of the object and other parts are fused to obtain the desired model using the method in [32].

## IV. EXPERIMENTS

In this section, we first conduct the experiments to investigate the impact of the weight $\gamma$ in our energy function defined in Equation (12), which consequently identifies the empirical value of $\gamma$ that generates the best deformation result. Next, we evaluate the model editing method proposed in this paper, and compare it with the existing methods in literature [25], [26]. Moreover, we use the assessment function based on the 2D Laplacian operator to compare the similarity between the edited model and the guide object in the selected image, which can validate the effectiveness of using our principle of similar triangles to locate the deformed positions of the feature points in the input 3D model.

We carried out all experiments using a graphic workstation with Intel Core i5-7300 2.50 GHz CPU, 8-GB RAM, and NVIDIA GeForce GTX 1050 Graphics Cards. The C++-based OpenGL and CGAL libraries are installed in the graphic workstation to implement the proposed 3D model editing method. The matrix was solved using the sparse Cholesky solver in the TAUCS library [33] and the standard SVD implementation  [34].

The models in the experiments are from the classification grid database of SHREC12 [35] and the Princeton University dataset [36].

### A. EMPIRICAL STUDIES OF THE PARAMETER $\gamma$

In Equation (12), the parameter $\gamma$ is used as the weight to balance the effect of local and global constraints. We conducted the empirical studies to determine the value of $\gamma$.

The work in [30] presents a distance metric $E_a$ to measure the similarity between two curves. The lower value of $E_a$ indicates that two curves are more similar. In this work, the two curves are the projection of the contour of the edited model P′ and the contour of the target object O (2D image). The metric $E_a$ is calculated by Equation (17), where $\mathrm{L_p}\left(\mathrm{v_i}\right)$ is the Laplacian coordinate of point $\mathrm{v_i}$ on a 3D mesh, T is the set of vertices on the contour of the target object O, and T″ is the set of vertices in the projection of the contour of the edited model $P'$. $\mathrm{L_p}\left(\mathrm{v_i}\right)$ is calculated by Equation (18), where $\mathrm{v_{i-1}}, \mathrm{v_{i+1}}$ are the adjacent points to $\mathrm{v_i}$ on the curve. Essentially, the metric $E_a$ calculates the distance in the Laplacian coordinates between two curves.

$$\mathrm{E_a} = \sum\nolimits_{\mathrm{v_i}\in \mathrm{T}, \mathrm{v_i}''\in T''}\left\|\mathrm{L_p}\left(\mathrm{v_i}\right)-\mathrm{L_p}\left(\mathrm{v_i}''\right)\right\|^2 \qquad (17)$$

$$\mathrm{L_p}\left(\mathrm{v_i}\right) = \mathrm{v_i} - \left(\mathrm{v_{i-1}}+\mathrm{v_{i+1}}\right)/2 \qquad (18)$$

In the experiments, we set different values of $\gamma$ in our 3D editing method. After the editing is completed, we calculate the value of $E_a$. For a lesser value of $E_a$, there is a higher level of similarity between the edited model and the target 2D image. The experimental results are presented in Table 2. It can be seen from this table that when $\gamma$ is very small such as 0.05, which means that the impact of local constraint (i.e., the vertex unit consisting of a vertex and its one-ring neighborhood) dominates in our energy function defined in equation (12), $E_a$ is relatively big. It suggests that

**TABLE 2.** Comparison of deformation similarity $E_a$ with different values of $\gamma$.

| Model | $E_a$ | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | $\gamma = 0.05$ | $\gamma = 0.1$ | $\gamma = 0.15$ | $\gamma = 0.2$ | $\gamma = 0.25$ | $\gamma = 0.3$ | $\gamma = 0.35$ | $\gamma = 0.4$ |
| air plane | 0.083 | 0.061 | 0.035 | 0.051 | 0.076 | 0.093 | 0.118 | 0.135 |
| bear | 0.143 | 0.133 | 0.12 | 0.123 | 0.125 | 0.129 | 0.131 | 0.131 |
| cup | 0.172 | 0.169 | 0.167 | 0.169 | 0.170 | 0.173 | 0.183 | 0.187 |
| teapot | 0.036 | 0.027 | 0.021 | 0.045 | 0.063 | 0.072 | 0.08 | 0.087 |
| wine glass | 0.049 | 0.032 | 0.015 | 0.037 | 0.054 | 0.061 | 0.078 | 0.086 |

