

A framework for automatic and perceptually valid facial animation generation

Hui Yu, Oliver Garrod, Rachael Jack, Philippe Schyns

Abstract Facial expressions are facial movements reflecting the internal emotional states of a character or in response to social communications. Realistic facial animation should consider at least two factors: believable facial expression visual effect and valid facial movements. However, most research tends to separate these two issues. In this paper, we present a framework for generating facial animations considering both visual effect of facial expression and the dynamic factor. A facial expression mapping approach based on local geometry encoding is proposed, which encodes deformation in the 1-ring vector. This method is capable of mapping subtle facial movements without considering those shape and topological constraints. Facial expression mapping is achieved through three steps - correspondence establishment, deviation transfer and movement mapping. Deviation is transferred to the conformal face space through minimizing the error function formed by the source neutral and the deformed face model related by those transformation matrices in 1-ring neighborhood. The transformation matrix in 1-ring neighborhood is independent of the face shape and mesh topology. Facial dynamics is then integrated with facial expressions for generating valid facial expressions. The dynamic factor was solved based on psychophysical methods. An application of the framework using various faces with different shapes and topological representations has been tested.

Keywords Facial Expression Mapping · Facial Animation · FACS · Psychophysical · Perceptually valid

H. Yu
University of Portsmouth, Portsmouth, UK
e-mail: hui.yu@port.ac.uk

O. Garrod · R. Jack · P. Schyns
University of Glasgow, Glasgow, UK

1 Introduction

Facial animation has been the interest in computer graphics and animation communities [1], [2]. A great deal of effort has been taken for generating realistic facial animation. Since the performance-driven animation method was introduced by Williams [11], it has been widely investigated [3], [4], [5], [6], [8], [9], [10], especially with the development of advances in 3D scanning and motion capture technologies [12], [37]. Readers are referred to the book by Parke and Waters [7] for an excellent survey about facial animation.

Due to the complexity of facial muscles and subtle movements involved in conveying emotional information, it is nontrivial to model realistic facial expression using muscle simulation or modeling methods [17], [42], [46]. Thus, how to efficiently reuse existing facial expression data to animate various new face models but avoid time-consuming and tedious manual intervention has been a challenging topic. To obtain various realistic facial expressions representing true muscle movements, we can simply make use of advanced scanning equipment, but it is impossible to capture all needed facial expressions for each subject we want. Another problem is about the quality of the facial animation. Since human beings are very sensitive to even subtle facial movements, it is critical to generate valid dynamic parameters for animation.

In this paper, we present a framework for generating facial animations considering both visual effect of facial expression and the dynamic factor. A facial expression mapping method has been proposed, which transfer facial expressions from a source model to the target 3D face through encoding deformation information in local geometric shape. Through recording FACS AU from real human beings and mapping to different face models, we demonstrate that the proposed method can easily generate facial animation reflecting true muscle movements given any 3D characters.

The proposed method is a semi-automatic with the aid of a few feature points manually labeled by the user. Fig. 1 demonstrates the template face model and manually labeled. It is capable of mapping facial expressions while keeping the characteristics of the target face even the target face shape is quite different from the source human face shape. Another advantage of the proposed method is that it is independent of the source and target mesh topology and easy to be implemented. We have tested the feasibility and validity of the proposed method on different face models with various sizes, shapes and mesh topologies.

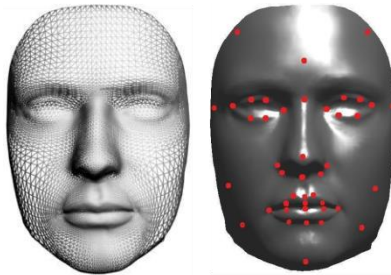


Fig. 1 Illustration of 3D mesh template and markers

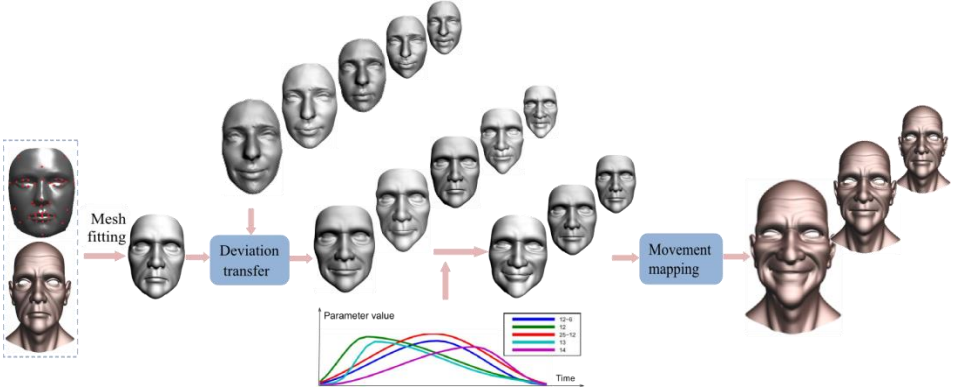


Fig. 2 Flowchart of facial expression mapping and animation for the expression of “happiness”. Five AUs – AU12-6, AU12, AU12-25, AU13 and AU14 are mapped to the conformed target model from the source face through deviation transfer to generate the happiness facial expression. Animation parameter curves are then applied to those two AUs for generating animation.

The remainder of the paper is organized as follows. We first review related work in Section 2. We then describe the correspondence establishment method in Section 3. And Section 4 details the deviation transfer method. Section 5 presents the movement and texture mapping procedure. Section 6 describes the animation application framework combining the facial expression transfer method. In Section 7, experiment results are presented. Finally, the paper is concluded in Section 7 with a short description of future work.

2 Related work

Since the pioneer work conducted by Parke [28], research on facial animation has been widely studied. Enormous research has been conducted on improving fidelity of facial expression as well as on reducing time-consuming and tedious intervention from users.

Facial expressions are actually driven by facial muscles beneath the skin. Physically-based methods try to simulate facial muscles and skin to animate face models [15], [16], [21], [44]. Those methods require high computation and nontrivial skills to control the parameters simulating muscle forces.

Geometry-based methods tend to animate face models using geometric information involving using feature points or controlling geometric parameters [5]. Zhang et al. [26] developed a geometry-driven method for facial expression synthesis using feature points and face region subdivision, which was capable to infer missing feature motions to generate expression detail. Pighin et al. [25] proposed an image-based method for generating photo-realistic 3D facial expressions from 2D images. It adopted a scattered data interpolation function, Radial Basis Function (RBF), to interpolate facial vertices. Facial expression of the source face is mapped to the target face model through transferring those obtained combinational coefficients. Pei et al. proposed a system transferring speech movement from video to 3D faces based on the assumed 2D and 3D viseme mapping [50]. Parke et al. [45] presented a parametric method for generating facial expressions using parameter vector as representation of the motion of a set of vertices. However, it is still time-consuming for reusing those two methods in application where various new faces are to be animated.

