

Received August 31, 2020, accepted September 6, 2020, date of publication September 9, 2020, date of current version September 23, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3022987

Developing a Parametric 3D Face Model Editing Algorithm

MIN PANG¹, LIGANG HE², (Member, IEEE), LIQUN KUANG¹,
MIN CHANG³, ZHIYING HE⁴, AND XIE HAN¹

¹School of Data Science and Technology, North University of China, Taiyuan 030051, China

²Department of Computer Science, University of Warwick, Coventry CV4 7AL, U.K.

³Jinxi Industries Group Company Ltd., Taiyuan 030001, China

⁴Software 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 Key Research and Development Program of China under Grant 2018YFB2101504; 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 Supported by Shanxi Scholarship Council of China (2020-113).

ABSTRACT In the fields of computer graphics and computer vision, a great amount of research and analysis has been conducted on expression-carrying face models. How to construct more realistic and effective 3D face models has become an immense challenge to researchers. This paper proposes a parametric 3D face model editing algorithm based on existing 3D face models. The algorithm takes a number of existing 3D face models with different expressions as input, edits the models through mostly through model deformation and interpolation, and generates a new 3D face model. In particular, the face model editing process begins with selecting multiple face models with different expressions as input. Second, with one of the selected models as the source model and others as the target models, the source model and all target models are registered one by one; meanwhile, the vertex correspondence between the registered models is established. Third, the selected 3D models are parameterized to a planar disc through quasi-conformal mapping. Fourth, relying on the vertex correspondence, a set of corresponding control points between different models are established. The model is then deformed and interpolated under the guidance of the control points and by using the quasi-conformal iteration method, which produces the 2D face models with transitional expressions between the source model and the target models. Finally, the 2D models are restored to the corresponding 3D face models using the model restoration algorithm. Additionally, this paper proposes to use the Beltrami coefficient to guide the quasi-conformal iteration in performing the mapping between two planes. This coefficient then serves as a measure to evaluate the similarity between the edited model and the original one. The proposed algorithm has been evaluated through extensive experiments. The results suggest that compared with existing editing methods, the proposed method is more effective and efficient in constructing various 3D face models.

INDEX TERMS Conformal mapping, deformation, interpolation, model parameterization, quasi-conformal iteration.

I. INTRODUCTION

In the human visual perception system, facial expressions take up an important position, because all human emotions are expressed through changes in facial features. Obtaining face models that carry facial expression features is the basis of advance in computer animation, 3D games and other application fields [1], [2]. Typically, a complete face model is made up of facial geometry and facial texture (that

is, the color traits of the surface). The focus of this study is on modeling face geometry. When the phrases such as “face model construction” are used in this paper, they refer to the construction of face geometry [3] unless otherwise stated.

The construction of face models can be achieved by 3D scanning equipments such as computers and Kinect [4]. They can also be constructed with specialized modeling tools such as Maya, DAZ Studio [5]. Some researchers turn to the 3D model editing technology to construct 3D faces models. Model editing is an important research subject in computer

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

graphics and has found extensive applications in animation, movies, games, computer-aided design, 3D printing, virtual reality, and the like.

Construction of face models was first proposed in 1974 by Parke [6], and has ever since been a hot spot among researchers. Face modeling methods fall into two categories: reconstruction modeling and creative modeling [7], [8].

Reconstruction modeling constructs geometric models of specific faces by extracting geometric attributes from the measurements of real faces. When extracting geometric shapes, researchers usually make use of 3D laser scanners or cameras. Beeler [9] and Abson [10] obtained accurate face models using this method. However, this method is remarkably time-consuming, and the resultant face models are without parameters, not quite suitable to 3D animation manipulation. Weissenfeld *et al.* [11] propose a parametric face model construction method, which works by deforming, based on 3D scanning, a generic face model. This method takes advantage of both a global adaptation process and a local adaptation process to improve modeling accuracy, which avoids undesirable resultant parameterless faces and their associated problems.

A creative modeling method, on the other hand, allows full control of the created face model and enables model editing and modification. There are two ways of creative modeling. One is to create face models for such applications as movie, animation, and games, by use of 3D modeling tools such as Maya; the other is to convert the existing face models into new ones, by use of a model editing method [12]–[15]. Patel *et al.* [16] propose to use the features extracted from a given frontal facial image and transform an existing face model through geometric transformation. Mandan *et al.* [17] proposes to include the convex weights into the traditional Laplace algorithm as a way to optimize facial expression cloning. Jiang *et al.* [14] propose a disentangled representation learning method for 3D face models. They argue that the face model contains the identity and expression features, and decompose these two parts in their algorithm. They create a framework to train decomposition networks and fusion network. Therefore, their algorithm is a face model editing algorithm based on machine learning, which only works with the data of the same type as that of the training data, while our facial model editing algorithm is based on the conformal geometry, and it is not restricted to the prior learning data type. Shouming *et al.* [18] proposes a 3D face modeling algorithm that is based on Kinect and differential mesh deformation technology; this method adds depth information to the construction of face models. It begins with acquiring the color and depth information of the facial object with the Kinect device. It then processes the depth and the color images and uses the color image to build a standard face model. Based on the depth information contained in the depth image, the Laplace transform is performed on the standard face model, to produce a more realistic 3D face model. With improvements in differential deformation technology, face model editing techniques based on differential deformation

have received more and more attention. It can be seen that the differential deformation technology, by rotating, translating, and scaling the face model, is satisfactory in preserving the model's local details. In essence, differential deformation is a way to solve linear systems. It involves a small amount of calculation and low time complexity. However, differential coordinates are sensitive to rotation, likely to cause local distortion of the mesh's local information – in the case of large-scale deformation, such distortion can be severe. In summary, for a face model with rich details but unclear skeleton, when performing direct deformation operations on the face model the existing differential-based deformation technology suffers from poor deformation effect and large local distortion [19].

Computational conformal geometry, an interdisciplinary field between mathematics and computer science, transforms geometric theory into the engineering computing theory. Conformal mapping [20] is one of its core components. The parameterization of a surface is a process that establishes a smooth bijection between the surface and a plane area. Angular or areal distortion may occur in this process. In the conformal geometry, the mapping that does not incur the angular distortion is called conformal mapping. The biggest advantage of conformal mapping lies in the local conformality and the invariance of the expansion and contraction, which allows the constancy of angles and shapes in infinitesimal objects, but not necessarily the size constancy. Quasi-conformal mapping [21], [22] is a generalization of conformal mapping. It is a diffeomorphism with bounded conformal distortion between Riemann surfaces and can be applied to the modulus problems and uniformization problems as well as harmonic analysis of Riemann surfaces. It has been used with success in computer-related fields, such as 3D model registration [23] and face recognition [24]. Quasi-conformal mapping allows conformal transformation of different spatial dimensions and is locally conformal [22], so theoretically it is applicable to 3D model editing to enable deformation and interpolation of 3D models. It also provides a new idea for 3D model editing so that editing operations in a 3D space are reduced to a 2D plane.

In view of the above investigations, this paper proposes a new model editing method, namely a parametric 3D face model editing algorithm based on Iterative Closest Point (ICP) registration. First, multiple 3D face models with different expressions are selected. One is defined as the source model and the others as target models. Second, the source model and all target models are sequentially registered by the ICP algorithm, and the vertex correspondence is established. The registered models are parameterized to a 2D plane through quasi-conformal mapping. The quasi-conformal iterative method is employed to complete the deformation and interpolation of the 2D plane. After the above processing, the editing problem of a 3D model is converted into a 2D plane problem, which simplifies the problem. Unlike deformation and interpolation of a 3D model, the processing on a 2D space involves fewer control points,

which not only simplifies the interactive selection of control points and makes the operation easier, but also avoids the elusive selection of points in free deformation and skeleton deformation and averts difficulty of skeleton extraction. To finish with, the 2D model from quasi-conformal iterative mapping is restored to a 3D model using the model recovery algorithm. By making full use of geometric topology information, the model editing problem to be processed in a 3D space is converted to a problem to be processed in a 2D space, which not only reduces the complexity of the problem but also avoids the problem of surface distortion caused by model rotation during model deformation, and thus overcomes the problem of rotation distortion during differential deformation. The experimental results suggest that the proposed method is able to preserve satisfactorily the local features of models, involves lower time complexity, gives good deformation effect, is applicable to any face model with borders, and offers a strong robustness. Furthermore, When we run the proposed algorithm in the experiments, we only need to specify the input model manually. All the rest operations are automatically completed by the algorithm. The algorithm is very user-friendly and suitable to 3D model designers of any level.

II. FRAMEWORK OF PARAMETRIC 3D FACE MODEL EDITING ALGORITHM BASED ON ICP REGISTRATION

This section overviews the framework of the parametric 3D face model editing algorithm based on the ICP registration. The algorithm will be presented in more detail in the next section.

One of the popular 3D modeling methods is to generate new 3D models with different appearance from existing 3D models through model deformation and interpolation. Unlike generating 3D face models through 3D scanning equipment, which is beset with low accuracy, low efficiency, and high computational complexity, to generate new 3D face models through model editing fares better in model accuracy and computational complexity.