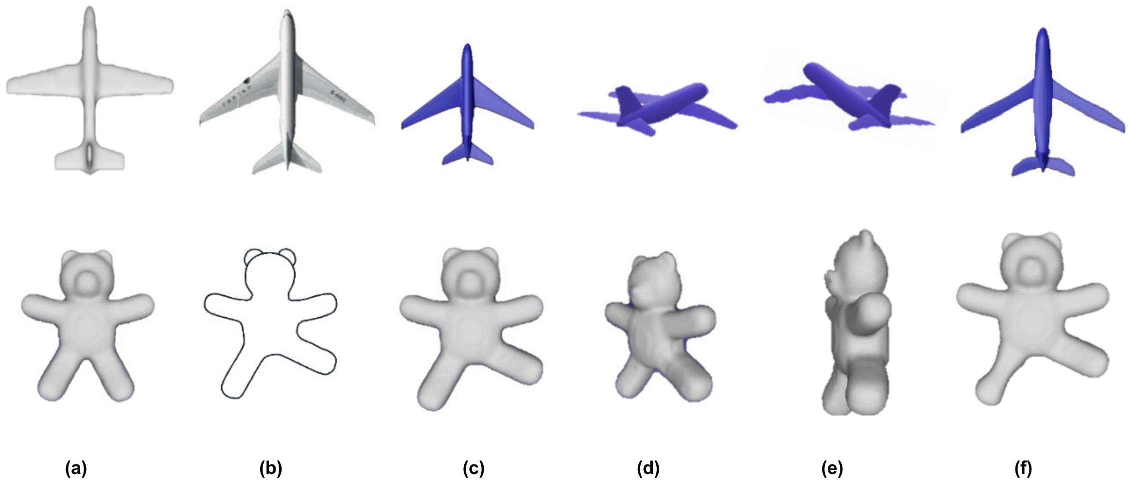


**FIGURE 8.** The editing results for non-rotationally symmetric models; a) Original 3D models; b) Input 2D images; c) Results of deformation; d) results from different perspectives; e) the models edited by the method in [26]; f) the models edited by the method in [25].

only considering the local constraint does not obtain the best deformation effect. As $\gamma$ increases, $E_a$ decreases. $E_a$ reaches the minimal value when $\gamma$ is 0.15. As $\gamma$ increases further, $E_a$ increases again. Therefore, 0.15 can be regarded as the best empirical value for $\gamma$. In the following experiments in this section, we set $\gamma = 0.15$ to achieve the best deformation effect.

## B. THE EFFECTIVENESS OF OUR 3D EDITING METHOD

The model editing method proposed in this paper has been tested by model designers with different level of proficiency. They selected an original 3D model from the model database and defined the region of interest in the model, and then selected an image or hand-drawn sketch. Then, the proposed method is applied to edit the model. Various types of models including airplanes, dolls, mugs, vases, etc are edited. The experimental results are shown in Fig. 8 and Fig. 9. Column (a) in Fig. 8 and Fig. 9 show the original 3D models selected by users, and column (b) show images the selected by users to guide model editing. Note that the second row of column (b) in Fig. 8 is the hand-drawn sketches, while others in column (b) are the selected 2D images. Column (c) shows the deformed models obtained by our method from the original models in column (a), guided by the images in column (b). Column (d) shows the deformed models observed from different perspectives. Column (e) and (f) shows the deformed models obtained by the methods in [26] and [25], respectively. It can be seen from Column (e) that the models edited by the method in [26] produce some noticeable distortions. For example, the nose bridge of the bear in Fig. 8 collapses and the rim of the cup in Fig. 9 curls. The models edited by the method in [25] contain some distortions too. For example, the wings of the airplane in Fig. 8f, the teapot and the cup in Fig. 9f are obviously out of shape. In contrast, it can be seen from Fig. 8 and Fig. 9 that no matter a 3D model is rotationally symmetric or not, our method generates the satisfactory model editing result compared with other methods. In addition, as shown in the second and third rows of Fig. 9, our method can also obtain the satisfactory results for complex models that need to be decomposed into simpler parts first as discussed at the end of Section III.D.

