

A Generic Framework for Efficient 2D and 3D Facial Expression Analogy

Mingli Song, *Member, IEEE*, Zhao Dong*, *Student Member, IEEE*,
 Christian Theobalt, *Member, IEEE*, Huiqiong Wang, Zicheng Liu, *Senior Member, IEEE* and
 Hans-Peter Seidel, *Senior Member, IEEE*

Abstract—Facial expression analogy provides computer animation professionals with a tool to map expressions of an arbitrary source face onto an arbitrary target face. In the recent past, several algorithms have been presented in the literature that aim at putting the expression analogy paradigm into practice. Some of these methods exclusively handle expression mapping between 3D face models, while others enable the transfer of expressions between images of faces only. None of them, however, represents a more general framework that can be applied to either of these two face representations. In this paper, we describe a novel generic method for analogy-based facial animation that employs the same efficient framework to transfer facial expressions between arbitrary 3D face models, as well as between images of performer’s faces. We propose a novel geometry encoding for triangle meshes, vertex-tent-coordinates, that enables us to formulate expression transfer in the 2D and the 3D case as a solution to a simple system of linear equations. Our experiments show that our method outperforms many previous analogy-based animation approaches in terms of achieved animation quality, computation time and generality.

Index Terms—Facial animation, Facial image synthesis, Expression analogy.

EDICS (Primary): 1-FACE
EDICS (Secondary): 3-VRAR

I. INTRODUCTION

THE creation of realistic animated faces is still one of the most challenging tasks for visual effect professionals. Many elements in a human face contribute to the realistic appearance of a facial expression. The shape of the mouth, the look of the eyebrows, as well as the gaze are just the most important clues that an observer perceives. These visual clues have also been used for recognizing facial expression [1]. However, also more subtle details, such as wrinkles and the tone of the skin under different illumination conditions, are important components of the overall picture of a face. It is thus no wonder that, if in a computer-animated face only one of these elements is not convincingly simulated, the illusion of looking at a real human will immediately be compromised.

M. Song is with Microsoft Visual Perception Laboratory, Zhejiang University, No. 38 Zheda Road, HangZhou, P.R.China. Tel: +86-137-5088-8252 Fax: +86-571-8795-1947 Email: brooksong@ieee.org

Z. Dong, C. Theobalt and H.-P. Seidel are with MPI Informatik, AG4, Stuhlsatzenhausweg 85 66123 Saarbrücken, Germany. Tel: +49-681-9325-552 Fax: +49-681-9325-499 Email: {dong, theobalt, hpseidel}@mpi-sb.mpg.de*

H. Wang is with the department of information science and electronic engineering, Zhejiang University, No. 38 Zheda Road, HangZhou, P.R.China. Email: huiqiongw@gmail.com

Z. Liu is with Microsoft Research, One Microsoft Way, Redmond, WA 98052. E-mail: zliu@microsoft.com

In the past, it has been tried to meet these high requirements in visual quality by exactly modeling and animating the human 3D face geometry [2]–[5]. However, it is a non-trivial and time-consuming task to tune the parameters of the underlying deformation framework, e.g. a simulated muscle system. Furthermore, it is hardly possible to transfer animations between different individuals.

Image-based approaches [6]–[8] aim at generating realistic talking heads by analyzing images or video data showing real performers. Many of them suffer from quality deterioration in the presence of image noise or require a database of example facial expressions that is not easy to build.

Only recently, novel scanning devices have been presented that enable real-time capturing of the dynamic 3D geometry of a performing actor’s face [9], [10]. However, although such high-quality dynamic shape data become more easily accessible, it is still complex and expensive to capture facial expression sequences for many different subjects with such a device. The animators are still in need of efficient methods to transfer captured expression sequences onto models of other individuals. While 3D acquisition is still complicated, photographs of facial expressions can be captured very easily. There exists already a number of image databases showing different people performing a variety of facial expressions, such as the FERET database [11]. For animation professionals, it would be a great leap forward if they were able to transfer photographed facial expressions onto portraits of arbitrary people. In both the 3D and the 2D expression mapping case, it is important that all the appearance details of the facial expression are transferred from one model to the other. Facial expression analogy provides animators with a tool that serves this purpose. Only a few algorithms for analogy-based animation have been presented in the past [4], [6], [12]–[16]. Unfortunately, they can either be exclusively applied to 3D face models or only allow for expression transfer between photographs.