The proposed model editing algorithm begins with selecting from a 3D model database a number of face models wearing different expressions, one of them being defined as the source model, from which a set of characteristic points are extracted. The rest of the selected face models are used, one by one, as the target models; ICP registration is performed with the source model sequentially, and the 3D models in different coordinate systems are unified into the same coordinate system. The spatial transformation relationship is found between the 3D models, and correspondence between their vertex sets is established. Second, the quasi-conformal mapping method is used to parameterize the selected 3D models to a 2D plane. Third, based on previously selected characteristic point sets, corresponding point sets are sequentially extracted on the 2D planes of the source model and the target models; with these point sets, control points are established for model editing. Then, on the basis of the source model, relying on the target model control point set

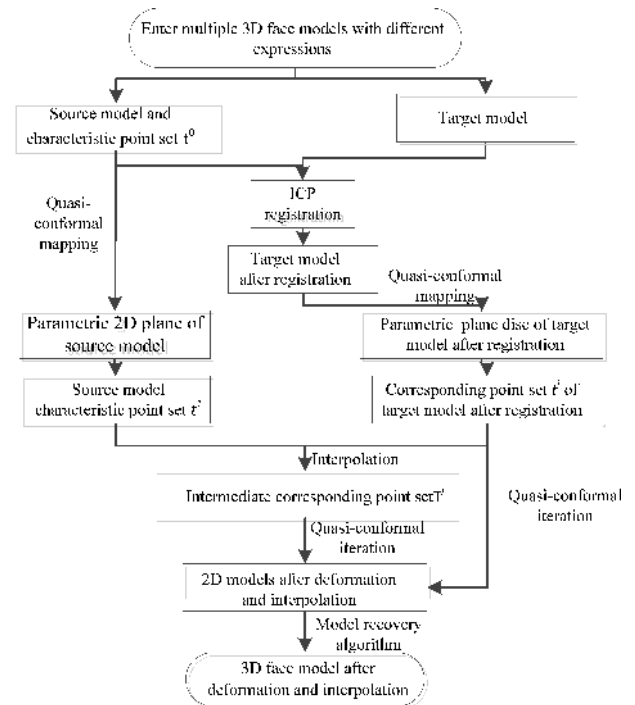


FIGURE 1. Flow chart of parametric 3D face model editing algorithm.

and the interpolation control point set, the deformation and interpolation of the 2D models are completed by the quasi-conformal iterative method, to arrive at 3D models approximate to the target models and arrive at intermediate models with expression lying between the source model and the target models. Finally, these generated 2D models are restored to 3D models, and this completes smooth transition of facial expressions.

The overall workflow of this algorithm is shown in Fig.1 (the detailed steps of the algorithm are described in Section III).

Algorithm 1: Parametric 3D face model editing algorithm based on ICP registration

Input: Multiple 3D face models with different expressions, the number of input models being k ($k \geq 2$)

Output: 3D face models similar to the target models, and 3D face models showing the transitional expression between the source model and the target models

Step 1: One of the selected face models is defined as the source model, with the rest $k-1$ models as the target models. The source model and the target models are sequentially registered by ICP algorithm, and the correspondence relationship between their vertex sets is established. After this, all the registered models will be used as the target models during subsequent model editing to create new face models and complete smooth transition of facial expressions;

Step 2: Select the characteristic point set $t^0 = \{t_i^0, i = 1, 2, \dots\}$ in the source model;

Step 3: Through quasi-conformal mapping, all the selected models are parameterized to a 2D planar disc;

Step 4: Based on the characteristic point set t^0 established in Step 2, the corresponding vertex set $t^1 = \{t_i^1, i = 1, 2, \dots\}$ is extracted from the parametric 2D plane of the source model; according to the vertex correspondence established in this algorithm step 1, a vertex set $t^j = \{t_i^j, i = 1, 2, \dots\}$ ($j = 2, \dots, k$) with vertex correspondence is obtained from the parametric 2D plane of the target model. From the source model vertex set t^1 and the target model vertex sets t^j , multiple intermediate point sets $T^j = \{T_i^j, i = 1, 2, \dots\}$ ($j = 1, 2, \dots$) are generated by interpolation;

Step 5: The vertex set t^1 in the parametric 2D plane of the source model is used as the starting positions of the control points for mesh editing, t^j and T^j constitute sequentially the target positions of the movement of the control points, and then, through mapping by use of the quasi-conformal iteration algorithm, the corresponding 2D model is reached;

Step 6: The 2D model generated in Step 5 is restored, by the model restoration algorithm, to the 3D space, to arrive at a 3D model approximate to the target model and produce a transitional expression model between the source model and the target model.

III. OUR METHOD

In this section, we present in detail the parametric 3D face model editing algorithm. Face models with different expressions are selected from a 3D face model database, using one of them as the source model and the rest as the target models. Then, using the ICP algorithm, the source model and the target models are registered one by one, and they are transferred to the same coordinate system, and the correspondence between their vertices are established in this process.

A. ICP REGISTRATION

The ICP algorithm is a 3D model registration algorithm proposed by Besl and McKay [25] in 1992. After continual improvements by more researchers [26]–[29], the ICP registration algorithm has become a mainstream 3D model registration algorithm, which is widely used in 3D model retrieval, 3D model recognition and reconstruction as an indispensable preprocessing step [30]. The ICP registration algorithm seeks to find the spatial transformation relationship between two given 3D data point sets, that is, the rotation matrix R and translation matrix T between the two 3D data point sets. The algorithm iteration ends when the matching data is such that a certain metric criterion for optimal matching is reached, and this also completes spatial matching of the 3D data point sets and establishes the vertex correspondence between them.

In this study, ICP registration is used to complete the preprocessing of the 3D face models, whereby two face models are unified in the same coordinate system through registration and in the meantime the vertex correspondence between them is established.

Algorithm 2: The ICP registration algorithm process is as follows

Input: Vertex sets of two 3D face models in different coordinate systems

Output: The spatial transformation relationship (rotation matrix R and translation matrix T) of the two 3D vertex sets, and the correspondence between the vertices

Step 1: Take two 3D vertex sets $M = \{M_i, i = 1, 2, \dots, n_M\}$, $N = \{N_i, i = 1, 2, \dots, n_N\}$ where the set M is the vertex set corresponding to the 3D face models other than the source model, and the set N is the vertex set corresponding to the source model;

Step 2: Find the points of the source model vertex set N that correspond to the points of the target model vertex sets M , select point $N_i^k \in N$ in the source model point set N (k being the number of iterations of the algorithm), find the distance between each point $M_i^k \in M$ in the point set M and $N_i^k \in N$, and let $\|M_i^k - N_i^k\| = \min$, then the corresponding point is found. Finally the point set $M1 = \{M_i, i = 1, 2, \dots, n_M\}$ is found, that is the set corresponding to point set N in point set M ;

Step 3: Find the center of gravity of the two data point sets $M1$ and N , and shift their center of gravity to the origin of coordinates;

Step 4: Calculate the covariance matrix H of the two point sets, and perform singular value decomposition on the covariance matrix H to get the rotation matrix R^k and the translation matrix T^k , such that

$$\sum_{i=1}^n \|R^k N_i^k + T^k - M1_i^k\| = \min \quad (1)$$

A new 3D point set is found from R^k and T^k

$$N^{k+1} = \{N_i^{k+1} = R^k N_i^k + T^k, N_i^k \in N\} \quad (2)$$

Step 5: Evaluate the new distance d^{k+1} ,

$$d^{k+1} = \frac{1}{n} \sum_{i=1}^n \|N_i^{k+1} - M1_i^k\|^2 \quad (3)$$

When $d^{k+1} < \tau$ (τ is a given convergence threshold) or when the number of iterations is greater than the preset maximum number of iterations, the algorithm ends, otherwise repeat step 2 to step 5. In our algorithm, generally $\tau = 0.001$. The value of τ influences the registration accuracy and, therefore, the final model editing result. Table 1 shows how the registration matching rate varies according to the values of τ . It can be seen that among all the values $\tau = 0.001$ corresponds to a matching rate closest to 1. The matching rate is based on the root mean square error of the vertex sets of the two registration models at the end of the registration. The root mean square error of the k -th iteration in registration is denoted by $ER(k)$, which is formulated in Eq. (4); and the matching rate at the end of registration is MR , formulated in Eq. (5):

$$ER(k) = \sqrt{\frac{\sum_{j=1}^n \|N_j^{k+1} - M1_j^k\|^2}{n}} \quad (4)$$

$$MR = \frac{ER(1) - ER(i)}{ER(1)} \quad (5)$$

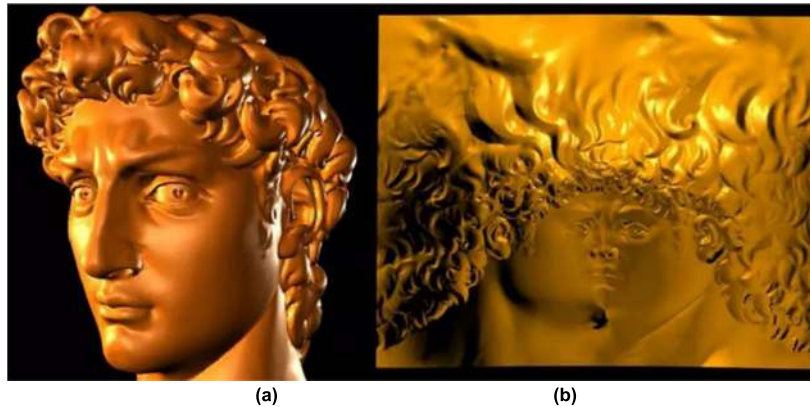


FIGURE 2. Conformal mapping from a curved surface to a plane area [11] (a) 3Dmodel; (b) Parametric 2D plane.