In addition to the visual effect of the edited models observed in Fig. 8 and Fig. 9, we also calculated the value of the distance metric $E_a$ defined in Equation (16) to quantify the effectiveness of our method. Table 3 compares the values of $E_a$ between our method and the methods in [25] and [26].

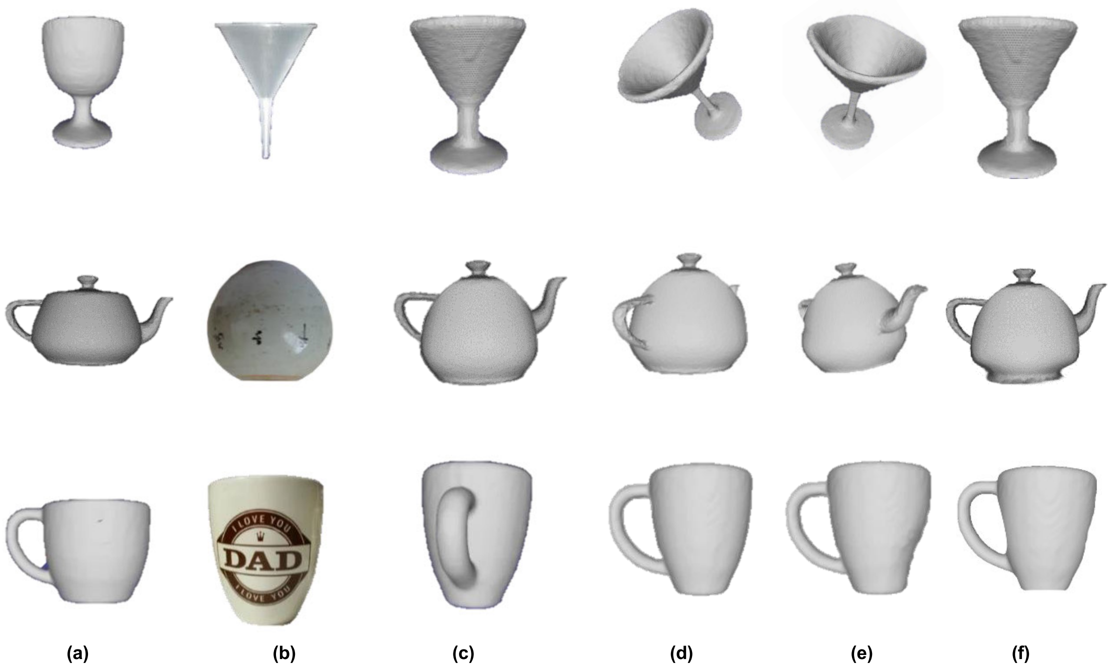


**FIGURE 9.** The editing results for rotationally symmetric models; a) Original 3D models; b) Input 2D images; c) Results of deformation; d) results from different perspectives; e) results from literature [26]; f) results from literature [25].

**TABLE 3.** The similarity between the contours of the deformed model and the target object.

| Model | Points | $E_a$ | $E_a$ in [25] | $E_a$ in [26] |
|---|---|---|---|---|
| airplane | 6701 | 0.035 | 0.189 | 0.245 |
| bear | 12561 | 0.12 | 0.193 | 0.206 |
| wine glass | 1401 | 0.167 | 0.171 | 0.195 |
| cup | 15198 | 0.021 | 0.098 | 0.12 |
| teapot | 13434 | 0.015 | 0.082 | 0.093 |

It can be observed from the table that the value of $E_a$ achieved by our method is much lower than that by the methods in [25] and [26] for all models. This result indicates that our method can generate a model which is much more similar to the target 2D image than the methods in literatures [25] and [26]. The reason is because our similar triangle-based method captures not only the distance similarity, but also the angle similarity between the selected 2D image and the edited model.