Noh et al [8] introduced a method for reusing animation data for motion retargeting. Movements of the source face are represented by motion vectors which were then transferred to the target face model. That method only works well for cloning facial expressions for highly resemble face shapes. Summer et al. [27] proposed a method for deformation transfer from the source mesh model to the target mesh based on triangle transformation. This method has limitations if the source and target meshes have different mesh topology or less shape resemblance. Darren et al [40] described a method for performance driven facial animation by re-mapping animation parameters between multiple types of facial models, which was able to map facial expressions through extracting meaningful facial actions parameters from video performance. Curio et al [5] presented a system for 3D motion retargeting combining a laser scanner and motion capture system. Their system produced a set of vectors of morph weights using optimization system applied on motion capture markers. Asthana et al [19]. proposed a real-time facial performance transfer method through modeling parametric correspondence for 2D images.

There has been some effort on mapping facial performance from the source face model to the target model in which the motion data is from motion capture systems or 3D scanners [14], [41], [43]. These methods either transfer the source deformation or transfer the blending weightings of the key source shapes to the target model.

FACS as one of the most popular theories and techniques for describing facial activities was created originally for the purpose of psychological research, but it began to draw attention by computer graphics community and entertainment industry in recent years [22]. FACS defines 46 units of primitive facial movements and poses called Action Unit (AU). Through using blendshape method to combine relevant AUs, various facial expressions can be obtained. Another method is Facial animation parameter (FAP), which is an extension of FACS. FAP defines 68 feature points for facial expression synthesis in the MPEG-4 standard [23], which are closely related to muscle actions. Chin et al [48] proposed an intensity-based facial expression cloning method for low-polygon-based application. Facial expression intensities are explicitly measured and represented by a set of parameters, which are then mapped to the target face using three mapping functions. Our method does not explicitly measure facial expression intensity but use local geometric information to represent the expression.

Most methods mentioned above have restriction on mapping facial motions between faces with less resemblance. Moreover, the issue of validity of the facial animation has not been usually taken into account. In this paper, we proposed a framework for generating facial animation considering both issues above.

3 Static facial expression mapping

As human beings are very sensitive to even subtle changes on facial expressions, any fake movements on faces would cause an Uncanny Valley effect. To overcome this problem from the source root, various techniques can be used to capture facial movements from real human beings [30, 50]. After recording the facial movements, the next questions are how to adapt these movements to different identities; how to adapt the limited facial movements to various facial expressions. We solve the first question by developing a facial expression method. And for the second question, we adopt the FACS method. The recorded movements are represented by 44 FACS AUs. In this section, we will describe an approach on how to map the recorded facial expression to different identities with various face shapes and topologies.

3.1 Correspondence establishment

Due to variable number of vertices and mesh types across different individuals, we need to establish correspondence between the source face and the target face model. Since it is a nonlinear problem to fit a template mesh to a target model data with different vertex number and topologies, the optimization could be trapped in local minima leading to fake mesh fitting. Recently, Sun et al [32] proposed a method for establishing vertex correspondences using a tracking-model-based approach and coarse-to-fine adaptation, which was used for analyzing dynamic 3D face model sequences. To avoid the local minima problem, we also adopt a coarse-to-fine strategy. A low resolution template model is fitted to the target model in the first place. Then, a higher resolution template, which is achieved through subdivision, is applied for final fitting. Then, the fitted intermediate face model is used to map the movements to the target model by projecting the target face model onto the deformed template mesh, so each projection point can be located according to the triangle it falls in on the deformed template mesh. The flowchart of the facial expression mapping and animation is illustrated in Fig. 2.

Mesh registration transforms the template mesh model to align with the target face model. The registration process consists of two stages, i.e., rigid registration and nonrigid registration.

In the rigid registration stage, the rigid transformation T is computed, which consists of rigid translation, rotation and scale. This is achieved by using the iterative scaling closest point algorithm (SICP) [51]. Since this is a rough alignment, the computation is only applied to landmarks. The template mesh model is then transformed to align with the target model. Apart from rough alignment, the rigid registration speeds up the whole registration procedure.

In the nonregistration stage, the transformed template face model is iteratively deformed. Nonrigid transformation is computed for across all vertices on both models. In iteration, the template mesh searches and deforms to approximate the closest vertices on the target model through minimizing a total energy consisting of three energy terms.

The first energy term E_c represents resemble-point energy which moves each vertex of the template mesh to the most resemble vertex on the target mesh. We define the resemble points based on geometric features of the mesh model, which is represented by Gaussian curvature at each vertex. In the case of the point cloud model, Gaussian curvature is calculated using the method [24].

$$E_c = \sum_{i=1}^n \|s_i - t_i\|^2 \quad (1)$$

where s_i is the most resemble point on the template mesh to the vertex t_i on the target mode.

The smoothness term E_s is used to keep the original topological relationship in one-ring neighborhood.

$$E_s = \sum_{i=1}^n \sum_{j \in N(i)} (\|v_j - v_i\| - \|v_j^o - v_i^o\|)^2 \quad (2)$$

where v_j is one-ring neighbouring vertices of vertex v_i on the deformed template mesh.

Vertex v^o is the template vertices before deformed.

To assist the registration procedure, we manually label 44 landmarks pairs on both template mesh and the target mesh. Those landmarks are deformation constraints in iteration.

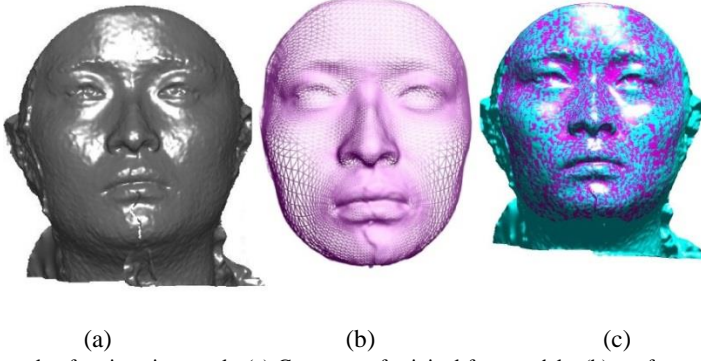


Fig. 3. An example of registration result. (a) Geometry of original face models; (b) conformed mesh models; (c) overlap of conformed face models on original models illustrating the error of registration.

