

# Transfer Facial Expressions with Identical Topology

Alice J. Lin

Department of Computer Science  
University of Kentucky  
Lexington, KY 40506, USA  
alice.lin@uky.edu

Fuhua (Frank) Cheng

Department of Computer Science  
University of Kentucky  
Lexington, KY 40506, USA  
cheng@cs.uky.edu

**Abstract**—We present a novel method to produce realistic expressions and animations by transferring existing expressions from a given facial model to a new facial model. The representation of the target model is first converted to make its topology the same as the source model. The radial basis function is employed to deform the template meshes to fit the target model. We then map all the vertices to the surface of the target model while preserving its spatial relationships with neighboring vertices. Therefore, facial expressions and motions can be faithfully transferred between models. With this method, a person can create many different models with the same animated expression even if these models have different topologies and, for any given model, a person can create many different expressions.

**Keywords**—*Facial expression; Animation; Topology*

## I. INTRODUCTION

The face is the primary part of the body that we use to recognize individuals. We can recognize a specific face from a vast universe of similar faces and are able to detect very subtle changes in facial expression. The ability to model human faces and then animate subtle nuances of facial expressions remains a significant challenge in computer graphics. Despite a heavy reliance on traditional computer graphics algorithms, facial modeling and animation are still being regarded as an area without broadly accepted solutions.

To generate animated facial expressions requires generating continuous and realistic transitions between different facial expressions. In general, morphing between arbitrary polygonal meshes is difficult, since it requires a set of correspondences between meshes with potentially different topologies that can produce a reasonable set of intermediate shapes.

Facial animations of 3D models derive from physical behaviors of the bone and muscle structures. Others focus on the surface of the face, using smooth surface deformation mechanisms to create dominant facial expressions. These approaches are only adequate for making individual models. If one wants to create a new model for an animation, method-specific tuning is inevitable (or otherwise the animation has to be produced from scratch).

A parametric approach associates the motion of a group of vertices to a specific parameter[1]. This manual association must be repeated for models with different mesh structures. Animation parameters do not simply transfer between models. If manual tuning or computational costs are high in creating animations for one model, creating similar animations for new

models will take similar efforts. Vector-based muscle models place the heuristic muscles under the surface of the face [2, 3]. This process is repeated for each new model. A three-layer mass-spring-muscle system requires extensive computation [4]. The final computed parameters are, however, only useful for one model. Free-form deformation manipulates control points to create key facial expressions [5], but there is no automatic method for mapping the control points from one model to another.

In practice, animators often sculpt key-frame facial expressions for every three to five frames to achieve the highest-quality animations. When a new model is considered, those fitting or sculpting processes must be repeated even if the desired expression sequences are similar. Lately, animators are concerned with achieving 3D facial realism, and want to transfer existing animations from old models to new models. We propose a new method to transfer existing expressions from a given facial model to a new facial model by constructing identical meshes between given models.

## II. RELATED WORK

Automatically transferring facial motions from an existing (source) model to a new (target) model can significantly save painstaking work and model-specific animation tuning for the new facial model. Transferring facial motions between two 3D face meshes can be performed through geometric deformations. Noh and Neumann [6] proposed a technique to transfer vertex displacements from a source model to target models. The basic idea is to construct vertex motion mappings between models. Sumner and Popovic [7] proposed a general framework that transfers geometric deformations between two triangle meshes, which can be directly applied to retarget facial motions from one source face mesh to a target face mesh.

A number of approaches were proposed to transfer source facial motions to blend shape face models [8-12] due to the popularized use of blend shape methods in industry practice. Choe, Lee and Ko [10] transferred tracked facial motions to target blend shape face models composed of hand-generated muscle actuation bases, by iteratively adjusting muscle actuation bases and analyzed weights through an optimization procedure. The work of Pyun et al. [11, 12] showed transfer of facial animations using example-based approaches. Essentially, these approaches require animators to sculpt proper blend-shape face models based on a set of key facial poses, delicately chosen from source facial animation sequences. Hence, it is difficult to apply these techniques to pre-designed blend-shape