In addition, the method proposed in this paper is compared with a piece of industrial software called Magic3D. Magic3D uses a different way of generating 3D models. It does not use the 2D image or sketches to guide the deformation, but directly establishes the deformation constraints by manually selecting control points on the 3D model and locating the control point target position. Therefore, Magic3D requires the users to have good experience in establishing the deformation constraints. The comparison results are shown in Figure 10. In Fig.10, Column (a) shows the original 3D models for editing; column (b) shows the images selected by the user to guide the model editing; column (c) shows the deformation model obtained by our method under the guidance of the image in column (b); column (d) shows the deformation results obtained by a user who has not mastered how to set the deformation constraints yet when using Magic3D; column (e) shows the deformation results obtained by a skilled 3D model designer using Magic3D. Comparing columns (d) and (e), we can see that the professional level of the model designer directly affects the quality of deformation results generated by Magic3D. When an unskilled user uses Magic3D to edit 3D models, the deformation results may be greatly distorted. For example, the limbs of the edited models in column (d) are severely distorted. Compared with Magic3D, our method uses 2D images to generate deformation constraints automatically. With our method, it is much easier for the inexperienced users to obtain satisfactory model editing results.

### C. EFFECTIVENESS OF GLOBAL ENERGY FUNCTION

In this section, we conduct the experiments to evaluate the effectiveness of incorporating the global energy function defined in Equation (12). The top row of Fig. 11 is the deformation models using our energy equation $E_{IARAP}$, which

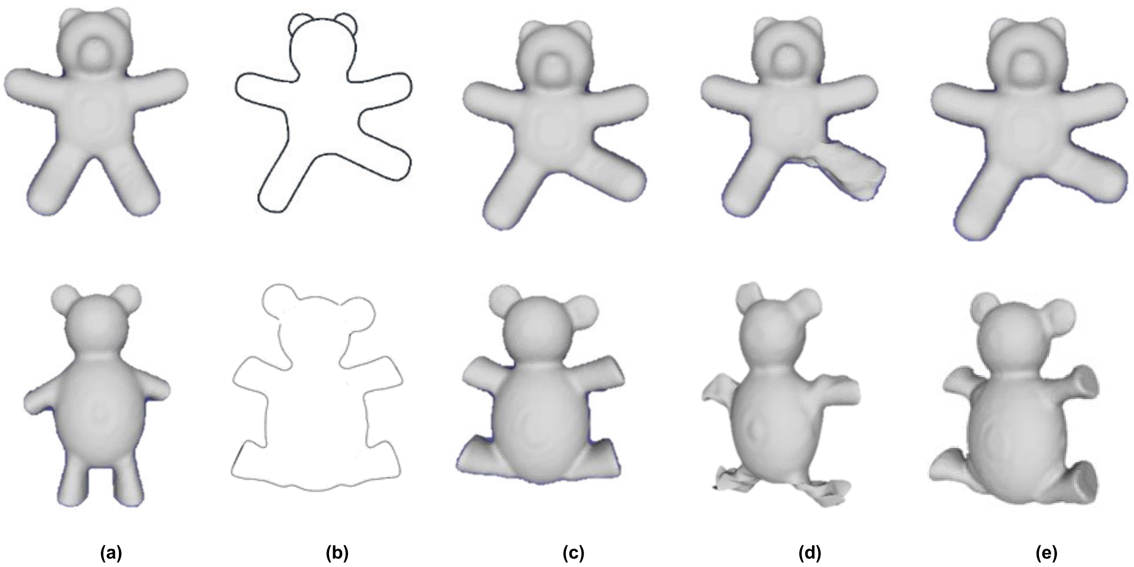


**FIGURE 10.** Comparing our method with Magic3D; a) Original 3D models; b) Input 2D images; c) Deformation results of our method; d) Deformation results of Magic3D used by an inexperienced user; e) Deformation results of Magic3D used by a skilled user.