TABLE 1. The influence of τ on the matching rate.

Model	Matching rate $\tau=0.001$	Matching rate $\tau=0.005$	Matching rate $\tau=0.01$	Matching rate $\tau=0.015$
Pig eight ring	0.9814	0.9281	0.8653	0.7367
Old man	0.9975	0.9925	0.9789	0.9132
Man	1.0000	0.9973	0.9932	0.9886
Woman	0.9956	0.9874	0.9435	0.8715

B. QUASI-CONFORMAL MAPPING

Conformal mapping [11] is the basis of computational conformal geometry. Conformal geometry mainly studies the invariants under conformal transformation. Conformal mapping is a local conformality-based process that maps the research object from one area to another through an analytical function. Relying on this feature of conformal mapping, the regional boundaries expressed by some irregular and inscrutable mathematical formulas can be mapped to regular or well-known regional boundaries. The biggest merit of conformal mapping lies in the local conformity and the invariance of the expansion and contraction rate, allowing it to keep unchanged the angle of any curve and the shape of an infinitesimal object, regardless of the size of the object.

A 3D surface in the real world always has a natural conformal structure, being of Euclidean geometry or spherical geometry or hyperbolic geometry. Therefore, these 3D surfaces can be parameterized onto a plane, a spherical or hyperbolic space through quasi-conformal mapping, making it possible to change 3D numerical geometric processing into processing in a 2D space [31].

In parameterizing a 3D model, the choice is usually areal conformality or angular conformality. In this study, the local angular conformality of quasi-conformal mapping was used to complete the conformal parameterization and dimensionality reduction of the models.

Definition 1: Given two reversible mappings $\varphi : (M_1, g_1) \rightarrow (M_2, g_2)$ between surfaces with Riemann metric

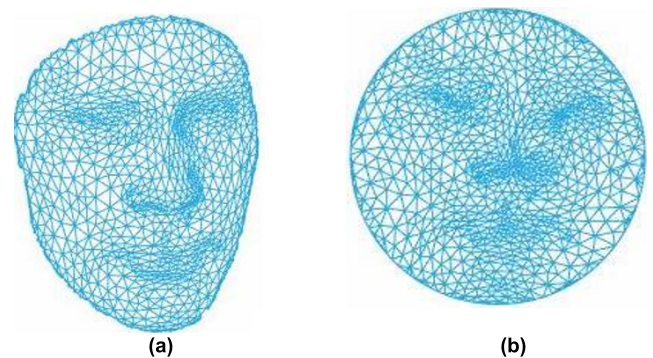


FIGURE 3. Parameterization of a 3D face model to a 2D plane (a) Geometric topology of the original 3D model; (b) Parameterizing to 2D plane.

(M_1 representing the source surface, M_2 representing the target surface, g_1 and g_2 representing the Riemann metrics on the source and target surfaces respectively), if the mapping φ -induced pullback metric φ^*g_2 and the initial metric g_1 differ by a scalar function $\lambda: S1 \rightarrow R$ (R is a complex plane), and the following equation is satisfied

$$\varphi^*g_2 = e^{2\lambda}g_1 \tag{6}$$

then, φ is called a conformal bijection, and the two surfaces are conformally equivalent and share the same conformal invariant.

Figure2 shows the conformal mapping from a 3D surface to a planar area. The head of King David in Fig. 2a is a 3D model obtained with a 3D scanner. It is a surface with complex geometric properties. On Fig. 2b, the 3D model is mapped to the 2D plane through conformal mapping. Figure 2b shows that although the curvature becomes zero, the details of the human face, including the shape of the eyebrows, nose, lips, and hair curls, are preserved. The parameterization of the curved surface through quasi-conformal mapping, though causes overall distortion, keeps local shapes unchanged.

Algorithm 3: 3D model parameterization based on quasi-conformal mapping

Input: 3D mesh model

Output: Mesh model parameterized to a 2D plane

Step 1: Calculate the boundary of the 3D model and record the boundary point set.

Step 2: Calculate the length and angle of the boundary lines, and parameterize the boundary to the disk boundary.

Step 3: Evaluate the cotangent Laplacian matrix of the internal point set, and parameterize this set into the interior of the 2D disc.

Figure 3 shows the result of parametric conformal mapping of a face model to a 2D plane. Figure 3a is the 3D geometric model of a face selected from the model database. Figure 3b is what results from parameterizing the 3D face model to a 2D plane.

C. DETERMINING THE CONTROL PARAMETERS OF THE MODEL EDITING PROCESS

The 3D model editing technology includes model deformation and model interpolation. 3D model deformation and 3D model interpolation are two basic technologies in the field of computer animation [32], [33] and are an important means to obtain new models from existing 3D models.

Model deformation technology seeks to direct the shape change of the entire model by use of the changes of selected existent model control points. The model interpolation technology attempts to establish from the given geometric models their transition sequence, which reflects changes between these models. Interpolation has an important task to establish a one-to-one correspondence between existing models and to generate a change path between corresponding vertex positions. In Section III.A, the ICP registration step establishes such vertex correspondence between the source model and the target models. The following paragraphs deal with how to generate a change path between the corresponding vertices and to find the control point coordinates of the intermediate models. The proposed method, relying on the characteristic point set of the target models while completing the deformation of the source model, finds approximate models of the target models. It also calculates the characteristic points of the source model and the target models, and finds the interpolation control point set, from which an interpolation model is established based on the source model, and then a transition model is built for the existing model.

1) DETERMINING THE CONTROL PARAMETERS OF THE MODEL DEFORMATION PROCESS

With the selected 3D human face source model above, its characteristic points can be obtained as described in step 2 of algorithm 1 and then saved in set $t^0 = \{t_i^0, i = 1, 2, \dots\}$. After that, the characteristic point set $t^1 = \{t_i^1, i = 1, 2, \dots\}$ corresponding to set t^0 are extracted from the parametric 2D plane of the source model, and this set is used as control points for model deformation. Next, according to the vertex correspondence established after ICP registration, characteristic point sets $t^j = \{t_i^j, i = 1, 2, \dots\}$ ($j = 2, \dots, k$) corresponding to t^0 are extracted sequentially from the parametric

2D planes of the target models and are taken as the target positions of the control points during model deformation. Then, using t^1 and t^j as what is called the input information in III. D quasi-conformal iteration step, the deformation of 2D models is completed through quasi-conformal iteration, to produce deformed models that are similar to the target models.

2) DETERMINING THE CONTROL PARAMETERS OF THE MODEL DEFORMATION PROCESS

Transitional expressions are generated by face model interpolation from the existing face models, to create a smooth transition of facial expressions. Based on the characteristic point set t^1 extracted from the parametric 2D plane of the source model described in III C (1), and on the characteristic point set $t^j = \{t_i^j, i = 1, 2, \dots\}$ ($j = 2, \dots, k$) extracted from the parametric 2D plane of the rest $k-1$ face models with different facial expressions corresponding to t^1 , the intermediate point set $T^j = \{T_i^j, i = 1, 2, \dots\}$ ($j = 1, 2, \dots$) is obtained through Eq. (7). Each T^j represents an intermediate point set, and the vertices contained by T^j represent the path along which the source model changes to the vertices of the target models. Finally, the characteristic point set t^1 of the source model and each intermediate point set T^j are used sequentially as the input information of the quasi-conformal iteration in Step III. D. They serve also as the control point information. After the quasi-conformal iteration, the mapped face model is the result of model interpolation, and this completes the transition from the source model to the other target models.

$$T_i = t_i^1 + \sum_{j=2}^k \alpha_{ij} (t_i^j - t_i^1) \quad (i = 1 \dots) \quad (7)$$

where i represents the index value of the corresponding characteristic point in the 2D model generated by the quasi-conformal mapping from the selected face model, α_{ij} is the weight, used to define the impact of each target model on the generated transition model in the interpolation process, $\alpha_{ij} = \frac{d_{im}}{(k-1)d_{ij}}$ ($j = 2 \dots k$), where $(k-1)$ is the number of selected target models, d_{ij} measures the distance between the characteristic points, of the j th face model and of the source model, corresponding to index i , and d_{im} is the distance (usually being $0 < d_{im} \leq \max(d_{ij})$ ($i=1, \dots; j = 2, \dots, k$)) between the characteristic points, of the to-be-generated transition model and the source model, corresponding to index i .

D. QUASI-CONFORMAL ITERATION

After the parameterized 2D models of both the source model and the target models are found in III.B, the mapping relationship between them is computed using the quasi-conformal iterative method, which completes source model-based model deformation and model interpolation processing. First, the previously selected characteristic point set t^1 is moved to the target positions of the characteristic points defined in III. C. When the target position is

t^j , the approximate model of the selected target model is obtained by mapping from the source model. When the target position is T^j , the source model is mapped to arrive at an intermediate model between the source model and the selected target model. Then the optimal value is found with the aid of the Beltrami coefficient. Next, the coordinates of the remaining vertices other than characteristic points are evaluated iteratively until the conformal distortion is minimized.

The Beltrami coefficient represents a set of differential homeomorphisms. The optimal Beltrami coefficient is found in a 2D plane, and from it the associated difference is reconstructed effectively. The linear Beltrami solver is then used to find the optimal Beltrami coefficient so as to compute the associated quasi-conformal mapping. The variability expressed by the Beltrami coefficient guarantees the varied properties of the mapping. Even in dealing with a great deformation, the mapping relationship between the two planes can be found with the quasi-conformal iteration algorithm.