models without considerable effort. Sifakis et al. [9] first created an anatomically accurate face model composed of facial musculature, passive tissue, and underlying skeleton structure. They then used nonlinear finite element methods to determine accurate muscle actuations from the motions of sparse facial markers. Anatomically accurate 3D face models are needed for this approach, which is another challenging task itself in computer animation. Deng et al. [8] proposed an automatic technique to directly map 3D facial motion capture data to pre-designed blend shape face models. In their approach, Radial Basis Function networks are trained to map a new motion capture frame to its corresponding blend shape weights, based on chosen training pairs between mocap frames and blend shape weights. Bilinear models and multilinear models were proposed to transform facial motions [13-15]. Chuang and Bregler [14, 15] learned a facial expression mapping/transformation function from training video footage using the bilinear models [16]. They then used this mapping function to transform input video of neutral talking to expressive talking. Vlasic et al. [13] proposed a framework to transfer facial motion in video to other 2D or 3D faces by learning statistical multilinear models from scanned 3D face meshes. In their work, the learned multilinear models are controlled via intuitive attribute parameters.

Our method enables animators to transfer existing expressions from a given facial model to a new facial model even if these models have different topologies. The approach first converts the representation of the target model to make its topology the same as the source model. Then, we transfer the facial expressions. The amount of time saved for animators is significant.

### III. OUR APPROACH

Figure 3(a) is an example of a template model. There is a separate facial model (target model) and its mesh structure is different from the template model. Figure 3(b) is an example of a target model. We want the target model to have the same animated facial expressions as the template model (e.g. smiles). Our goal is to generate a mesh (see Figure 3(c)) that has the same topology as Figure 3(a) for the target model (Figure 3(b)). Our novel approach is to transform the mesh representation of the template model into a mesh representation of the target model so that topology of the target model's mesh structure is the same as the template model's mesh structure. Once the target and the template models have the same mesh structure, we transfer facial expressions from the template model to the target model by using its motion data.

#### A. Template Models

We first construct a neutral face model, and then perform deformation on the neutral model to get a variety of facial expressions. Figure 1 shows eight facial expressions. Once the animated facial expressions are created, we install them as the template models.

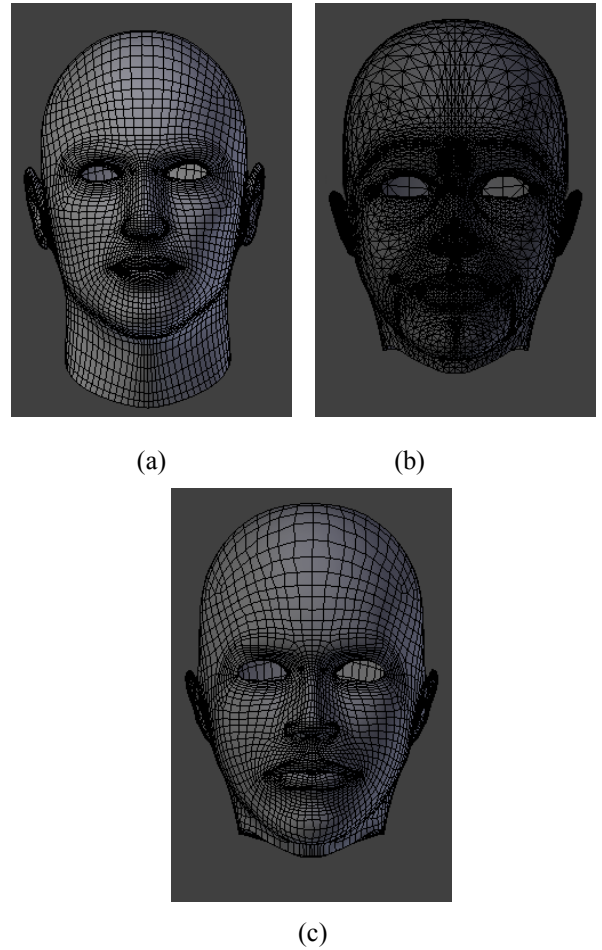


Figure 3: (a) Template model. (b) Target model. (c) New mesh representation created for the target model.

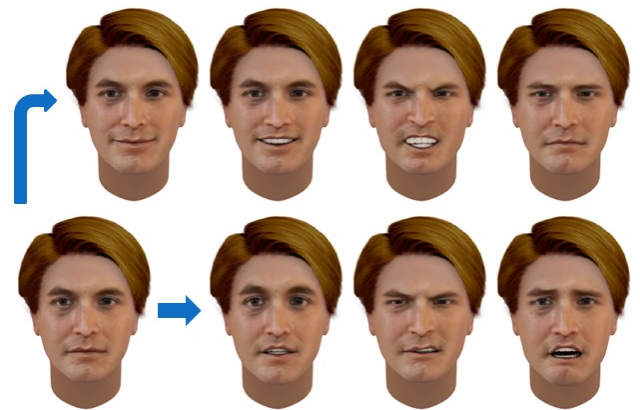


Figure 1: Eight facial expressions.