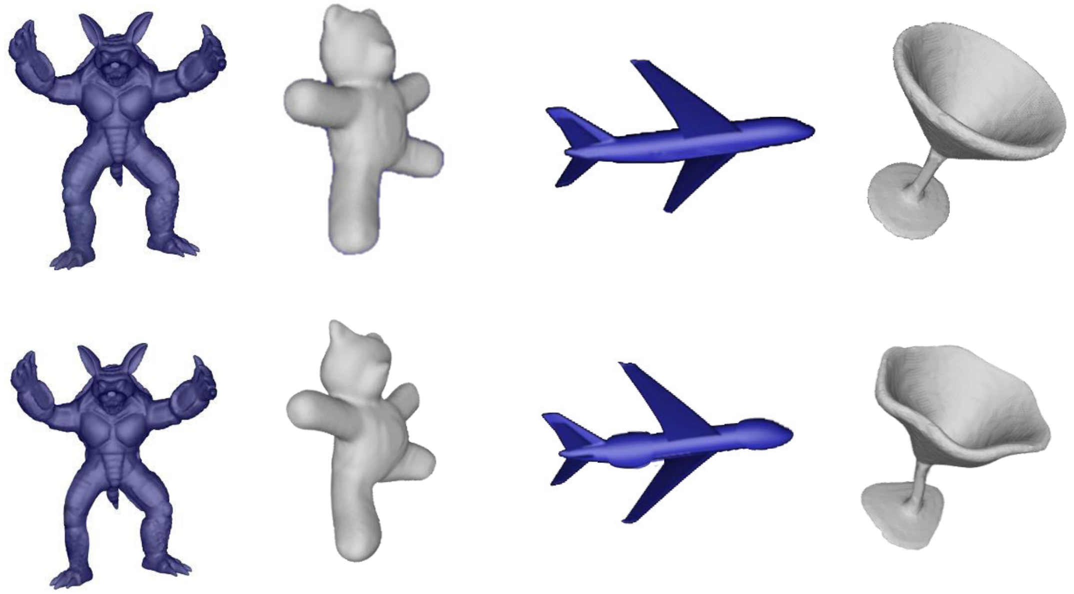


**FIGURE 11.** The deformation effect with and without the global energy functions. The models in the top row are the deformation results using the energy function containing $E_C$. The models in the bottom row are the deformation results using the original energy equation [21] without containing $E_C$.

incorporates the global energy function. The bottom row is the deformation models only using the energy function $E\left(P, P'\right)$ that only considers the local constraints [21]. It can be seen from the figure that our optimized energy equation can generate the deformed model with less distortion. For example, see the outstretched right leg of the model in the first column of Fig. 11. In general, we can see from the second row of Fig. 11 that the vertex distribution of these deformation results is uneven and there exists poor distortion. However, the deformation in the first row, generated by incorporating the global energy function, makes the vertex distribution more uniform and the distortion is not noticeable.

### D. EFFICIENCY OF OUR 3D EDITING METHOD

In this subsection, we conduct the experiments to compare the running time of our method with the existing methods in literature [25] and [26]. The results are listed in Table 4. It can be seen from this table that our editing method is much faster than other methods in Table 4. The reason why our 3D editing method is faster is because the method in [26] has to run the process of comparing the outlines and cross-sections of 3D shapes multiple times in order to determine the starting position and the target position of the feature point. The method in [25] has to model a 2D illustration as a planar mesh and represent the shape with four components: the object contour,

**TABLE 4. Model editing time(second).**

| Model | Faces/ Points | Total time of our method | Total time of the method in [25] | Total time of the method in [26] |
|---|---|---|---|---|
| air plane | 13398/6701 | 6.42 | 20.35 | 15.46 |
| bear | 25118/12561 | 7.93 | 21.56 | 13.87 |
| cup | 30396/15198 | 4.84 | 16.88 | 10.65 |
| teapot | 26864/13434 | 3.99 | 15.73 | 11.03 |
| wine glass | 2750/1401 | 3.36 | 13.26 | 9.86 |

the context curves, user-specified features and local shape details, then establish the vertex correspondence between the input model and the 2D illustration, and finally formulate the shape deformation as a style-constrained differential mesh editing problem. However, our method only compares the model contour and the image contour once to determine the starting position (the set $T'$) and the target position (the set $T''$) of the control points. The differential deformation can then be carried out. As the result, our method can run faster while generating more realistic deformation effect as shown in Fig. 8, Fig. 9, Fig. 10 and Table 4.