The third item is the landmark constraint E_m . Thus the deformation procedure can be defined as follows:

$$\arg \min_{\tilde{v}_1, \tilde{v}_2, \dots, \tilde{v}_n} (\alpha E_c + \beta E_s + \gamma E_m) \quad (3)$$

To accelerate the registration and improve the fitting, we employ a coarse-to-fine strategy. A coarse template face model with 1200 vertices is used first for the registration. And then a dense mesh of the same template model with 4735 vertices is applied after the convergence of the registration of the coarse model. One result of registration example is demonstrated in Fig. 3.

3.2 Deformation transfer

Sorkine et al. [38] proposed a method for Laplacian surface editing, which is invariant to rotation and scaling. The sensitivity to linear transformation of Laplacian coordinate is overcome by applying a transformation T_i for each vertex v_i .

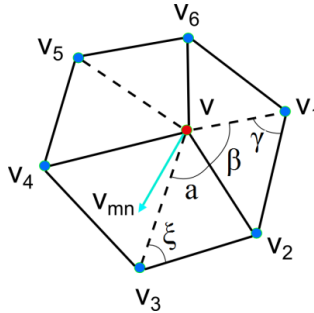


Fig. 4. Illustration of 1-ring vector

As a consequence, $T_i(V')$ is a function of V' , so the reconstructed shape is obtained by minimizing the following error function:

$$E(\tilde{V}) = \sum_{i=1}^n \|T_i(\tilde{V})\delta_i - L(\tilde{v}_i)\|^2 + \sum \| \tilde{v}_i - u_i \|^2 \quad (4)$$

Inspired by the Laplacian coordinate method, we propose an approach based on local geometry encoding, which is based on the relationship of 1-ring vectors between the original shape and the deformed shape. Laplacian coordinate encodes vertex v_i in 1-ring neighbours to one vector δ_i , while we try to explicitly use these 1-ring neighbours. Before we detail the proposed method, let us first have a brief review of Laplacian coordinates.

Suppose the mesh is represented by a graph $G=(V,E)$, where $V=[v_1, v_2, \dots, v_n]$ is the set of vertices and E is the edges. And v_i denotes vertex i in \mathfrak{R}^3 and δ_i is the Laplacian of v_i . Laplacian coordinates is assumed as discretization of continuous Laplace-Beltrami operator [13]. Laplacian coordinates are defined as the difference between vertex v_i and its 1-ring neighbours.

$$\delta_i = (\delta_{ix}, \delta_{iy}, \delta_{iz})^T = \sum_{j \in N(i)} w_{ij} (v_i - v_j) \quad (5)$$

where w_{ij} is the weight on the edge e_{ij} with $\sum_{j \in N(i)} w_{ij} = 1$ and w_{ij} can be computed as follows:

$$w_{ij} = \frac{a_{ij}}{\sum_{j \in N(i)} a_{ij}} \quad (6)$$

Several methods for calculating edge weight have been proposed in recent years. Here is a brief introduction to three commonly-used weights:

$$a_{ij} = 1 \quad (7)$$

$$a_{ij} = \cot \alpha + \cot \beta \quad (8)$$

$$a_{ij} = \frac{\tan(\gamma/2) + \tan(\xi/2)}{\|v_i - v_j\|} \quad (9)$$

where equation (7) are uniform, (8) the cotangent weight, and equation (9) the mean-value weight. Fig. 4 demonstrates these angles used in above equations.

Laplacian operators described above encode geometric information the mesh, so mesh detail is preserved in deformation using Laplacian coordinates to some degree. Taubin proposed that the length of the Laplacian vector is equal to the product of the average edge length times the mean curvature [36]:

$$\delta_i = \sum_{j \in N(i)} w_{ij} (v_i - v_j) \bullet k_i N_i \quad (10)$$

where k_i is the mean curvature of the surface at vertex v_i and N_i is the surface normal at v_i . More specifically, Meyer et al. [35] proposed to approximate the mean curvature normal vector using Laplacian coordinates:

$$\begin{aligned} k_i N_i &= \delta_i \\ &= \frac{1}{4|A_i|} \sum_{j \in N(i)} (\cot \alpha_{ij} + \cot \beta_{ij}) (v_i - v_j) \end{aligned} \quad (11)$$

where $|A_i|$ is the 1-ring area of the Voronoi region around vertex v_i , and α_i, β_i are the two angles sharing the edge $E(i, j)$ illustrated in Fig. 4.

Therefore, mean curvature normal as an approximation of Laplacian coordinates in 1-ring neighbourhood has the similar characteristic of capturing the local geometric information with Laplacian coordinates, which is demonstrated in Fig. 5.

In this paper, we propose to use mean curvature normal as the extra vector of 1-ring vector for encoding the local deformation information which is to be introduced as follows.

We use 1-ring vector to capture local mesh detail. Let V be 1-ring vector, $V = [v_i - v_{i1}, v_i - v_{i2}, \dots, v_i - v_{im}]$, where $v_{i1}, v_{i2}, \dots, v_{im}$ are 1-ring neighbours of v_i . We further take the mean curvature into account in 1-ring vector to capture more local surface information in 1-ring neighbourhood. Therefore 1-ring vector $V = [v_i - v_{i1}, v_i - v_{i2}, \dots, v_i - v_{im}, v_n]$, where v_n is the mean curvature normal vector at vertex v_i .

We suppose that there exists a transformation matrix T_i , which relates the 1-ring vector V of the original shape to the deformed shape. Thus the error function can be formulated as follows:

$$E(\tilde{V}) = \sum_{i=1}^n \|T_i(V_i) - \tilde{V}_i\|^2 \quad (12)$$

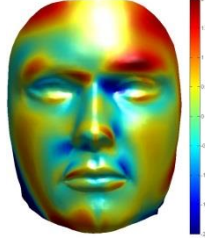


Fig. 5 Color coded face of mean curvature normal.

Where \tilde{V}_i is the 1-ring vector of \tilde{V}_i on the deformed shape. Specifically, for vertex v_i , this relationship can be expressed as below:

$$T V = \tilde{V} \quad (13)$$

We notice that in (12), for a given shape M and its deformed shape \tilde{M} and 1-ring vector \tilde{V}_i we can simply work out the transformation T for vertex v_i :

$$T = \tilde{V} V' (V V')^{-1} \quad (14)$$

Specifically, our goal is to transfer this deviation from the source face to a different neutral face model. The basic idea is that we expect the transformation matrix T to encode the deformation information for each 1-ring vector, which is transferable to any face models. Then, we can apply transformation matrix T to the target face model to obtain the deformation. Ideally, by applying the transformation matrix T to the neutral face model, we can obtain its deformed expression model. Thus, the minimization of error function in (12) can be reformulated as follows:

$$\arg \min_{T_i} (\|T_i V_i^t - \tilde{V}_i^t\|^2) \quad (15)$$

where V_i^t and \tilde{V}_i^t are the 1-ring vector for the target neutral and unknown deformed face model.