In this paper, topographically the mesh surfaces to be processed are connected open surfaces or closed surfaces with zero genus. The deformation algorithm of the 2D model based on quasi-conformal iteration^[16,34] is described as follows:

Algorithm 4: 2D quasi-conformal iteration algorithm

Input: the 2D plane resulting from the parameterization of the source face model, and the control point set t^1 and its target position t^j or T^j

Output: 2D model after deformation

Step 1: From the control point set t^1 contained in the 2D plane after the parameterization of the source model, and from the target position set t^j or T^j of the control points, the initial mapping relationship f_0 and v_0 between these control points is found, as given by:

$$f_0 = F_b(\mu_0 := 0) \tag{8}$$

$$v_0 = \mu(f_0) \tag{9}$$

where: μ is the Beltrami coefficient, and F_b is the linear Beltrami solver.

Step 2: From the vertex information v_n of the source model mesh structure, the following equations are used to find μ_{n+1} , f_{n+1} , and v_{n+1} .

$$\mu_{n+1} := A(\delta(v_n)) \tag{10}$$

$$f_{n+1} := F_b(\mu_{n+1}) \tag{11}$$

$$v_{n+1} := \mu(f_{n+1}) \tag{12}$$

In Eq. (10), A is the Laplace average operator, and δ is the matrix consisting of vertex coordinates parameterized to the 2D plane.

Step 3: When $\|v_{n+1} - v_n\| \geq \varepsilon$ (ε is a very small number, generally $\varepsilon = 0.01$), Step 1 and 2 are repeated, otherwise, the algorithm ends.

Figure 4 shows a graph resulting from running the 2D model deformation algorithm after quasi-conformal iteration. Figure 4a shows a parametric 2D face model resulting from

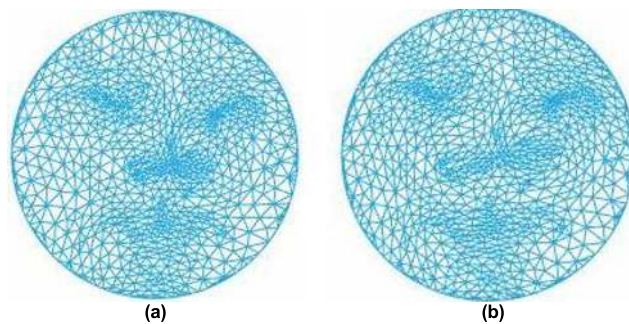


FIGURE 4. Results from quasi-conformal iterative (a) Parametric 2D face model; (b) 2D face model deformed after quasi-conformal iteration.

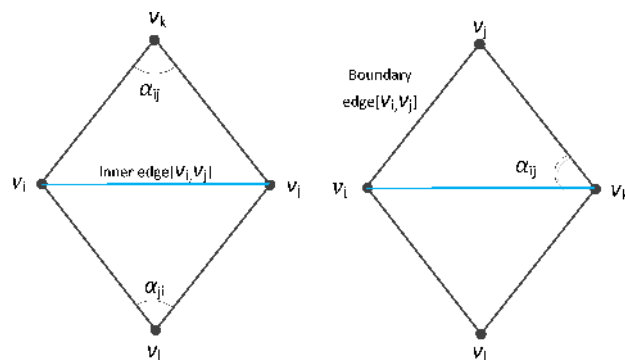


FIGURE 5. Illustration of cotangent evaluation.

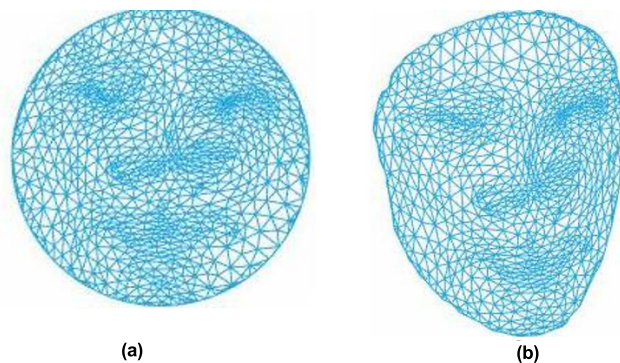


FIGURE 6. Result of 2D to 3D model recovery algorithm operation (a) 2D face model; (b) 3D face model from the model restoration algorithm.

the previous steps, and Figure 4b gives the result of deforming Fig. 4a after quasi-conformal iteration.

E. MODEL RECOVERY ALGORITHM

The 2D to 3D space model recovery algorithm [35] is a process that derives an approximate 3D geometric model from both the topology information of the 2D model mesh and the geometric information of some of its vertices. The first step is to find the cotangent Laplace matrix of the source model (containing not only the topological information of the model but also its geometric information); second, some of the model's points are used as constraints; finally, the vertex information of the 3D model that satisfies the constraints to

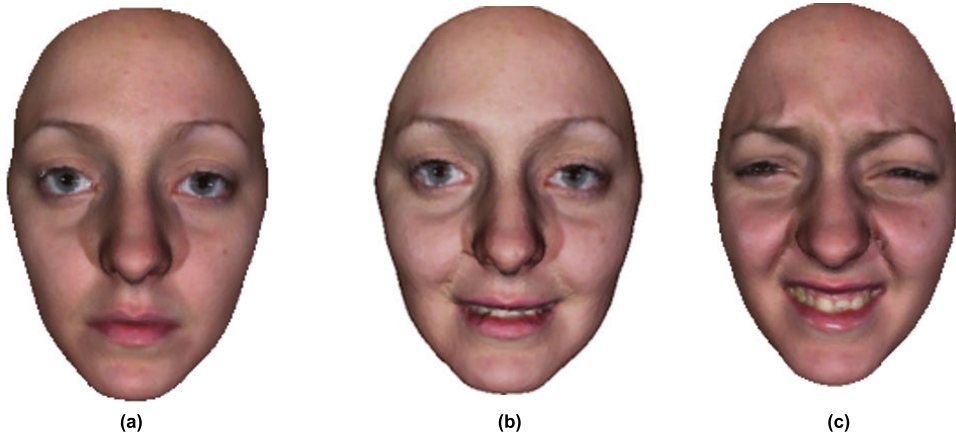


FIGURE 7. Face models with different expressions (a) Source models; (b) Target model; (c) Target model 2.

the highest degree is found using the least square method. The specific algorithm is as follows:

Algorithm 5: 2D → 3D Deformation model restoration algorithm

Input: The 2D face model from model editing in Algorithm 4

Output: A 3D face model approximate to the target model and a 3D face model with transitional expression between the source model to the target model

Step 1: Find the cotangent Laplace matrix of the model as

$$W_{ij} = \begin{cases} \frac{1}{2} (\cot \alpha_{ij} + \cot \alpha_{ji}) & \text{inner edge} \\ \frac{1}{2} \cot \alpha_{ij} & \text{boundary edge} \end{cases} \quad (13)$$

where: α_{ij} and α_{ji} are two angles opposite to side (x_i, x_j) , to be decided as shown in Fig. 5.

Step 2: Build linear systems $Wx = 0$, $Wy = 0$, and $Wz = 0$ with respect to x , y , and z axis respectively

Step 3: Select m vertices as known points, each vertex being such that $V_s = (x_s, y_s, z_s)$, $s \in C$.

$$C = \{s_1, s_2, s_3, s_4, \dots, s_m\} \quad (14)$$

Step 4: Introduce m vertices into the linear equation constructed in Step 2, to form a system $(n+m) \times n$ long, that is, $Ax = b_x$, $Ay = b_y$, $Az = b_z$ where,

$$A = \begin{bmatrix} W \\ F \end{bmatrix} \quad (15)$$

$$F_{ij} = \begin{cases} 1 & j = s_i \in C \\ 0 & \text{otherwise} \end{cases} \quad (16)$$

$$b_{xk} = \begin{cases} 0 & k \leq n \\ x_{s_{k-n}} & n < k \leq n + m \end{cases} \quad (17)$$

$$b_{yk} = \begin{cases} 0 & k \leq n \\ y_{s_{k-n}} & n < k \leq n + m \end{cases} \quad (18)$$

$$b_{zk} = \begin{cases} 0 & k \leq n \\ z_{s_{k-n}} & n < k \leq n + m \end{cases} \quad (19)$$

Step 5: Using the least square method to find the minimum values that satisfy the following linear equations:

$$\|Ax - b_x\|^2 = \|Wx\|^2 + \sum_{s \in C} |x_s - v_s^{(x)}|^2 \quad (20)$$

$$\|Ay - b_y\|^2 = \|Wy\|^2 + \sum_{s \in C} |y_s - v_s^{(y)}|^2 \quad (21)$$

$$\|Az - b_z\|^2 = \|Wz\|^2 + \sum_{s \in C} |z_s - v_s^{(z)}|^2 \quad (22)$$

Step 6: The minimum values are $x = (A^T A)^{-1} A^T b_x$, $y = (A^T A)^{-1} A^T b_y$, $z = (A^T A)^{-1} A^T b_z$, and finally the vertex coordinates are $V_{new} = (x, y, z)$.

Figure 6 shows the rendering images of the face model restored from the 2D plane to the 3D space. Figure 6a is the 2D plan view of the face model, and Figure 6b is the restored 3D face model.