The running time for processing the models is shown in Table 4.

## V. CONCLUSION

3D model editing technology plays a significant role in the development of computer vision. With the advance of computer vision technology, the demand for 3D models has been growing rapidly. There is much need to obtain high-quality 3D models in a quick and efficient way.

This paper proposes a model editing method, which makes use of the widely available 2D image data to guide the model designers to edit the existing 3D models quickly and effectively, and generate more 3D models with new appearances. This method is based on the as-rigid-as-possible deformation method and preserves the local features of the model surface in the editing process. This paper also proposes an energy function to prevent undesired local deformations in the editing process. We have conducted extensive experiments; the results show that compared with the methods in the literature our method can achieve better deformation results with higher efficiency.

The focus of our future research is on rectifying the defects of the existing model editing method. First the user has to manually select the view, which limits the applicability of the method in this paper. We plan to improve the method by enabling automatic view selection. In addition, the proposed method obtains the target position of the control point through the comparison with the 2D contour, and directly uses the depth of the control point defined on the original 3D model contour to restore the target position of the control point from 2D coordinates to 3D coordinates. By doing so, the actual depth of the points on the deformed model may differ from the ideal depth of points on the object in the 2D image. In future, we plan to use the deep learning technique to obtain the ideal depth of points on the object based on 2D images, and use it to restore the target position depth of control points, based on which a better deformed 3D model can be generated.

## REFERENCES

[1] L. Gao, Y.-K. Lai, J. Yang, Z. Ling-Xiao, S. Xia, and L. Kobbelt, "Sparse data driven mesh deformation," *IEEE Trans. Vis. Comput. Graph.*, Sep. 16, 2019, early access, doi: 10.1109/TVCG.2019.2941200.

[2] I. K. Kazmi, L. You, X. Yang, X. Jin, and J. J. Zhang, "Efficient sketch-based creation of detailed character models through data-driven mesh deformations," *Comput. Animation Virtual Worlds*, vol. 26, nos. 3–4, pp. 469–481, May 2015.

[3] J. Zimmermann, A. Nealen, and M. Alexa, "SilSketch: Automated sketch-based editing of surface meshes," in *Proc. 4th Eurographics Workshop Sketch-Based Interfaces Modeling (SBIM)*, Riverside, CA, USA, 2007, pp. 23–30.

[4] L. Gao, Y.-K. Lai, D. Liang, S.-Y. Chen, and S. Xia, "Efficient and flexible deformation representation for data-driven surface modeling," *ACM Trans. Graph.*, vol. 35, no. 5, pp. 1–17, Sep. 2016.

[5] Y. Hu, X. Guo, B. Zhao, S. Lin, and X. Luo, "3D model editing from contour drawings on orthographic projection views," in *Image and Signal Processing*. Cham, Switzerland: Springer, 2014, pp. 612–619.

[6] B. Li, X. Liu, Y. Liu, P. Hu, M. Liu, and C. Wang, "Image-driven panel design via feature-preserving mesh deformation," in *Proc. IFIP Adv. Inf. Commun. Technol.*, vol. 345, 2017, pp. 30–40.

[7] B. Hou, B. Khanal, A. Alansary, S. McDonagh, A. Davidson, M. Rutherford, J. V. Hajnal, D. Rueckert, B. Glocker, and B. Kainz, "3-D reconstruction in canonical co-ordinate space from arbitrarily oriented 2-D images," *IEEE Trans. Med. Imag.*, vol. 37, no. 8, pp. 1737–1750, Aug. 2018.

[8] Y. Wu, Z. Sun, Y. Song, Y. Sun, and J. Shi, "Slicenet: Slice-wise 3D shapes reconstruction from single image," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Barcelona, Spain, May 2020, pp. 1833–1837, doi: 10.1109/ICASSP40776.2020.9054674.