#### B. Preprocess

In general, pre-defined models have extra dangling mesh pieces. Dangling mesh pieces were generated when the model was created. They are located inside the model. They are extra

mesh pieces. If removing them, the model shape will not be affected. The purpose that we remove dangling mesh pieces is to correctly calculate the distance when a vertex of the template model is mapped to the surface of the target model. For instance, Figure 2(a) is the front view of the surface of a mouth and Figure 2(b) is the back view of the surface. The preliminary process is to remove these dangling pieces. The results are shown in Figure 2(c) and Figure 2(d). This process needs to be done both in template model and target model.

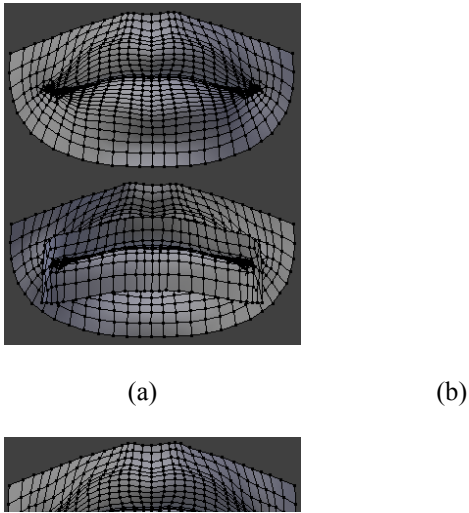


Figure 2: (a) Original front view. (b) Original back view. (c) Front view of the processed mouth. (d) Back view of the processed mouth.

### C. Generating Identical Topologies

Essentially all human faces have the same basic structure and are similar in shape. For models with human facial structures, we choose the same starting facial expression (e. g. neutral) for both the template model and the target model, and then establish correspondence by marking feature points on the template model and the target model (Figure 4). Facial feature points may include eye corners, upper and lower eyelids, mouth corners, uppermost and lowermost parts of the lips, nose tip and edges, etc. We resize the template model to try to establish the same size facial features as the target model, and then superpose the template model on the target model according to correspondence between features (Figure 5).

When specific facial feature points are matched between the two models, a morphing is performed on the mesh of the template model. The template model is deformed to certain extent, depending on the differences between these two models. The radial basis function (RBF) based shape interpolation techniques are employed to deform the template meshes to fit the target models. The benefits of modeling surfaces with RBFs have been recognized in [17-19] [20-22]. The radial basis functions associated with a surface can be evaluated at any location to produce a mesh at the desired resolution.

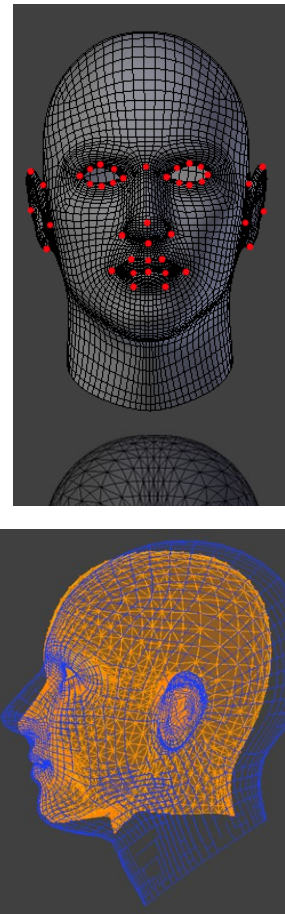


Figure 5: Superpose template model on target model.

Radial deformations offer the greatest simplicity and efficiency of all the point-based spatial deformations. The deformations are determined by an arbitrary number of constraints, each consisting of a spherical radius of effect centered on a constraint point with an associated displacement [23]. The idea is to have a small number of control points scattered over the face. Each of these control points will have an influence area in which they will affect the movement of neighboring mesh points. This influence is experienced through a normalized decay function that depends on the distance to the control point and the displacement suffered by the associated control point.

As template and target model not have the same mesh topology, we map each vertex in the template model to the surface of the target model to generate a new mesh with an identical topology as the template model. The process is as follows.