IV. EXPERIMENTS

This section begins with an experiment of verifying the effect of the proposed ICP registration-based 3D face model editing method. A comparison with a previous method in literature is also given. Then, the similarity of the face models is compared in terms of the Beltrami coefficient, which confirms the effectiveness of the proposed editing algorithm.

The experiments were all conducted on a graphics workstation equipped with an Intel Core i5-7300 2.50 GHz processor and 8GB of memory. All face models used in the experiment came from the BU3DFER 3D face database of State University of New York at Binghamton [36] and MeshDGP [37].

A. EFFECTIVENESS OF THE MODEL EDITING METHOD

The face models of a same person but with different facial expressions were tested in experiment. Figure 7a shows the source models, and Figure 7b and Figure 7c show the target

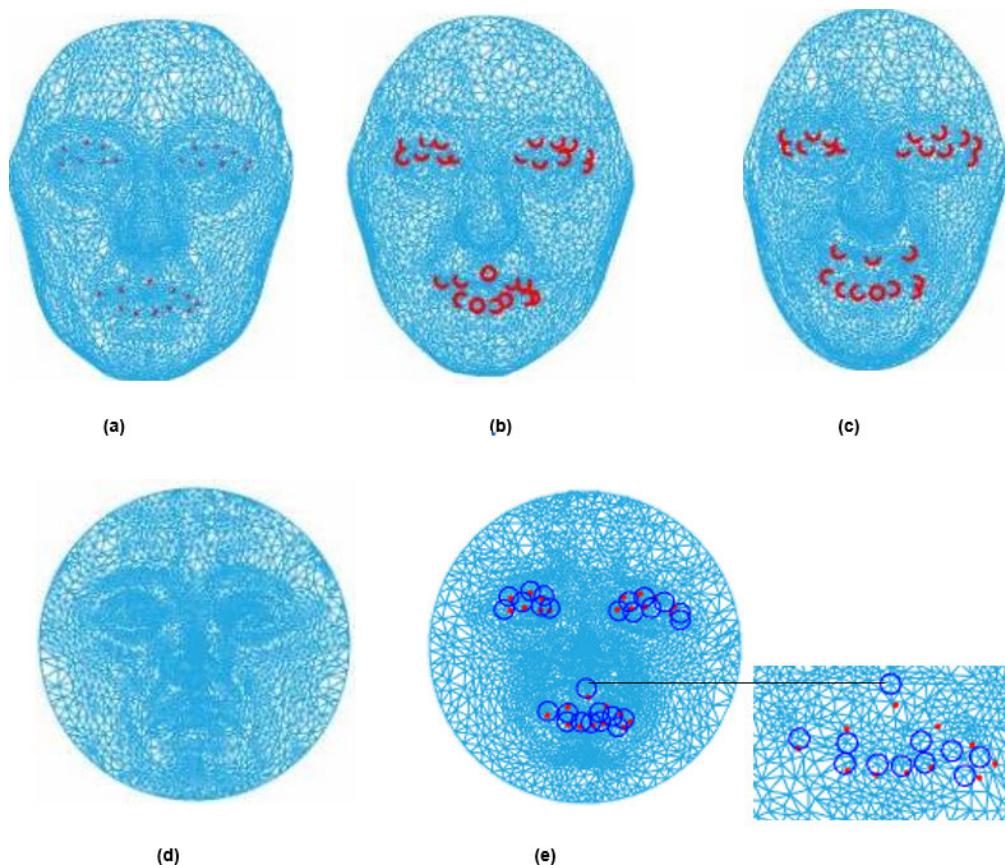


FIGURE 8. Preprocessing of face models (a)Mesh model of source model; (b) Mesh mode of target model 1; (c) Mesh model of target model 2 I; (d) The source model parameterized to a 2D plane I; (e) Red dots indicate starting positions, and blue ones ending positions.

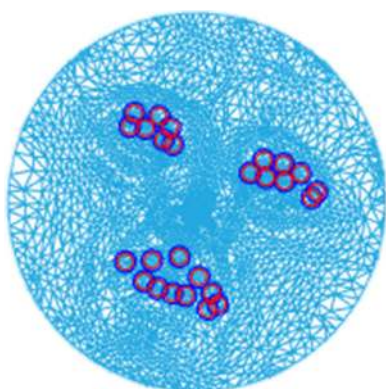


FIGURE 9. Effect of 2D plane deformation.

models corresponding to different expressions. The characteristic points were mostly taken from the eyes and mouths of the models. From these points, constraints were developed for the editing of the source models. The chosen characteristic points are given in Fig. 8a, Fig. 8b and Fig. 8c. Figure 8d is the rendering of the parametric 2D plane of the source models. Figure 8e shows the starting position and target position maps of the characteristic points when deformation is completed on the 2D plane. Figure 9 gives the schematic diagram of the 2D

plane result of the source model using the quasi-conformal iteration as described in Section III.D.

In our experiment, the source model and the two target models were the input. Relying on the deformation parameters, the approximate models of the two target models were restored; relying on the interpolation parameters, a transition model was generated of the expression between the source model and the target models. From these generated 2D model, corresponding 3D models were restored using the model restoration algorithm. The restored 3D models are shown in Fig. 10. Figure 10b and Figure 10c are the face models derived from the control point sets of the source and target models in Fig.8 by mapping the source model through quasi-common iteration. Figure 10a gives a face model with a transitional expression, derived from the source model in Fig. 8 and from the interpolation control point set, after quasi-conformal iterative mapping processing.

Figure 11 gives a point cloud comparison of the two models of Fig. 8c and Fig.10c respectively. As can be seen, there is general consistency, which verifies the accuracy of the proposed model editing algorithm.

We verify the proposed model editing algorithm with various face models, covering different ages, genders and even

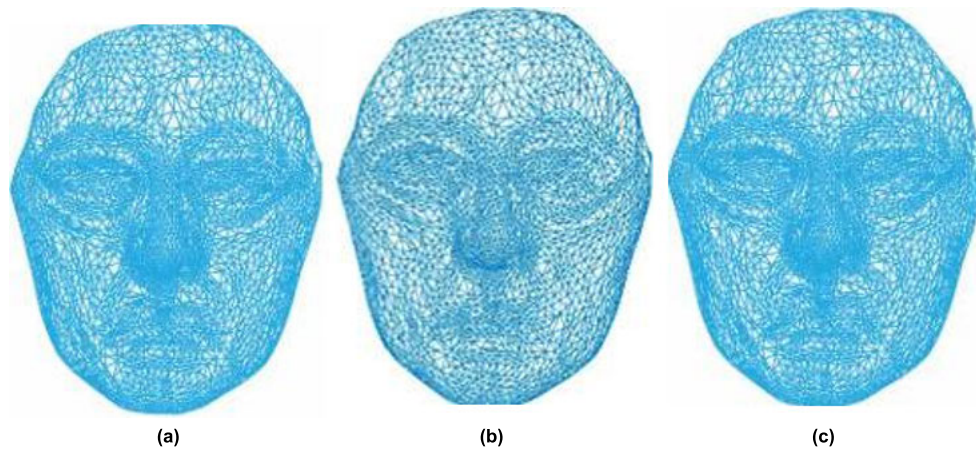


FIGURE 10. Face model after editing (a) Intermediate model 1 after interpolation; (b) Target model 1; (c) Target model 2.

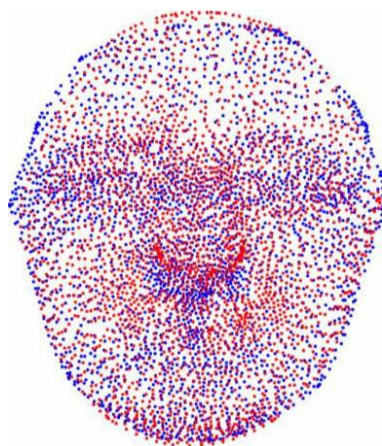


FIGURE 11. The blue point cloud corresponds to the original target model and the red one shows the effect of the proposed deformation algorithm.

cartoon face models. In Fig. 12, column (a) is the source models, column (b) and column (c) are two target models, column (d) show the 3D face model with transitional expressions generated by the proposed editing method, and column (e) is the observation of the resulting models from different perspectives. The results of the experiments with these 3D face models from different data sets show that the proposed method achieves expression changes while maintaining the local details of the model. In the first row of the cartoon head, for example, the source model's mouth in (a) is open and the eyes are half closed, while in target models (b) and (c) the mouths and eyes vary from "open" to "close". The target objects, as seen in (d) and (e), are produced from interpolation. It can be seen from different perspectives, the mouth and eyes movements of the interpolated face models are different from the previous three models, but retain their local details.

The proposed model editing algorithm has its core step rested on the mapping of two planes using the quasi-conformal iteration method. The quasi-conformal

iteration is a process in which the source model, guided by the control points and their target positions, is evaluated for the optimal value using the Beltrami coefficient, and then iteration is performed so that all the vertices of the model are so positioned that the conformal distortion is minimized. A comparison between the 3D models in Fig. 8a and Fig. 10a shows that the model editing implemented by the proposed algorithm is well able to maintain the local features of the face model. Figure 13 show, before and after model editing respectively, the Beltrami coefficient histogram of the 2D plane after the parameterization of the different target model. As can be seen, their change trends are basically the same, which also proves that the proposed method gives very little conformal distortion.