[9] X. Xie, K. Xu, N. J. Mitra, D. Cohen-Or, W. Gong, Q. Su, and B. Chen, "Sketch-to-design: Context-based part assembly," *Comput. Graph. Forum*, vol. 32, no. 8, pp. 233–245, Dec. 2013.

[10] B.-S. Shin and H. S. Jung, "Fast reconstruction of 3D terrain model from contour lines on 2D maps," in *Proc. Asian Simulation Conf.*, in Lecture Notes in Artificial Intelligence, vol. 3398, 2004, pp. 230–239, doi: 10.1007/978-3-540-30585-9_26.

[11] Z. Hui, X. Gu, and L. Na, *Deformation Algorithms for 3D Model: Theory and Practice*. Beijing, China: Publishing House of Electronics Industry, 2017.

[12] C. Rother, V. Kolmogorov, and A. Blake, "Grabcut: Interactive foreground extraction using iterated graph cuts," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 309–314, 2004.

[13] J. Liu, Y. Pan, M. Li, Z. Chen, L. Tang, C. Lu, and J. Wang, "Applications of deep learning to MRI images: A survey," *Big Data Mining Anal.*, vol. 1, no. 1, pp. 1–18, Mar. 2018, doi: 10.26599/BDMA.2018.9020001.

[14] S. Suzuki and K. Be, "Topological structural analysis of digitized binary images by border following," *Comput. Vis., Graph., Image Process.*, vol. 30, no. 1, pp. 32–46, Apr. 1985.

[15] P. Arbeláez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 898–916, May 2011.

[16] J. Mille, R. Boné, P. Makris, and H. Cardot, "2D and 3D deformable models with narrowband region energy," in *Proc. IEEE Int. Conf. Image Process.*, Sep./Oct. 2007, pp. II-57–II-60.

[17] D. R. Martin, C. C. Fowlkes, and J. Malik, "Learning to detect natural image boundaries using local brightness, color, and texture cues," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 5, pp. 530–549, May 2004.

[18] Z. Guo and S. Zhang, "Sparse deep nonnegative matrix factorization," *Big Data Mining Anal.*, vol. 3, no. 1, pp. 13–28, Mar. 2020, doi: 10.26599/BDMA.2019.9020020.

[19] O. Sorkine, D. Cohen-Or, Y. Lipman, M. Alexa, C. Rössl, and H.-P. Seidel, "Laplacian surface editing," in *Proc. Eurographics/ACM SIGGRAPH Symp. Geometry Process. (SGP)*, 2004, pp. 175–184.

[20] Y. Lipman, O. Sorkine, D. Cohen-Or, D. Levin, C. Rossl, and H.-P. Seidel, "Differential coordinates for interactive mesh editing," in *Proc. Shape Modeling Appl.*, 2004, pp. 181–190.

[21] O. Sorkine and M. Alexa, "As-rigid-as-possible surface modeling," in *Proc. Eurographics Symp. Geometry Process.*, Eurographics Association, 2007, pp. 109–116.

[22] Y. Tang, D. Zou, J. Li, and X. Chen, "Local and nonlocal guidance coupled surface deformation," in *Proc. SIGGRAPH Asia Tech. Briefs*, 2014, pp. 1–4.

[23] Y. Yu, K. Zhou, D. Xu, X. Shi, H. Bao, B. Guo, and H.-Y. Shum, "Mesh editing with Poisson-based gradient field manipulation," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 644–651, Aug. 2004.

[24] S.-Y. Chen, L. Gao, Y.-K. Lai, and S. Xia, "Rigidity controllable as-rigid-as-possible shape deformation," *Graph. Models*, vol. 91, pp. 13–21, May 2017.

[25] G.-H. Tan, W. Chen, and L.-G. Liu, "Image driven shape deformation using styles," *J. Zhejiang Univ. Sci. C*, vol. 11, no. 1, pp. 27–35, Jan. 2010.