In contrast, we present a generic method for analogy-based facial animation that on one hand can transfer expressions from a source 3D face model to a target 3D face model, and on the other hand can map expressions from a source image of a face to an arbitrary target image. Our method enables the target face to mimic even subtle expression details in the source face, such as wrinkles. In the 2D case it even enables convincing expression mapping if the lighting situations in the source and the target image differ. To achieve this purpose, we represent both 3D face geometry as well as 2D images of

faces as 3D triangle meshes. In both the 2D and the 3D case motion and expression details are mapped from input to output models by making use of a novel local geometry encoding, vertex tent coordinate. This representation enables us to map facial expressions from input to output models via solving a simple system of linear equations. This paper introduces the following key-contributions:

- a generic method for high-quality analogy-based expression transfer between 3D face meshes, as well as 2D face images,
- VTC (Vertex Tent Coordinate) – a new local geometry encoding method for 3D surface representations,
- an approach that handles not only triangle meshes but also quadrangle meshes,
- a method that realistically transfers all details of facial expressions despite differences in lighting and facial proportions,
- A formulation of facial expression analogy as the solution of simple linear equation systems to get high efficiency.

The remainder of this paper is organized as follows: We review important related work in Sect.II, and give an overview of our method in Sect. III. Sect. IV details the theoretical fundamentals of our local geometry encoding based deformation transfer algorithm that is the heart of our facial expression analogy method. Sect. V presents the nuts and bolts of 3D facial expression analogy, while Sect. VI deals with expression mapping between images of faces. We describe experimental results and a comparison to related methods from the literature in Sect. VII, and conclude with an outlook to future work in Sect. VIII.

II. RELATED WORK

In the past decades, an enormous amount of scientific work has been presented in the field of facial animation. Since it would be virtually impossible to name all of these methods, we refer the interested reader to the book by Parke and Waters [17], and constrict our review to analogy-based approaches. Expression analogy [12], [18], [19] (also called expression mapping, expression retargetting) is a generic term for a body of methods that allow for the transfer of a facial expression from a source face to a target face. A target face (2D image or 3D model with/without texture) with the same expression as the source is necessary to perform expression analogy. For ease of acquisition, a neutral target face is usually employed in the related work. In our work, we also employ a neutral target face as input.

Many of the previously presented expression retargetting approaches are applicable to 3D face models only. Pighin et al. [13] parameterized each person's expression space as a combination of some elementary but universal expressions. The expression is decomposed into a set of coefficients. The coefficients are applied to another model to obtain a similar expression. As an extension of Facial Action Coding System (FACS) [20], the concept of facial expression parameters (FAPs) for facial expression synthesis has also found its way into the MPEG-4 standard [21], [22]. By using 68 FAPs, generic face motion are defined to synthesize the facial expression, which can be applied conveniently to facial expression

analogy. Pyun et al. [4] improved the parameterization method by introducing radial-basis-function(RBF)-based interpolation. Park [23] extended this further to feature-based expression cloning. Unfortunately, these approaches usually use low resolution face model which is lower computation complex. Neither of these methods can mimic subtle expression details (wrinkles, furrows, etc.) on the target due to their sparse vertex distribution. Though textures are added to the 3D face model to enhance the realism, it's difficult to assess the quality of the geometric 3D deformation as they are masked by the texture. In fact, Park et al. employed fairly coarse face meshes such that they were not able to show the same geometric detail in transferred expressions as we do. Noh and Neumann [14] developed a different 3D facial analogy method. They apply a semi-automatic technique to find correspondence between the source and target face, which is also applied as part of our feature point localization method. They also developed a new motion mapping technique that adjusts the directions and magnitudes of the motion vectors by taking into account the difference in local geometry between the source and the target.