The proposed model editing algorithm makes full use of the geometric topology information, which makes up for the lack of depth information associated with the 2D-to-3D surface process when using images to complete 3D modeling. Additionally, it also minimizes the rotation distortion likely to arise in the model editing. Figure 14a shows the original face model. Figure 14b is the model generated by the proposed algorithm. Figure 14c is the result from the face model editing process presented in [17], which was realized by the differential coordinate mesh editing algorithm. Figure 14d is the result obtained by the adaptive 3D facial model deformation algorithm in [8], which was developed by using Laplacian mesh editing as the mathematical method. It can be seen from Column (c) that the models edited by the method in [17] produce some noticeable distortions. For example, the nose bridge of the man in the second row of Fig. 14 collapses and the mouth of the woman in the fourth row of Fig. 14 twists. The models edited by the method in [8] contain some distortions too. For example, the eyes of the pig eight ring, the mouth of the woman, and the nose of the old man in Fig. 14d are obviously out of shape. In contrast, it can be seen from Fig. 14 that the proposed algorithm steered clear of the rotation distortion associated with the face model editing algorithm in [8] and, [17] and preserved the local features of



FIGURE 12. Results of face model editing algorithm (a) Source models; (b) Target models 1; (c) Target models 2; (d) Resulting Models; (e) Resulting models from a different perspective.

the model effectively. This is also why our algorithm is more robust and produces more realistic results.

B. EFFICIENCY OF THE MODEL EDITING METHOD

Table 2 tabulates the number of control points and the running time required for face model editing. It can be seen

that the proposed algorithm is of a lower time complexity, able to produce a model within a comparative short time. Second, with the proposed algorithm, fewer control points is required for editing, allowing face model editing with fewer control points and able to preserve the local details of the model, which involves less conformal distortion.

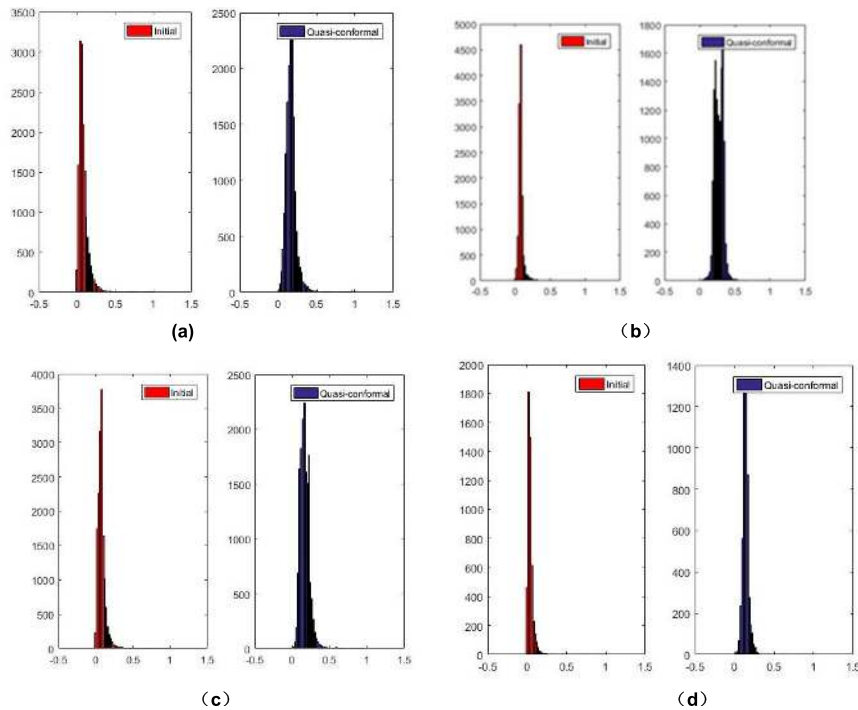


FIGURE 13. The Beltrami coefficient histogram of different model (a)Pig eight ring; (b) old man; (c)man; (d) woman.

TABLE 2. Parameters and running time of the deformation model.

Model	Points	Control points	Time of quasi-conformal iteration (s)	Time of model restoration (s)	Total time (s)
Woman target model 1	2680	26	10.527	5.685	20.201
Woman target model 2	2680	26	9.826	5.876	18.562
Woman intermediate model	2680	26	10.126	5.657	16.035
Pig eight ring target model 1	7556	24	25.012	5.875	32.817
Pig eight ring target model 2	7556	24	24.664	5.753	31.528
Pig eight ring intermediate model	7556	24	24.556	5.787	31.146

For 3D models with rich details and no obvious skeletons, such as the human face model, neither the free deformation method nor the skeleton-based deformation method is satisfactory in model editing. The mesh deformation method based on differential coordinates is therefore more commonly used for face model editing. Below, the proposed method, the differential coordinate mesh editing algorithm proposed by Zhang *et al.* [17] and the Laplacian-based face model deformation proposed by Li *et al.* [8] are compared for their number of control points and running time. The comparison is summarized in Table 3. The method proposed in this paper

produces more realistic deformation effects, all with fewer control points and less running time, as shown in Fig. 14 and Table 3.

C. ABLATION STUDY

In our proposed algorithm, we have two important designs including i) the use of conformal geometry to transform the 3D model editing operations from the 3D space to the 2D space, and ii) the determination of control points for the interpolation, which greatly improve the representation ability of our algorithm. To investigate the effectiveness of each

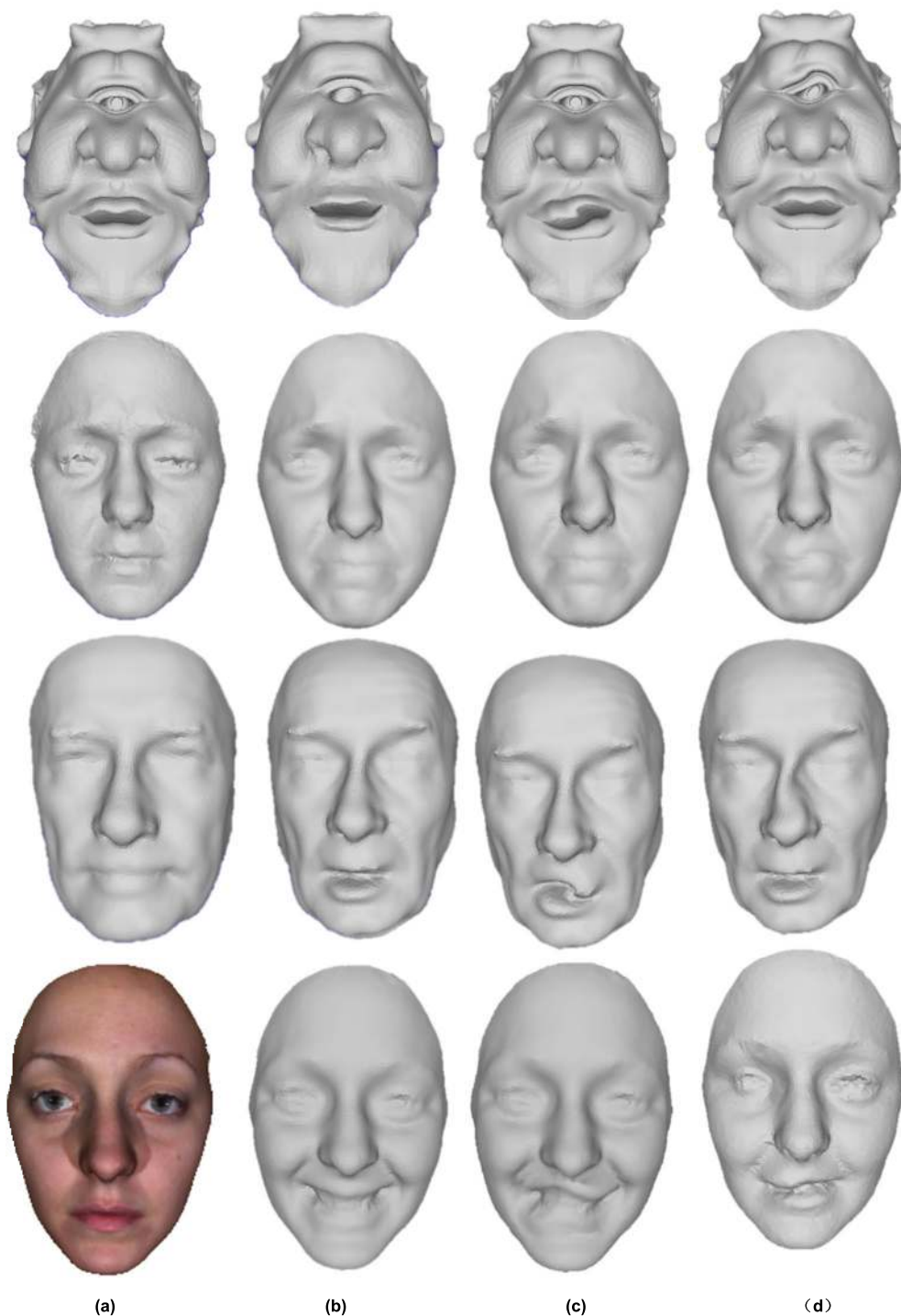


FIGURE 14. Comparison of mesh editing algorithms (a) Original 3D face model;(b) models generated by our proposed algorithm;(c) models.

of these two designs, we conducted the following ablation studies.