To prevent the edge length of 1-ring neighbourhood from changing in any cases, we apply the following edge energy term presented in [31]:

$$\sum \| (v_i - v_j) - r(v_i - v_j) \|^2 \quad (16)$$

$$r(v_i - v_j) = \frac{l_{ij}}{l_{ij}^o} (v_i - v_j)$$

Where $l_{i,j}^o$ and l_{ij} are the original and the current length of edge e_{ij} respectively. The above

equation can be reformulated as matrix form:

$$\|HV - r(v)\|^2 \quad (17)$$

where H is nx3n sparse matrix.

Thus, deformation can be achieved by minimizing the following energy:

$$E = \|T_i V_i^t - \tilde{V}_i^t\|^2 + \|HV - r(V)\|^2 \quad (18)$$

The solution to the minimization problem of the above equation is equivalent to solving a system of linear equation, which can be represented as the following form:

$$\arg \min_{\tilde{v}_1, \dots, \tilde{v}_n} \|A\tilde{X} - b\|_2^2 \quad (19)$$

where $\tilde{X} = [\tilde{v}_1, \tilde{v}_2, \dots, \tilde{v}_n, \tilde{v}_{m1}, \tilde{v}_{m2}, \dots, \tilde{v}_{mn}]'$ is a vector consisting of $3xn$ unknown entries of the deformed face model: vertex \tilde{v}_i and mean curvature normal vector \tilde{v}_{mi} . And b contains information from the source deformation and given target neutral face vertices. Each entry of b is computed as follows:

$$b_i = [T_i^t T_i]^{-1} T_i^t V. \quad (20)$$

The dimension of large and sparse matrix A is $[3xn, 3xn]$, which associates unknown \tilde{X} with b. The linear system can be solved using the normal equation:

$$\tilde{X} = (A^t A)^{-1} A^t b. \quad (21)$$

It should be noted that by solving (18), we obtain both vertex and mean curvature normal of the deformed face model. The first n entries of \tilde{X} correspond to vertices of the new transformed face model. Since the model is in 3D space, each entry actually consists of 3 components (x, y, z).

It is expensive to solve the linear system directly. It takes more time to solve with a dense model. Therefore, it is more efficient to decompose $A^t A$ by LU factorization in advance for computation. Then component x, y and z are solved separately in (21), whereas we only need to decompose $A^t A$ once. Fig. 6 demonstrates the examples of facial expression mapping.

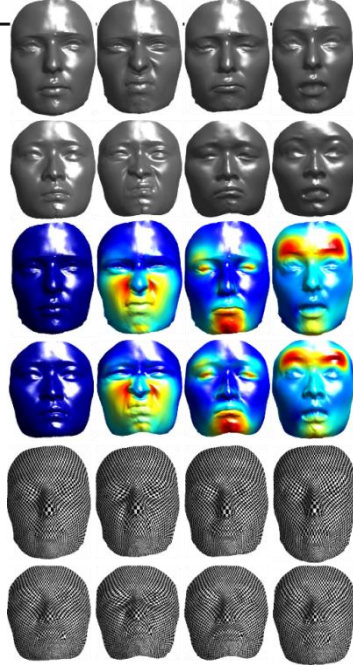


Fig. 6. Facial expressions are transferred from the source face models (row 1) to the target face model (row 2). Column 1 is neutral face model. Expressions from column 2 to 4 are disgusting, sad and surprise. Row 3 and row 4 are color map of source and target facial expressions, where cold-color and hot-color indicate minimum deviation and maximum deviation for each individual face deformation. Row 5 and 6 illustrate the check board mapping faces showing the deformation pattern of facial expression.

4 Movement and texture mapping

4.1. Movement mapping

After deforming the template mesh to the target face model, the deformed template mesh is nicely fitted to the target face. We then parameterize the target face in the space of the source model. More precisely, we concern only those vertices falling in the surface of the source model, since those vertices are possible to move driven by the conformed template mesh. Vertices outside the range of conformed template mesh will not be considered for nonrigid deformation in this paper. This is actually not the truth in reality. Since in some cases, when human beings make facial expressions, there are movements in the neck area. This happens especially when facial expressions involve mouth movements.

In this paper, we focus on the front view facial expression and animation only on the face area, so the template face model mainly covers the front view. There might be slightly unrealistic around the neck area when generating facial animation. We have removed the rigid head motions from the source model to reduce the effect in the neck area. Here, two

steps are conducted to achieve this movement mapping.

- Step 1: vertex projection

It is used to find those vertices of the target model which falls into each triangle of the source face model. To this end, we project vertices of the target face model onto the deformed template mesh along their normal direction. By using normal projection, we can avoid missing those vertices on the deformed template which do not exactly lie in the triangle of the target mesh due to registration errors.

- Step 2: barycentric coordinate calculation

We calculate the barycentric coordinates of those vertices falling in each triangle. Then we can obtain the movement of the vertices falling in triangles based on their barycentric coordinates. Suppose projection point v_p corresponding to vertex p of the target model falls in triangle t of the deformed template mesh, so the deviation of each vertex p can be obtained using its barycentric coordinates. Assume deviation at triangle t (v_1, v_2, v_3) is (d_1, d_2, d_3) . Then deviation d_p of point v_p is calculated using the following equation:

$$d_p = d_1 u_p + d_2 v_p + d_3 w_p \quad (22)$$

where (u_p, v_p, w_p) is barycentric coordinate of v_p with respect to (v_1, v_2, v_3) .

Fig. 7 shows examples of the movement mapping result for a face model of point cloud data. The point cloud human face has not been preprocessed. It contains noises and even there is a small but clearly noticeable crevice on the lower jaw. We have generated six primary facial expressions for this point cloud data using the methods introduced above. Since we only captured 3D surface from a real human being, there is no opening on the mouth area. When mapping facial movements to this point cloud data for some facial expression eg. angry and fear etc. involving mouth opening, those points in this area will not dealt. This problem could be solved by manual creating an opening in the mouth area and rigging the head with teeth etc.

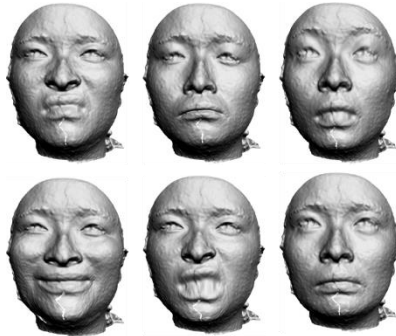


Fig. 7 Illustration of movement mapping result of female face (point cloud). Top row: expression of disgust, sad and surprise corresponding to those conformed expression in Fig. 6. Bottom row: three facial expression happiness, anger and fear.