For each vertex of the template model  $v_i$ ,  $i = 1, 2, \dots, n$ , if  $v_j$  is a vertex of the target model such that the distance between  $v_i$  and the vertices of the target model is the minimum,  $d = \min\|v_i - v_j\|$ , then  $v_i$  is mapped to  $v_j$ . Symbolically, the transformation from the template model to the target model is

defined by a replacement vector  $w_i$  for each vertex  $v_i$ . Using the above notation,  $w_i = v_i$  on the generated mesh (Figure 6).

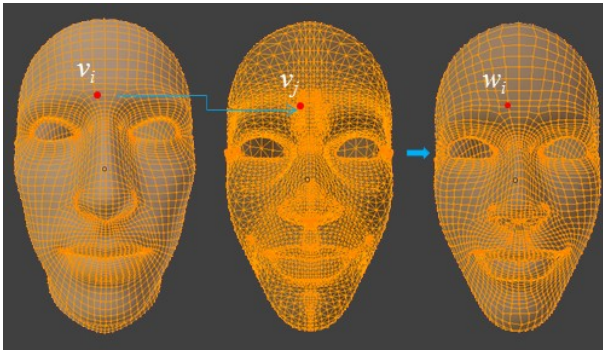


Figure 6: Mapping each vertex in the template model to the surface of the target model

Each mapped vertex  $w_i$  in the generated mesh is subject to the following constraint. The vertex  $v_i$  in the template model has spatial relationships with its neighboring vertices. For example, in Figure 7(a), there are nine vertices in the template model. Suppose  $v_i=v_9$ , then  $v_9$  has spatial relationships with eight vertices  $v_1, v_2, \dots, v_8$ . These nine vertices are mapped to the target model (see Figure 7(b)). The corresponding vertex of  $v_9$  is  $w_9$ .  $w_9$  has to keep the same spatial relationships with  $w_1, w_2, \dots, w_8$ , including the nine vertices' order. Figure 7(b) shows a correct relationship on the generated mesh. The situation in Figure 7(c) is something we want to avoid. However, if  $w_7$  and  $w_8$  overlap, then there is no problem. To avoid this situation shown in 7(c), we perform mesh relaxation [24]. The result is that the  $n$  vertices of the template model are all mapped onto the surface of the target model. The mesh now represents the target model (see Figure 3(c)).

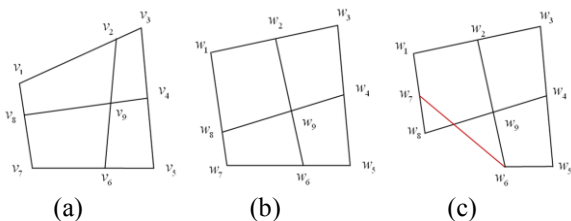


Figure 7: Spatial relationship between neighboring vertices. (a) Vertex relationships on the template model. (b) Correct relationship on the generated mesh. (c) Incorrect relationship on the generated mesh.

We smooth and denoise the newly generated mesh which represents the target model. We use the algorithm in [25-28] to obtain feature-preserving mesh smoothing. A bilateral filtering is conducted on the normals instead of the vertex positions [29] to preserve sharp features. Using normals to preserve features on a mesh is more intuitive since normals typically change abruptly at edges and creases.

#### D. Transferring and Animating Expressions

Once the topologies of the two facial meshes are identical, there is already a natural correspondence between their vertices. A satisfactory 3D morphing sequence can be obtained using linear interpolation between geometric coordinates of corresponding vertices in the two facial meshes. To transfer an expression from the template model to the target model we need motion data for the template model animation from one expression (e.g. the neutral face) to another expression (e.g. the smiling face). The motion data contains each vertex's positional change between these two expressions. We proportionally apply the motion data to the newly generated mesh. The newly generated mesh thus gets an expression (e.g. smiling). For the animation, we transform one (neutral face) expression into another facial expression (e.g. smiling). We can simply interpolate the two shapes since they share the same vertex-edge topology. Facial expressions are transformed by interpolating the positions of vertices on a vertex-by-vertex basis. The correspondence between the two shapes is established by the vertex-edge connectivity structure shared by the two models.

The above approach faithfully transfers facial expressions and motions between models. With this method, anyone can create many different models with the same animated expression, even if these models have different topologies. For each model, anyone can create a variety of expressions. Figure 7 shows examples of transferring facial expressions between three models.