First, our algorithm uses the conformal geometry to convert 3D model editing to the 2D space. Another straightforward method is to directly use differential deformation in the 3D space. We conducted the experiments to compare the results

of our algorithm when it is equipped with the conformal geometry method and with the straightforward differential deformation. The results are shown in Fig.15b, where w/o means “without”. It can be seen from Fig.15b that when using the differential deformation, there is the distortion in the relatively large deformation area of the model, such as

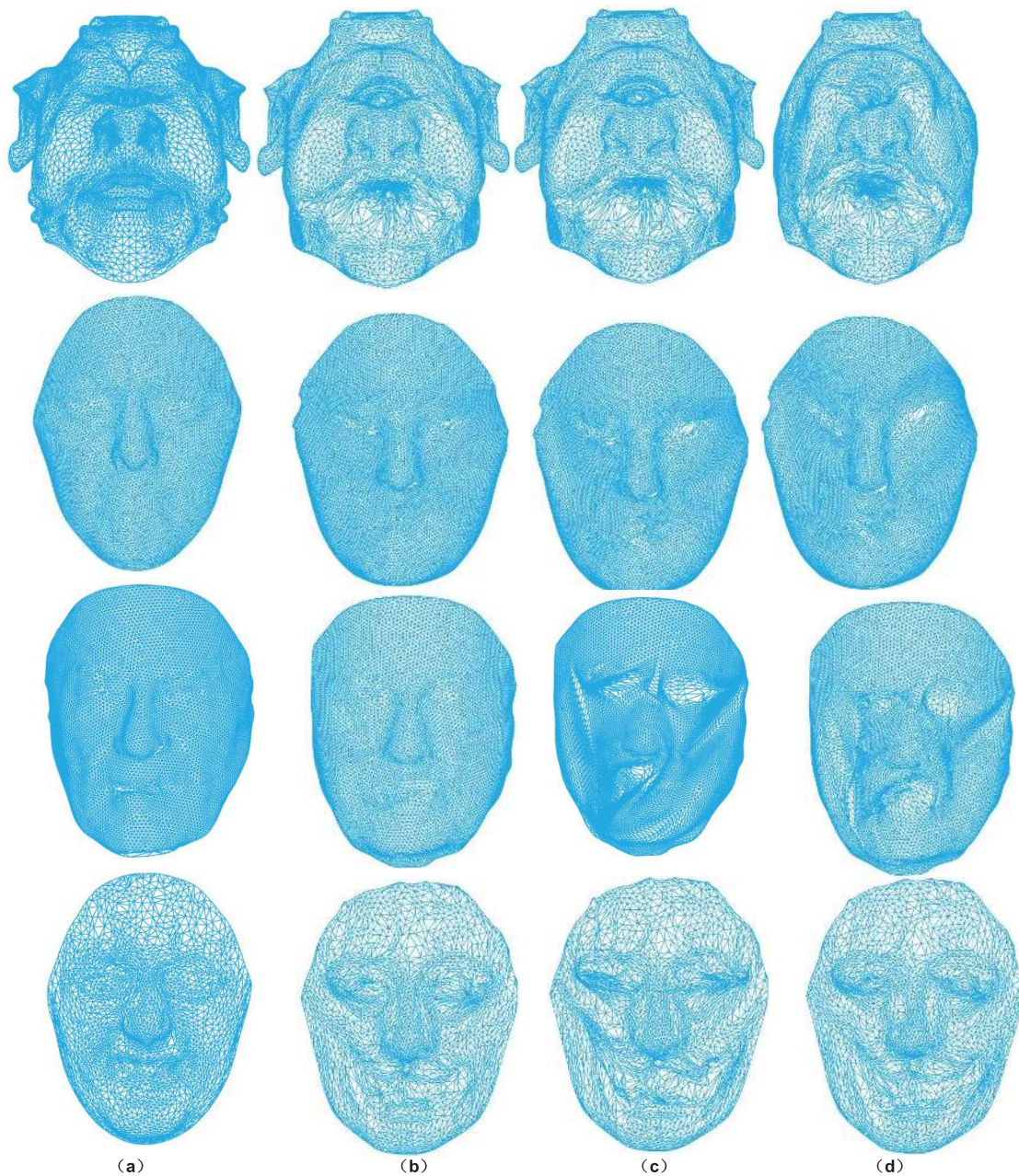


FIGURE 15. Qualitative results of each component function (a) Results of our proposed algorithm; (b) Results of our algorithm without the conformal geometry; (c) Results of our algorithm without our careful determination of the control points for the interpolation; (d) Results of our algorithm without both designs.

the features of the noses in the first row and the features of eyes in the second row.

Second, another important design in our algorithm is the careful calculation of the control points for the interpolation. An intuitive alternative is to define the control point positions directly based on the distance between the corresponding vertices in two models. We conducted the experiments to compare the results of our algorithm when it is with our proposed method of determining the control points and when it is with the intuitive method. The results are shown in Fig. 15c. It can be seen that there is obvious distortion in a relatively

large deformation area of the model. For example, the mouth and eyes in the third row are obviously distorted.

We also conducted the experiments with the algorithm that does not incorporate both of the above designs. The results are shown in Fig. 15d. It can be seen that there are severe distortions in the facial region of every model.

It can be observed from the above ablation studies, both conformal geometry for converting the 3D space model editing to the 2D space, and careful determination of control points for the interpolation contributes to the effective of the proposed algorithm.

TABLE 3. Performance comparison between our algorithm and other deformation algorithms.

Model	Points	Control points	Total time of our algorithm (s)	control points of the method [17]	Total Time of the method [17] (s)	control points of the method [8]	Total Time of the method [8] (s)
Woman target model 1	2680	26	20.201	150	21.685	135	24.371
Woman target model 2	2680	26	18.562	150	20.876	135	23.968
Woman interpolation model	2680	26	16.035	150	20.657	135	23.749
Pig eight ring target model 1	7556	24	32.817	200	37.637	180	41.832
Pig eight ring target model 2	7556	24	31.528	200	35.981	180	40.632
Pig eight ring intermediate model	7556	24	31.146	200	35.841	180	40,153

V. CONCLUSION

Face model construction technology plays an important role in the development of computer vision. With the advancement of computer vision and 3D animation technologies, 3D face models are put to more extensive applications. An unavoidable challenge before researchers is to find a way to obtain more high-quality 3D face models quickly and effectively.

Drawing on the characteristic of local conformality of conformal mapping, this paper proposes a 3D face model editing algorithm based on parameterization. Our algorithm parameterizes several existing registered 3D face models to 2D plane through quasi-conformal mapping, and interpolates the face model by applying quasi-conformal iteration method under the guidance of fewer characteristic points to obtain the 2D plane with transitional expressions, and then restores it to a 3D model. Finally our algorithm achieves the purpose of smooth transition of facial expressions. It can be seen that the proposed algorithm enabled smooth transition of 3D facial expressions through a face model experiment, the result of which suggests that the proposed algorithm is well able to preserve the local features of the model and is robust.

In future, we plan to further address the deficiencies in model editing methods. First, in selecting characteristic points, the ICP registration algorithm is used to register the source model and the target model. ICP registration can deal with normal facial expression deformation. However, according to the principle of ICP algorithm, it is impossible for it to correctly register the expression changes beyond the normal limit. For large expression changes, we will investigate the effectiveness of deep learning-based registration approaches, aiming to enhance the registration accuracy and create more accurate correspondence between feature points. Second, improvement can be considered in the selection of characteristic points for model deformation and interpolation. For example, the selection of characteristic points can be made automatic and more reasonable through comparing

the differences of expressions. Still, the model restoration algorithm requires knowing some vertices, and in the future, improvement can be made on the definition method of these known points.

REFERENCES

- [1] W. S. Lee, P. Kalra, and N. Magnenat-Thalmann, "Model based face reconstruction for animation," in *Proc. Multimedia Modeling*, Singapore, 1997, pp. 323–338.
- [2] I. Kemelmacher-Shlizerman and S. M. Seitz, "Face reconstruction in the wild," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 1746–1753.
- [3] C.-M. Cheng and S.-H. Lai, "An integrated approach to 3D face model reconstruction from video," in *Proc. IEEE ICCV Workshop Recognit., Anal., Tracking Faces Gestures Real-Time Syst.*, Jul. 2001, pp. 16–22.
- [4] Z. Mandun, H. Jianglei, N. Shenruoyang, and H. Chunmeng, "Performance-driven facial expression real-time animation generation," in *Proc. Int. Conf. Image Graph.* Cham, Switzerland: Springer, 2015, pp. 46–54.
- [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—ICISP (Lecture Notes in Computer Science)*, vol. 8509, A. Elmoataz, O. Lezoray, F. Nouboud, and D. Mammass, Eds. Cham, Switzerland: Springer, 2014, doi: 10.1007/978-3-319-07998-1_70.
- [6] B. Li, X. Liu, Y. Liu, P. Hu, M. Liu, and C. Wang, "Image-driven panel design via feature-preserving mesh deformation," in *Computer and Computing Technologies in Agriculture IV—CCTA (IFIP Advances in Information and Communication Technology)*, vol. 345, D. Li, Y. Liu, and Y. Chen, Eds. Berlin, Germany: Springer, 2011, doi: 10.1007/978-3-642-18336-2_5.
- [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, doi: 10.1109/TMI.2018.2798801.
- [8] J. Li, Y. Zhang, P. Xu, S. Lan, and S. Li, "Adaptive 3D facial deformation based on personal characteristics," in *Proc. 12th Int. Conf. Fuzzy Syst. Knowl. Discovery (FSKD)*, Aug. 2015, pp. 1258–1261.
- [9] T. Beeler, B. Bickel, P. Beardsley, B. Sumner, and M. Gross, "High-quality single-shot capture of facial geometry," *ACM Trans. Graph.*, vol. 29, no. 4, pp. 40.1–40.9, doi: 10.1145/1833351.1778777.
- [10] B. S. Shin and H. S. Jung, "Fast reconstruction of 3D terrain model from contour lines on 2D maps," in *Systems Modeling and Simulation: Theory and Applications—AsiaSim (Lecture Notes in Computer Science)*, vol. 3398, D. K. Baik, Ed. Berlin, Germany: Springer, 2005, doi: 10.1007/978-3-540-30585-9_26.
- [11] A. Weissenfeld, N. Stefanoski, S. Qiuqiong, and J. Ostermann, "Adaptation of a generic face model to a 3D scan," in *Proc. Workshop Immersive Commun. Broadcast Syst.*, 2005, pp. 1–4.