Recently, deformation transfer between triangle meshes has become an important research topic in geometric modeling and high resolution 3D face modeling. Sumner et al. [15] modeled the wrinkles by a series of triangle-based transformations and map transformations from input to output meshes by applying per-triangle local deformations and zippering the so-created disconnected mesh. In contrast, we propose to formulate deformation transfer on a per-vertex basis, which enable us to not only deal with triangle meshes but also with quadrangle ones. It has also been popular to use Laplacian or differential coordinates for wrinkle modeling [16], [24], [25]. The Laplacian coordinate of a vertex is a vector whose direction approximates the normal direction and whose length is proportional to the mean curvature. This implicit encoding of local normals has a couple of disadvantages. For instance, if the one-ring neighborhood triangles of source vertex are coplanar, the direction of its Laplacian coordinate can not correctly approximate the vertex normal and some deformation details will be lost. Inspired by [15], in our representation, we explicitly model the local surface normals in order to preserve subtle shape details during expression mapping. Very recently, Botsch et. al. [26] presented a thorough analysis of the relationship between gradient-based deformation and the deformation transfer method. It shows their equivalence for surface meshes. Deriving a similar correspondence for our method may be feasible. In the context of facial expression editing, these methods require manual feature point localization.

A second category of mapping algorithms aims at transferring facial expressions between images of people. Liu et al. [6] proposed a new 2D data representation, called the expression ratio image (ERI), that can be used to map one person's expression details to a different person's face. The appearance of wrinkles is modeled in an image-based way by the variation of the ERI between pixels. Although their results are convincing, the authors concede that large differences in illumination between the source and the target image cannot faithfully be handled, and an adaptive Gaussian

filter needs to be applied automatically to reduce artifacts caused by misregistrations. This process is very time consuming. The misregistrations is usually due to the imprecise location of the feature points in the face whether manually or automatically. Different from this, our method can avoid the artifacts without the filtering. Blanz et al. [27] developed an algorithm to exchange faces in images that can also cope with large differences in viewpoint and illumination between input and output. Unfortunately, subtle expression details (such as wrinkles and furrows) that vary between different facial expressions cannot be fully represented. Song et al. [28] tried to tackle this problem by a vector field decomposition method. It is a triangle-based solution which can be regarded as an extension of Sumner's method [15] being applied to 2D images. However, the triangulation operation led to longer runtimes and higher memory consumption. Different from [28], our method treats the image as an quadrangle mesh directly without triangulation. Manual labeling of feature points is needed in these methods for high-quality results.

Most closely related to our method is the approach by Zhang et al. [7]. They propose a technique to synthesize detailed facial expressions of a target face model (2D or 3D) by analyzing the motion of feature points on a performer's face. Their algorithm requires example expressions of the target face model, which is not always available. And, the feature points in the face are labeled manually.

In contrast to the aforementioned methods, we present a generic solution to both 3D and 2D facial expression analogy. It allows for high-quality expression transfer between 3D face models, and also enables robust expression mapping from the source expression image to a neutral target one even in the presence of lighting differences.

Our facial expression analogy approach has a number of important advantages over related algorithms. Firstly, our method outperforms related approaches in terms of quality and speed in both the 2D and the 3D case. Secondly, the fact that the facial expression analogy problem in both the 2D and the 3D domain can be formulated in terms of the same efficient framework is an important theoretical insight by itself. Thirdly, our technique based on vertex-tent coordinates renders our approach very flexible since we can process both triangle and quadrangle meshes in the same way. As an additional benefit, our method fits well into the standard production pipeline for movies and games. In this application scenario, it is nowadays state-of-the-art to capture facial expression sequences with a structured light scanner [9]. Our approach enables significant cost reductions as facial expression sequences only need to be captured once from a single actor and can thereafter be mapped onto arbitrary other actors from whom only a static scan is available. Finally, a unified and fast framework like ours reduces implementation complexity, as the same routines can be employed for both the 2D and the 3D case.

III. PROBLEM STATEMENT AND OVERVIEW

In a nutshell, our method enables us to transfer a facial expression of an arbitrary input face to an arbitrary output face. It is equally applicable to both 3D face models and 2D

face images. The input to our method comprises of a source face with a neutral expression, henceforth termed source neutral face S , the same source face in a desired expression, henceforth termed source expressive face S' , and a target face in a neutral expression, henceforth termed target neutral face T . Our algorithm maps the expression of S' onto the target face, thereby creating the output T' . In the 3D case, either of the faces comes as a 3D triangle mesh. In the 2D case, each face is represented as a picture. To make 3D and 2D faces accessible to the same expression analogy method, we transform each face picture into a 3D mesh. By means of a local geometry encoding termed vertex-tent-coordinate (VTC), expressions can be transferred between source and target faces by solving a simple system of linear equations.