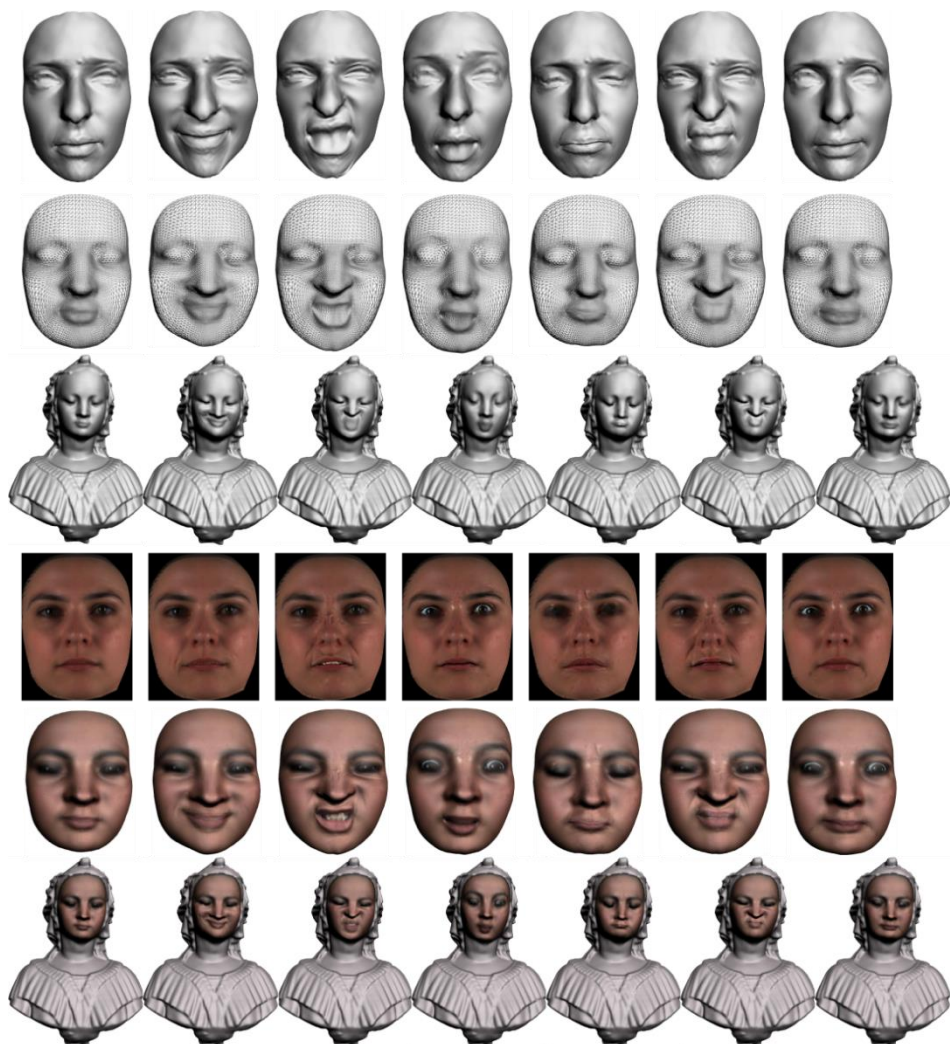


Fig. 8. Result of facial expression mapping. Columns from left to right are neutral face and six primary facial expressions for the source and the target face: happy, anger, surprise, sad, disgust and fear. Top row is the source face; conformed target face (row 2); Row 3 is the target model, where the left column is the original neutral face model and the rest are six facial expression model mapped from the source facial expression. Row 4 is textures. Row 5 is the conformed target faces with textures and bottom row is the textured target model. The bust model contains 47516 vertices and 95028 faces. The height of the bust model is 100 mm, which makes the face about 30mm, whereas the length of the source neutral face model in the first two rows is about 85 mm.



Fig. 9. Example AUs of the target old man face model transferred from the recorded AUs of certified FACS actors. Top row is the neutral face. Row 2 from left to right: AU4, AU6L, AU9, and AU10LOpen; Row 3: AU11L, AU11R, AU12 and AU12L. Row 4: AU12R, AU14, AU14L and AU14R.



Fig. 10. Illustration of recorded AU examples. Top row: AU2, AU4, AU5, AU12; Bottom row: AU20, AU15, AU25 and AU43.

4.2 Texture mapping

Our face model consists of both geometric structure and texture information. The texture is mapped to the target face model through two steps:

- Map the texture to the conformed face model

During the process of reconstruction of 3D face data, the 3D coordinates of each pixel of the face image are obtained from epipolar geometry information of stereo vision [20, 34].

The 3D coordinates corresponding to the template mesh are sampled. An UV is then constructed by projecting those 3D sampled points back to 2D image space based on the obtained camera parameters.

- Generate texture on the target face model

Barycentric coordinates of projection vertices of the target model have already been calculated using the method in section 5.1. Each projection vertices has a corresponding pixel on the 2D image, which can be sampled using its barycentric coordinate:

$$I_p = I_1 u_p + I_2 v_p + I_3 w_p \quad (23)$$

where (I_1, I_2, I_3) is the RGB value of the pixel corresponding to triangle (v_1, v_2, v_3) . We compare the texture sampled using barycentric coordinates with the one using bilinear interpolation.

5 Animation generation

In this paper, we propose to integrate dynamic parameters for generating valid facial animation. We consider these data as “valid” since they were generated by using psychophysical methods. We want to make sure that the messages that the synthesized facial animation intended to deliver can be precisely perceived the same by the viewers. To this end, we used the reverse correlation method from observer response to determine the dynamic parameters. We generated random facial animation stimuli using the 3D Morphable Model method based on Facial Expression Coding System (FACS) [29]. We tend to automatically generate facial animation for the target face model reflecting the true muscle movements, which were recorded from real human beings.

5.1 Create FACS database

We used FACS AUs as the key shapes for expression generation. Advanced technologies make it possible to reconstruct or capture 3D shapes [30, 33]. We have created a database of FACS Action Unit (AU) by recording AUs from certified actors using a 4D stereo imaging system [30].