#### IV. CONCLUSION AND FUTURE WORK

We have presented a method to make two facial models have the same mesh structure. Thus, 3D morphing sequences can be obtained using linear interpolation between coordinates of corresponding vertices in the two facial meshes. Our method works on the models represented by polygonal meshes and polygon soup. One major limitation of this approach is that it relies on a manual preliminary process to remove dangling pieces in the models. In the future, we would like to overcome this limitation, and would like to extend our method to generate identical meshes for models with point clouds. Our technique produces natural looking expressions and animations for arbitrary mesh structures. This method is not developed with ad hoc techniques, so it is easily extendible. Generating the same expression models often is a tedious and complex process requiring substantial artistic skills. Our method is not only for animation by a trained artist, but also for ordinary engineers.

#### ACKNOWLEDGMENT

Research work of the second author is supported by National Science Foundation of China (6102010661, 61170324), National Science Council of ROC (NSC-100-2218-E-007-014-MY3), and a joint grant of National Tsinghua University and Chang-Gung Memorial Hospital (101N2756E1).



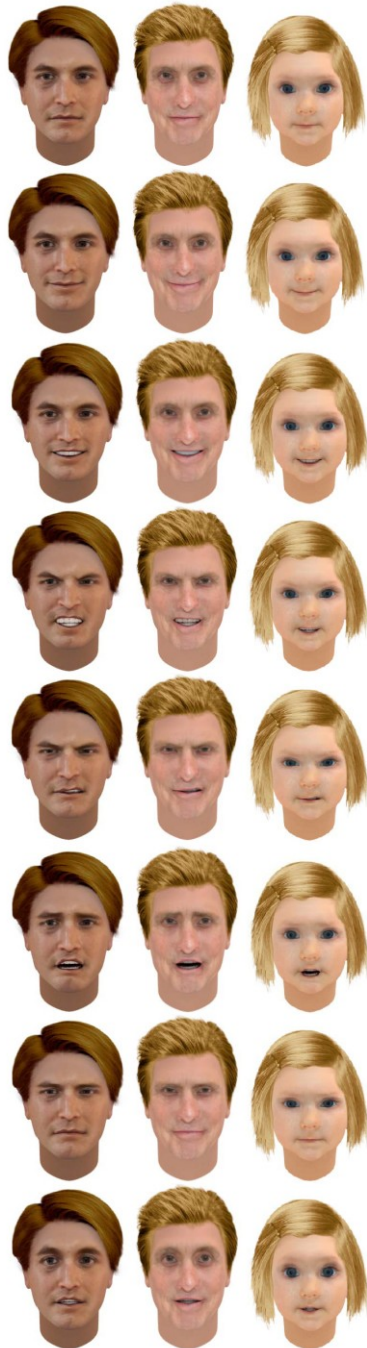


Figure 7: 1  
expression

[1] F. I. Parl  
Computer Gi  
[2] K. Water  
Expressions,” Computer Graphics (SIGGRAPH Proceedings) vol. 21, no. 4,  
pp. 17-24, 1987.  
[3] K. Waters, and F. J, “A Coordinated Muscle Model for Speech  
Animation,” Graphics Interface, pp. 163-170, 1995.  
[4] Y. C. Lee, D. Terzopoulos, and K. Waters, “Realistic Face Modeling For  
Animation,” SIGGRAPH Proceedings, pp. 55-62, 1995.  
[5] P. Kalra, A. Mangili, N. Magnenat-Thalmann et al., “Simulation of Facial  
Muscle Actions Based on Rational Free Form Deformations,” Proc.  
Eurographics ’92, Cambridge, U.K., Computer Graphics Forum, vol. 2, no. 3,  
pp. 59-69, 1992.