- [12] A. Zhiqi and W. Shaorong, "Free form deformation based face cartoon stylization," *Comput. Aided Drafting, Des. Manuf.*, vol. 26, no. 2, pp. 27–33, 2016.
- [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, pp. 1–18, 2018, doi: [10.26599/BDMA.2018.9020001](https://doi.org/10.26599/BDMA.2018.9020001).
- [14] Z.-H. Jiang, Q. Wu, K. Chen, and J. Zhang, "Disentangled representation learning for 3D face shape," in *Proc. CVPR*, Jun. 2019, pp. 11949–11958.
- [15] T. B. Mudiyansele and Y. Zhang, "Feature selection with graph mining technology," *Big Data Mining Anal.*, vol. 2, no. 2, pp. 73–82, Jun. 2019.
- [16] N. Patel and M. Zaveri, "3D facial model reconstruction, expressions synthesis and animation using single frontal face image," *Signal, Image Video Process.*, vol. 7, no. 5, pp. 889–897, Sep. 2013, doi: [10.1007/s11760-011-0278-9](https://doi.org/10.1007/s11760-011-0278-9).
- [17] M. Zhang, X. Ge, and J. Huo, "Facial expression cloning optimization method based on laplace operator," *Comput. Eng. Appl.*, vol. 52, no. 6, pp. 178–181, 2016, doi: [10.3778/j.issn.1002-8331.1403-0338](https://doi.org/10.3778/j.issn.1002-8331.1403-0338).
- [18] S. Zhang, "Sparse deep nonnegative matrix factorization," *Big Data Mining Anal.*, vol. 3, 2017, doi: [10.26599/BDMA.2019.9020020](https://doi.org/10.26599/BDMA.2019.9020020).
- [19] O. Sorkine, D. Cohen-Or, Y. Lipman, M. Alexa, C. Roessl, and H.-P. Seidel, "Laplacian surface editing," in *Proc. Symp. Geometry Process., Eurographics (SGP)*, vol. 274, 2004, pp. 179–188.
- [20] Y. Lipman, O. Sorkine, D. Cohen-Or, D. Levin, C. Rossi, and H. P. Seidel, "Differential coordinates for interactive mesh editing," in *Proc. Shape Modeling Appl.*, Genova, Italy, 2004, pp. 181–190, doi: [10.1109/SMI.2004.1314505](https://doi.org/10.1109/SMI.2004.1314505).
- [21] C. P. Herzog, K. W. Huang, and K. Jensen, "Universal entanglement and boundary geometry in conformal field theory," *J. High Energy Phys.*, vol. 2016, no. 1, p. 162, 2016, doi: [10.1007/jhep01\(2016\)162](https://doi.org/10.1007/jhep01(2016)162).
- [22] M. Goswami, X. Gu, V. P. Pingali, and G. Telang, "Computing Teichmüller maps between polygons," *Found. Comput. Math.*, vol. 17, no. 2, pp. 497–526, 2017, doi: [10.1007/s10208-015-9294-4](https://doi.org/10.1007/s10208-015-9294-4).
- [23] L. M. Lui, K. C. Lam, S.-T. Yau, and X. Gu, "Teichmüller mapping (T-map) and its applications to landmark matching registration," *SIAM J. Imag. Sci.*, vol. 7, no. 1, pp. 391–426, 2014, doi: [10.1137/120900186](https://doi.org/10.1137/120900186).
- [24] F. Marcolin and E. Vezzetti, "Novel descriptors for geometrical 3D face analysis," *Multimedia Tools Appl.*, vol. 76, no. 12, pp. 13805–13834, Jun. 2017, doi: [10.1007/s11042-016-3741-3](https://doi.org/10.1007/s11042-016-3741-3).
- [25] P. J. Besl and N. D. McKay, "A method for registration of 3-D shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, pp. 239–256, Feb. 1992.
- [26] J. Cook, V. Chandran, S. Sridharan, and C. Fookes, "Face recognition from 3D data using iterative closest point algorithm and Gaussian mixture models," in *Proc. 2nd Int. Symp. 3D Data Process., Visualizat. Transmiss.*, Sep. 2004, pp. 502–509.
- [27] J. Yang, Y. Zhang, and L. Huang, "Research on 3D model registration by improved ICP algorithm," *J. Frontiers Comput. Sci. Technol.*, vol. 12, no. 1, pp. 153–162, 2018.
- [28] Y. Yang, D. Fan, S. Du, B. Chen, and Y. Gao, "Point set registration with similarity and affine transformations based on bidirectional KMPE loss," *IEEE Trans. Cybern.*, early access, Oct. 18, 2019, doi: [10.1109/TCYB.2019.2944171](https://doi.org/10.1109/TCYB.2019.2944171).
- [29] H. M. Le, T.-T. Do, T. Hoang, and N.-M. Cheung, "SDRSAC: Semidefinite-based randomized approach for robust point cloud registration without correspondences," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 124–133.
- [30] J. Liu, X. Zhang, and J. Zhu, "ICP three-dimensional point cloud registration based on K-D tree optimization," *Eng. Surveying Mapping*, vol. 25, no. 6, pp. 15–18, 2016.
- [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](https://doi.org/10.1109/ACCESS.2020.2976847).
- [32] A. S. Jackson, A. Bulat, V. Argyriou, and G. Tzimiropoulos, "Large pose 3D face reconstruction from a single image via direct volumetric CNN regression," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1–1031039, doi: [10.1109/iccv.2017.117](https://doi.org/10.1109/iccv.2017.117).
- [33] J. Huang, Y. Tong, K. Zhou, H. Bao, and M. Desbrun, "Interactive shape interpolation through controllable dynamic deformation," *IEEE Trans. Vis. Comput. Graphics*, vol. 17, no. 7, pp. 983–992, Jul. 2011.
- [34] K. C. Lam, X. Gu, and L. M. Lui, "Genus-one surface registration via Teichmüller extremal mapping," in *Medical Image Computing and Computer-Assisted Intervention—MICCAI 2014 (Lecture Notes in Computer Science)*, vol. 8675, P. Golland, N. Hata, and C. Barillot, Berlin, Germany: Springer-Verlag, 2014, pp. 25–32.
- [35] B. Li, A. Godil, M. Aono, X. Bai, T. Furuya, L. Li, R. López-Sastre, H. Johan, R. Ohbuchi, and C. R. Cabrera, "SHREC'12 track: Generic 3D shape retrieval," in *Proc. Eurographics Workshop 3D Object Retr. (EG DOR)*, 2012, pp. 119–126, doi: [10.2312/3DOR/3DOR12/119-126](https://doi.org/10.2312/3DOR/3DOR12/119-126).
- [36] L. Yin, X. Wei, Y. Sun, J. Wang, and M. J. Rosato, "A 3D facial expression database for facial behavior research," in *Proc. 7th Int. Conf. Autom. Face Gesture Recognit. (FGR)*, Apr. 2006, pp. 211–216, doi: [10.1109/FGR.2006.6](https://doi.org/10.1109/FGR.2006.6).
- [37] H. Zhao, *MeshDGP: Mesh Processing Framework [EB/OL]*. Accessed: May 8, 2019. [Online]. Available: <http://meshdgp.github.io/MeshDGP/MeshDGP.html>



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



LIGANG HE (Member, IEEE) is currently a Reader with the Department of Computer, 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 focus on parallel and distributed processing, and big data processing.



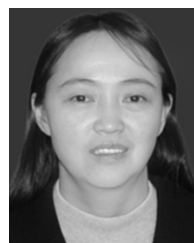
LIQUN KUANG was born in China, in 1976. He received the master's degree in computer application technology and the Ph.D. degree in modeling and simulation of complex systems from the North University of China (NUC), China, in 2003 and 2019, respectively. He is currently an Associate Professor with the Department of Computer Science and Technology, NUC. His research interests include brain imaging, computer vision and graphics, and virtual reality.



MIN CHANG received the bachelor's degree in computer science and technology from Taiyuan Normal University and the master's degree in computer science and technology from the North University of China. She is currently working with Jinxi Industrial Group Company Ltd. Her current research interests include computer vision, simulation, and visualization.



ZHIYING HE was born in China, in 1978. He received the master's degree in computer application technology from the North University of China, China, in 2009. He 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.