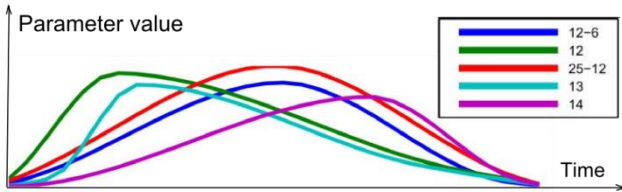


Fig. 11. Interpolation Curve for dynamic happy facial animation

Fig. 10 demonstrates some example images of those recorded AUs. Those certified actors must learn not only to identify FACS codes in static face images, but also be able to produce the FACS codes themselves in terms of FACS. Each actor in our database was capable of performing most AUs (between 20 and 30). We instructed actors to perform only those FACS AUs that they were confident about to produce in isolation from others. Those recorded AUs are among the most often used in generating facial expressions. During the recording, we instructed actors to produce each AU with smoothly progress starting from a neutral face through to full AU activation, and then moving back to the neutral. To assist post-processing the recorded data, we asked each actor to wear a black cap to hide the hair and a small white head-mounted board marked with nine black dots.

After data post-processing we obtained a sequence of 3D face models for each AU with the same mesh topology. We extracted the deviation of each AU at peak frame based on the neutral frame. Then we obtained a set of AU deviations, which are transferred to the conformed model of the target face using the proposed deviation transfer method. Therefore, a set of AU models based on the target model are generated. Some example AUs of the old man model are shown in Fig. 9.

5.2 Dynamic parameters

To animate the FACS 3D Morphable Model, we generated random unimodal temporal morph functions per FACS AU. Those morph functions is formed by 6 parameters including “Onset Latency”, “Offset Latency”, “Acceleration”, “Deceleration”, “Peak Latency” and “Peak Amp”. They served as the source dynamic parameters for the stimuli, which was presented as 30 time frames facial animation video with 24 frames per second. Basically, we generated 2400 random facial animations used this method. We then instructed 8 novel western Caucasian observers to categorize these facial animation in terms of 7 Alternative Forced-Choice into 6 primary facial expression of emotion based on FACS. These facial expression categories included “Happiness”, “Surprise”, “Fear”, “Disgust”, “Anger” and “Sadness”. We also provided a response option “Other” for those observers for the cases that they thought no categories the observed facial expression belonging to. Observers rated the perceived emotional intensity of each stimulus using a 5 rating scale. When performing a linear regression between the intensity ratings for the emotion and each of the 6 random temporal parameters for those AUs, we were able to obtain the temporal curve for each facial expression in the FACS space. Since not all of those involved AUs activated and stopped at the same time, each individual AU has a separate temporal curve. Fig. 11 illustrates a set of parameter curves for generating animation of happiness expression. Readers are referred to the paper by Yu etc. [47] for the detailed experiment setup for generating the dynamic parameters.

General speaking, after obtaining static facial expressions for the given face and dynamic parameters, facial animation can be achieved using two methods: the direct animation sequences mapping or the indirect blendshape method. For the direct method, the mapping algorithm is applied to every frame of source face model to get the sequence of the sequence of the target face model. For the indirect method, we first obtain the set of blendshapes of the

Table 1 Various model information.

TABLE I MODEL INFORMATION			
Model Name	Mesh Type	Vertex number	Face number
Male head	Triangular mesh	33234	66160
Female bust	Triangular mesh	47516	95028
Ogre	Triangular mesh	11139	22058
Female head	Point cloud	49953	--
Female elf	Quad mesh	2551	4412
Mount Rushmore	Quad mesh	20600	40904

conformed model of the target face model by mapping the set of source blendshapes to the conformed face model. Then intermediate facial expression model of the conformed model is formed by combining those blendshapes of the conformed target model. Finally, the new facial expression is generated by movement mapping from intermediate expression model. We have generated animation of six facial expressions for each face of Mount Rushmore data. The Mount Rushmore data figure and animation video can be found in the supplementary material related to this paper. The animation is created in 30 frames per second. To illustrate the facial animation effect, only one face is animated in each frame.

6 Experiments

We have tested the proposed method on various types of models including triangular mesh, quadrilateral mesh model, point cloud model, dense mesh model and sparse mesh model. Table 1 presents the detail of some example models shown in this paper. Fig. 12 shows some results of face model old man, ogre and female elf with decreasing vertex numbers. Each facial expression is shown at its peak frame. Four faces in mount rushmore data are animated consecutively. The female elf model is sparse polygon model. After facial expression mapping, the result still shows clear deformation detail around lips and eyes (bottom row in fig. 12). Those face model have various face shapes and topologies but could be animated easily with the proposed framework.