[26] Q. Fu, X. Chen, X. Su, and H. Fu, "Natural lines inspired 3D shape redesign," *Graph. Models*, vol. 85, pp. 1–10, May 2016.

[27] M.-M. Cheng, N. J. Mitra, X. Huang, P. H. S. Torr, and S.-M. Hu, "Global contrast based salient region detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 3, pp. 569–582, Mar. 2015.

[28] Y. S. Jiang, J. Tang, and X. Chen, "Extracting method of geometrical principal axis for object from bi-tone image," *Comput. Eng.*, vol. 31, no. 18, pp. 56–58, 2005.

[29] J. Shi and C. Tomasi, "Good features to track," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2000, p. 600, doi: 10.1109/CVPR.1994.323794.

[30] Y. Weng, W. Xu, Y. Wu, K. Zhou, and B. Guo, "2D shape deformation using nonlinear least squares optimization," *Vis. Comput.*, vol. 22, nos. 9–11, pp. 653–660, Sep. 2006.

[31] X. Yang, X. Han, Q. Li, L. He, M. Pang, and C. Jia, "Developing a semantic-driven hybrid segmentation method for point clouds of 3D shapes," *IEEE Access*, vol. 8, pp. 40861–40880, 2020, doi: 10.1109/ACCESS.2020.2976847.

[32] L. Shuyu, H. Xie, and J. Caiqin, "Cubic B-spline-interpolation-based mesh splicing and fusion," *J. Image Graph.*, vol. 23, no. 12, pp. 129–137, 2018.

[33] S. Toledo. (Sep. 2003). TAUCS: A library of sparse linear solvers. version 2.2. Tel-Aviv University. [Online]. Available: http://www.tau.ac.il/~stoledo/taucs/

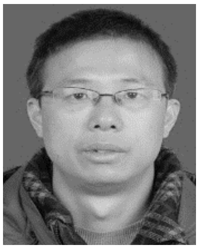
[34] W. H. Press, S. A. Teukolsky, T. Vetterlingw, and B. P. Flannery, *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge, U.K.: Cambridge Univ. Press, 1992.

[35] B. Li, A. Godil, M. Aono, T. Furuya, and L. Li, "SHREC'12 track: Generic 3D shape retrieval," in *Proc. Eurographics Conf. 3D Object Retr.*, Eurographics Association, 2012, pp. 119–126.

[36] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba, "SUN database: Large-scale scene recognition from abbey to zoo," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 3485–3492.

**MIN PANG** received the master's degree in computer application technology from the North University of China (NUC), China, in 2009, where she is currently pursuing the Ph.D. degree in system simulation and modeling. She is also a Lecturer with the School of Data Science and Technology, NUC. Her current research interests include computer graphics, 3D shape deformation, and virtual reality.

**LIGANG HE** (Member, IEEE) is currently a Reader with the Department of Computer, The University of Warwick. He has published more than 130 articles in international conferences and journals, such as the IEEE TC, TPDS, TACO, IPDPS, SC, and VLDB. His research interests include parallel and distributed processing and big data processing.



**FENGGUANG XIONG** was born in China, in 1979. He received the master's degree in computer application technology and the Ph.D. degree in system simulation and modeling from the North University of China (NUC), China, in 2005 and 2018, respectively. He is currently a Lecturer in computer science and control engineering with NUC.

**XIAOWEN YANG** received the master's degree in computer application technology from the North University of China (NUC), China, in 2005. She is currently pursuing the Ph.D. degree in communication and information system. She is also an Associate Professor with the School of Data Science and Technology, NUC. Her research interests include computer graphics, 3D model editing, and virtual reality.

**ZHIYING HE** was born in China, in 1978. She received the master's degree in computer application technology from the North University of China, China, in 2009. She is currently a Lecturer with the Software School, North University of China.

**XIE HAN** received the master's degree in computer science and technology from the North China Institute of Technology, China, and the Ph.D. degree from the Institute of Information Engineering, University of Science and Technology Beijing, China, in 2002. She is currently a Professor with the School of Data Science and Technology, North University of China. Her current research interests include computer vision, simulation, and visualization.