[6] J. Y. Noh, and U. Neumann, “Expression Cloning,” Proceedings of ACM  
SIGGRAPH’01, pp. 277-288, 2001.  
[7] R. W. Sumner, and J. Popovic, “Deformation Transfer for Triangle  
Meshes,” Proceedings of ACM SIGGRAPH 2004, vol. 23, no. 3, pp. 399-405,  
2004.  
[8] Z. Deng, P. Y. Chiang, P. Fox et al., “Animating Blendshape Faces by  
Cross-Mapping Motion Capture Data,” ACM SIGGRAPH Symposium on  
Interactive 3D Graphics and Games, pp. 43-48, 2006.  
[9] E. Sifakis, I. Neverov, and R. Fedkiw, “Automatic Determination of  
Facial Muscle Activations from Sparse Motion Capture Marker Data,” ACM  
Transactions on Graphics, vol. 24, no. 3, pp. 417- 425, 2005.  
[10] B. Choe, H. Lee, and H. S. Ko, “Performance Driven Muscle Based  
Facial Animation,” Journal of Visualization and Computer Animation, vol. 12,  
no. 2, pp. 67-79, 2001.  
[11] J. Ma, R. Cole, B. Pellom et al., “Accurate Automatic Visible Speech  
Synthesis of Arbitrary 3d Model Based On Concatenation of Diviseme  
Motion Capture Data,” Computer Animation and Virtual Worlds, vol. 15, pp.  
1-17, 2004.  
[12] H. Pyun, Y. Kim, W. Chae et al., “An Example-Based Approach for  
Facial Expression Cloning,” In Proceedings of Symposium on Computer  
Animation, pp. 167-176, 2003.  
[13] D. Vlastic, M. Brand, H. Pfister et al., “Face Transfer with Multilinear  
Models,” Proceedings of ACM SIGGRAPH 2005, vol. 24, no. 3, pp. 426-433,  
2005.  
[14] E. S. Chuang, H. Deshpande, and C. Bregler, “ Facial Expression Space  
Learning,” In Proceedings Of Pacific Graphics, pp. 68-76, 2002.  
[15] E. Chuang, and C. Bregler, “Moodswings: Expressive Speech  
Animation,” ACM Transactions On Graph, vol. 24, no. 2, 2005.  
[16] J. B. Tenenbaum, and W. T. Freeman, “Separating Style and Content  
with Bilinear Models,” Neural Computation, vol. 12, no. 6, pp. 1247-1283,  
2000.  
[17] P. BORREL, and A. RAPPOPORT, “Simple constrained deformations  
for geometric modelling and interactive design,” ACMTrans. Graph, vol. 13,  
no. 2, pp. 137-155, 1994.  
[18] M. BOTSCH, and L. KOBELT, “Real-time shape editing using radial  
basis functions,” Comput. Graph. Forum vol. 24, no. 3, pp. 611-621, 2005.  
[19] J. Noh, D. Fidaleo, and U. Neumann, “Animated deformations with  
radial basis functions,” In VRST ’00: Proceedings of the ACM symposium on  
Virtual reality software and technology, New York, NY, USA, pp. 166-174,  
2000.  
[20] G. YNGVE, and G. TURK, “Robust creation of implicit surfaces from  
polygonal meshes,” IEEE Transactions on Visualization and Computer  
Graphics vol. 8, no. 4, pp. 346-359, 2002.  
[21] J. CARR, R. BEATSON, J. CHERRIE et al., “Reconstruction and  
representation of 3d objects with radial basis functions,” In Proceedings of  
SIGGRAPH, vol. 67, no. 76, 2001.  
[22] D. COHEN-OR, A. SOLOMOVIC, and D. LEVIN, “Three dimensional  
distance field metamorphosis,” ACM Transactions on Graphics vol. 17, no. 2,  
pp. 116-141, 1998.  
[23] J. Gain, “A survey of spatial deformation from a user-centered  
perspective,” ACM Transactions on Graphics, vol. 27, no. 4, 2008.  
[24] A. Nealen, O.Sorkine, M.Alexa, and D.Cohen-Or, “A sketch-based  
interface for detail-preserving mesh editing”, Proceeding of SIGGRAPH’07,  
ACM SIGGRAPH 2007 courses Article No. 42  
[25] Yizhou Yu, Kun Zhou, Dong Xu et al., “Mesh Editing with Poisson-  
Based Gradient Field Manipulation,” ACM Trans. Graph. In SIGGRAPH, vol.  
23, no. 3, pp. 644-651, 2004.  
[26] G. TAUBIN, “Linear anisotropic mesh filtering,” Tech. rep., IBM  
Research Report RC2213, 2001.  
[27] T. TASHIZEN, R. WHITAKER, P. BURCHARD et al., “Geometric  
surface smoothing via anisotropic diffusion of normals,” In Proceedings IEEE  
Visualization,, pp. 125-132, 2002.  
[28] H. YAGOU, Y. OHTAKE, and A. BELYAEV, “Mesh denoising via  
iterative alpha-trimming and nonlinear diffusion of normals with automatic  
thresholding,” In Proc. Computer Graphics Intl. 2003.  
[29] S. FLEISHMAN, I. DRORI, and D. COHEN-OR, “Bilateral mesh  
denoising,” ACM Trans. Graphics vol. 22, no. 3, pp. 950-953, 2003.

facial

E  
2.

al Facial