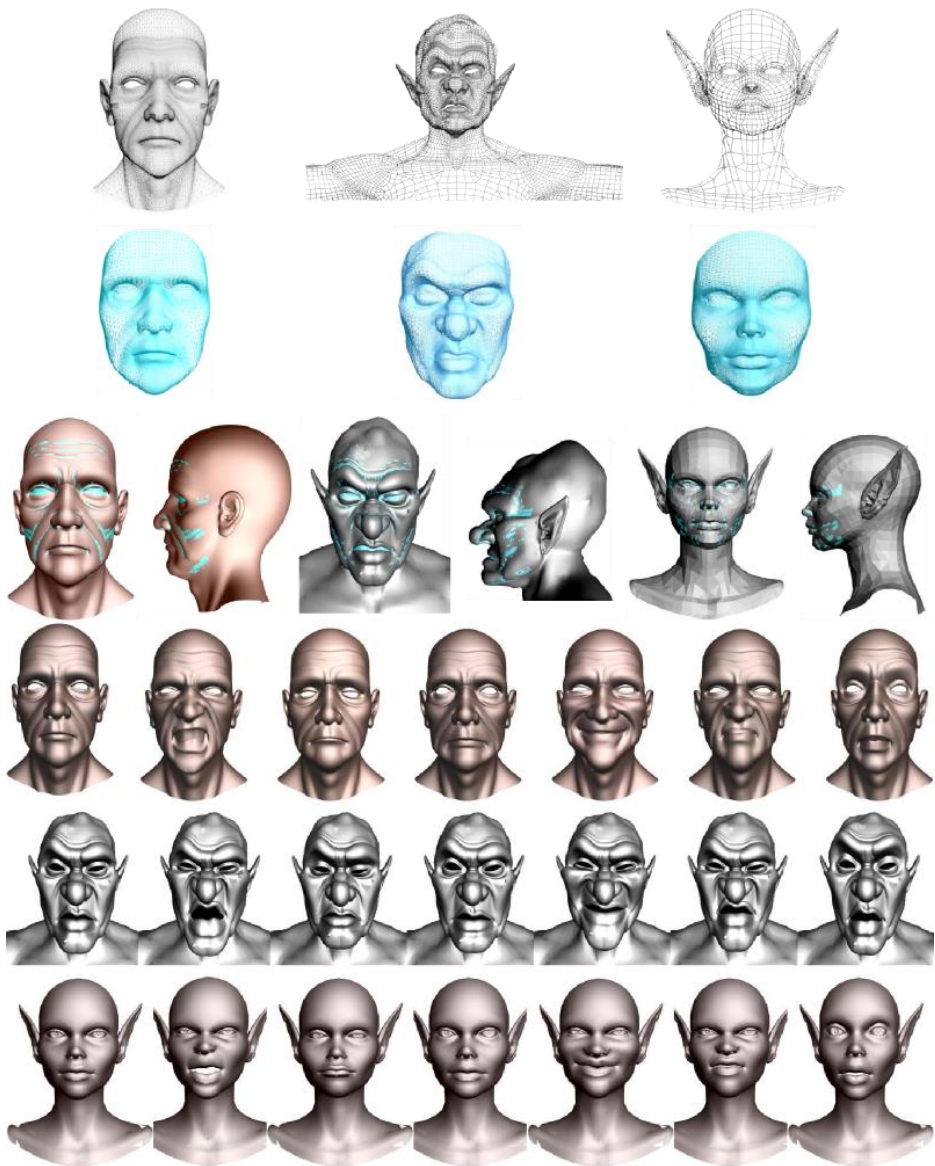


Fig. 12. The top row and second row show three wireframe models and their corresponding conformed mesh: ogre (column one), female elf (column two) and male head (column three). Row three demonstrates fitting error distribution, where different color represents the fitting difference. From row four to row 2 to the bottom row are the results of six facial expression mapping for those three models in row 1, where column one are the neutral expression; from column two to column 7 are six facial expressions: anger, sadness, fear, happiness, disgust and surprise.

7 Conclusion and future work

We have presented a 3D facial animation framework for automatic facial animation generation. It included methods for automatic static facial expression mapping and valid dynamic facial parameter integration. The facial expression deviation is transferred based on local encoded geometric information of face models. The advantage of the proposed method enables us to Map facial movements to facial models regardless of their mesh topologies triangular or quadrilateral mesh. We have demonstrated that the proposed method works for different types of face models, such as point cloud model, triangular face model and quadrilateral mesh models. Apart from manually labeling a few feature landmarks on the face, all the rest procedure is automatic, which enables the method work very efficient.

Though the manual labeled markers on the faces play an important role in fitting especially at the early stage of optimization, the whole optimization procedure of the fitting balances the importance of those marks and the actual shape. Therefore, the quality of the fitting depends on many factors including the complexity of the target face, point density and marker accuracy etc. In the future, to improve the current method, we would like to make it as a fully automatic procedure by developing an automatic and precise feature point detection method on 3D faces [52]. The parameters generated by the unimodal method have demonstrated a pleasant effect, but it can be used for generating only a single modal animation. In future, we could still need to explore some more flexible methods for generating multimodal dynamic facial parameters.

Acknowledgments

References

1. Ersotelos N., Dong F., Building Highly Realistic Facial Animation: A Survey, *The Visual Computer*, Vol.24, Iss.1, Pp.13,2008
2. Deng Z. and Noh J. Y. , Computer Facial Animation: A Survey, , Pages 1-28, *Springer Press, Data-Driven 3d Facial Animation*, 2007
3. Zhang L., Snavely N., Curless B., Seitz S.M., Spacetime Faces: High Resolution Capture For Modeling And Animation. *ACM Transactions On Graphics* 23, 3 548-558,2004
4. Ma J., Cole R., Pellom B., Ward, W., And Wise, B. 2004. Accurate Automatic Visible Speech Synthesis Of Arbitrary 3d Model Based On Concatenation Of Diviseme Motion Capture Data. *Computer Animation And Virtual Worlds* 15, 1–17.
5. Curio C., Breidt M., Kleiner M., Vuong Q., Giess M., Bulthoff H.. Semantic 3D Motion Retargeting for Facial Animation.. In: Proceedings of the ACMAPGV; 2007. p.59–64.
6. Zalewski L., Gong S.: ‘2D Statistical Models Of Facial Expressions For Realistic 3d Avatar Animation’. *Proc. Ieee Computer Vision And Pattern Recognition*, 2005, Vol. 2, Pp. 217–222
7. Parke F. I. And Waters K.. *Computer Facial Animation*. A K Peters, Wellesley, Massachusetts, 1996.
8. Noh J., Neumann U.: ‘Expression Cloning’. *Proc. Siggraph*, 2001, Pp. 277–288
9. Blanz V., Basso C., Poggio T., Vetter T.: ‘Reanimating Faces In Images And Video’. *Proc. Eurographics*, 2003, Pp. 641–650
10. Zhang Q., Liu Z., Guo B., Terzopoulos D., Shum H.: ‘Geometry-Driven Photorealistic Facial Expression Synthesis’, *IEEE Trans. Vis. Comput. Graphics*, 2006, 12, (1), Pp. 48–60
11. Williams L., 1990. Performance-Driven Facial Animation. In *Siggraph '90, Acm Press*, 235–242.
12. Zhang L., Snavely N., B. Curless And S. M. Seitz, Spacetime Faces: High Resolution Capture For Modeling And Animation, *Acm Transaction On Graphics*, Vol. 23 No.3 Pp.548-558, 2004
13. Do M. Carmo P.: *Differential Geometry Of Curves And Surfaces*. Prentice-Hall, 1976.

14. Gleicher M. "Retargetting Motion To New Characters", *Acm Siggraph 98 Conference Proceedings*, Pp. 33-42, 1998.
15. Kahler K. Haber J. H. Y.H.S. Head Shop: Generating Animated Head Models With Anatomical Structure. *Acm Siggraph/Eurographics Symposium On Computer Animation*, Pp.55-63
16. Sifakis E., A. Selle, A. Robinson-Mosher, R. Fedkiw. Simulating Speech With A Physics-Based Facial Muscle Model. *Eurographics* 2006, 261-270.
17. Kahler K. K., J. Haber, and H. P. Seidel, "Geometry-Based Muscle Modeling For Facial Animation," in *Proc. Graphics Interface*, 2001, pp. 37–46.
18. Kalra P., Mangili A., N. Magnenatthalmann, D. Thalmann: Simulation Of Facial Muscle Actions Based On Rational Free Form Deformations. In *Eurographics '92*, pp. 59–69, 1992.
19. Asthana A., M. de la Hunty, A. Dhall, & R. Goecke, Facial performance transfer via deformable models and parametric correspondence. *IEEE Transactions on Visualization and Computer Graphics*, 18(9), 1511-1519. 2012.
20. <http://opencv.org/>
21. Terzopoulos D. And K. Waters. Physically-Based Facial Modelling, Analysis, And Animation. *The Journal Of Visualization And Computer Animation*, 1:73–80, 1990.
22. <http://www.sonypictures.com/homevideo/monsterhouse/>
23. Gao W., Y.Chen, R.Wang, S. Shan, And D. Jiang, Learning And Synthesizing Mpeg-4 Compatible 3d Face Animation From Video Sequence. *Ieee Transactions On Circuits And Systems For Video Technology*. Vol. 13, No 11, Pp 1119-1128, 2003
24. Kalogerakis E., Simari P., E. Nowrouzezahrai And K. Singh. Robust Statistical Estimation Of Curvature On Discretized Surfaces *Proceedings Of The Fifth Eurographics Symposium On Geometry Processing* Pages 13-22, 2007
25. Pighin F., Hecker J., D. Lischinski, R. Szeliski, And D. H. Salesin, "Synthesizing Realistic Facial Expressions From Photographs," In *Siggraph'98: Proc. Of The 25th Annual Conference On Computer Graphics And Interactive Techniques*. New York, Ny, Usa: Acm Press, 1998, Pp.75–84.
26. Zhang Q., B. G. Z. Liu, And H. Shum, "Geometry-Driven Photorealistic Facial Expression Synthesis," In *Proc. Eurography/Siggraph Symposium On Computer Animation*, 2003, Pp. 177–187.
27. Sumner R., P. J. Deformation Transfer For Triangle Meshes. In: *Proceedings Of The Acm Siggraph 04; 2004*. P. 399–405.
28. Parke F., Computer Generated Animation of Faces, In *Acm National Conferences*, Vol. 1. Acm Press, 1972, Pp.451-457.
29. Ekman P., Friesen W.. Facial Action Coding System: A Technique For The Measurement Of Facial Movement. Palo Alto, Ca: Consulting Psychologists Press; 1978.
30. Rafael F.V. Saracchini, Jorge Stolfi, Helena C.G. Leitão, Gary A. Atkinson, Melvyn L. Smith Robust 3D face capture using example-based photometric stereo, *Computers in Industry*, Volume 64, Issue 9, December 2013, Pages 1399-1410
31. Weng Y., W. Xu, Y. Wu, K. Zhou, B. N. Guo. 2D Shape Deformation Using Nonlinear Least Squares Optimization *The Visual Computer*. Volume 22 Issue 9, September 2006
32. Sun Y., X. Chen ; M. Rosato, L. J. Yin. Tracking Vertex Flow and Model Adaptation for Three-Dimensional Spatiotemporal Face Analysis. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, Volume: 40 , Issue: 3. Page(s): 461- 474. 2010
33. Lee I., Mahmood M. T., Shim S. and Choi T., "Optimizing image focus for 3D shape recovery through genetic algorithm," *Multimedian Tools and Applications*, DOI: 10.1007/s11042-013-1433-9
34. Hartley R., A. Zisserman. Multiple View Geometry In Computer Vision. 2nd Ed. Cambridge University Press, 2004.
35. Meyer M., M. Desbrun., P. Schröder, A. H. Barr: Discrete Differential-Geometry Operators For Triangulated 2-Manifolds. In: *Proc. Of Visualization And Mathematics*, Pp. 35–57 (2003)
36. Taubin G.: Geometric Signal Processing On Polygonal Meshes. In: *Proc. Of Eurographics'2000: Star-State Of The Art Report* (2000)
37. Chai J., J. Xiao, And J. Hodgins. Vision-Based Control Of 3d Facial Animation. In *Sca '03: Proceedings Of The 2003 Acm Siggraph/Eurographics Symposium On Computer Animation*, Pages 193–206, Airela-Ville, Switzerland, 2003. Eurographics Association.
38. Sorkine O., D. Cohen-Or, Y. Lipman, M. Alexa, C. R'Ossl, and H.-P. Seidel. Laplacian Surface Editing. In *Proc. Of Eurographics Symposium On Geometry Processing 04*, Pages 179–188, 2004.
39. Shin H. J., Y. Lee ,Expression Synthesis And Transfer In Parameter Spaces, *Computer Graphics Forum* Volume 28, Issue 7, Pages 1829–1835, October 2009
40. Cosker D., R. Borkett, D. Marshall And P. L. Rosin, "Towards Automatic Performance Driven Animation Between Multiple Types Of Facial Model", *Iet Computer Vision*, Vol. 2, No. 3, Pages 129-141. 2008.
41. Bickel B., M. Lang, M. Botsch, M. Otaduy, M. Gross.: Pose-Space Animation and Transfer Of Facial Details. In: *Proceedings Of The 2008 Acm Siggraph/Eurographics Symposium On Computer Animation* (2008)

-
42. Choe B., H. Lee, H. Ko: Performance-Driven Muscle-Based Facial Animation. *J. Vis. Comput. Animat.* 12, 67–79 (2001)
 43. Deng Z., Neumann U., Lewis J., T. Kim, M. Bulut, et al. Expressive facial animation synthesis by learning speech coarticulation and expression spaces. *IEEE Transactions on Visualization and Computer Graphics*, 12(6), 1523-1534.2006.
 44. Popovic Z. And Witkin A.. "Physically Based Motion Transformation", In *Proceedings Of Siggraph 99*, Pp. 11-20, 1999.
 45. Parke F. I.. 1982. Parameterized models for facial animation. *IEEE computer graphics and applications*. Vol.2, No 9, pp 61-68.
 46. Yu H., Garrod O. and Schyns P.. Perception-Driven Facial Expression Synthesis, *Computer & Graphics*, Vol. 36, Issue 3, May 2012, pp152–162.
 47. Chin S., K. Kim. Emotional Intensity-Based Facial Expression Cloning for Low Polygonal Applications *IEEE Trans on Systems, man, and Cybernetics-Part C: Applications and Reviews*, Vol. 39, No, 3, May 2009.
 48. www.imdb.com/title/tt0360717/
 49. Samuel S.-M. Li, Charlie C.L. Wang, and Kin-Chuen Hui, "Bending-invariant correspondence matching on 3D human bodies for feature point extraction", *IEEE Transactions on Automation Science and Engineering*, vol.8, no.4, pp.805-814, October 2011.
 50. <http://Www.Di3d.Com/Index.Php>.
 51. Ying S., J. Peng, S. Du, H. Qiao. A Scale Stretch Method Based on ICP for 3D Data Registration. *IEEE Transactions on Automation Science and Engineering*. Volume: 6 , Issue: 3. Pp559-565. 2009
 52. Vezzetti F., Marcolin F., Geometry-based 3D face morphology analysis: soft-tissue landmark formalization, *Multimedia Tools and Applications* (2013), <http://dx.doi.org/10.1007/s11042-012-10913>