Before an expression is mapped, the source neutral face, the source expression face, and the target neutral face need to be aligned so that they have the same orientation, location and scale. Consequently, per-vertex correspondences between source and target models are established. In our method, a strategy to locate feature points is proposed that requires only a minimum of manual interaction. Our strategy consists of two substeps. In the 3D case , the first substep is an automatic labeling process which locates feature points by means of heuristic rules originally proposed in [14]. In the 2D case, the first substep adopts an AAM (Active Appearance Model) [29] based tracking method to locate the feature points automatically. Though all human faces exhibit the same anatomical features, their occurrences may greatly differ across different individuals. Therefore, to robustly account for these anatomical differences during correspondence finding, we ask the user to manually adjust the feature points after automatic initialization in the second substep. These feature points specify a set of corresponding locations between the source and the target face. The feature point set should include the following elements (as shown in Fig. 1 for the 3D and the 2D case):

- (1) Feature points on the contour of the eye brows.
- (2) Feature points on the contour of the eyes.
- (3) Feature points on the nose, including tip and wings.
- (4) Feature points on the mouth.
- (5) Feature points on the jaw.
- (6) Feature points on the cheek.
- (7) Feature points on forehead.

Once the correspondences between the input and the output face have been established, our VTC-based deformation transfer method is applied to map the source expression onto the target. Note that apart from the manual adjustment substep mentioned above, the whole facial expression analogy pipeline is fully-automatic. The above sequence of processing steps forms the algorithmic backbone of both our 3D and 2D expression analogy workflow. However, their individual implementations slightly differ which we detail in the respective subsections of Sect. V and Sect. VI.

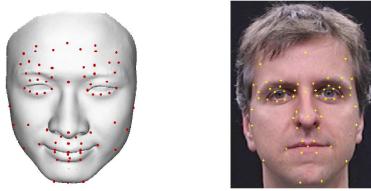


Fig. 1. Feature points on 3D face model (left) and 2D face image (right)

IV. DEFORMATION TRANSFER

We use 3D triangle or quadrangle meshes to represent source and target faces in both the 3D and the 2D case (see also Sect. VI-A for the specifics of the 2D case). A novel local vertex-centered geometry encoding of a 3D mesh enables us to better preserve subtle shape details while mapping expressions from source to target. The target expression is obtained with solving a simple system of linear equations.

For the purpose of expression analogy, we represent a vertex and its one-ring neighborhood by means of a set of vectors, which we call the vertex-tent-coordinates. Given a vertex \mathbf{v}_0 and its one-ring neighboring vertices $\{\mathbf{v}_1, \dots, \mathbf{v}_n\}$, shown in Fig. 2, the first component of our vertex-tent-coordinate (VTC) is matrix formed by a set of vectors $\mu = [\mathbf{v}_1 - \mathbf{v}_0 \ \mathbf{v}_2 - \mathbf{v}_0 \ \dots \ \mathbf{v}_n - \mathbf{v}_0 \ 0]$. The second component of the VTCs is defined as $\nu = [0 \ 0 \ \dots \ \mathbf{v}_{n+1} - \mathbf{v}_0]$ where we have introduced an additional vertex \mathbf{v}_{n+1} in the vertex normal direction (Fig. 2). The complete local geometry is thus encoded by $\mu + \nu$.

The main advantage of our vertex-tent representation is that it represents geometry not on a per-triangle-basis (such as Sumner et al. [15]) but rather encodes geometry in terms of a vertex, its one ring neighborhood and an explicit vertex normal. Our method also resembles the local pyramid coordinates presented by [30] which are invariant under rigid transformations. However, the reconstruction process from pyramid coordinates to vertex coordinates requires to solve a non-linear system iteratively which is rather time-consuming. In contrast, our VTC representation enables us to solve the expression transfer problem by quickly solving a linear system.

Consequently, local deformation in the neighborhood of a vertex can be described by the variations of the vertex-tent-coordinates. If we assume that $\mathbf{w} = \mu + \nu$ and $\mathbf{w}' = \mu' + \nu'$ are the VTCs of a vertex before and after deformation respectively, and \mathbf{Q} is the applied transformation matrix, the following equation holds:

$$\mathbf{Q}\mathbf{w} = \mathbf{w}' . \quad (1)$$

which can be reformulated as follows:

$$\mathbf{Q} = \mathbf{w}'\mathbf{w}^T(\mathbf{w}\mathbf{w}^T)^{-1} . \quad (2)$$

We call this formulation of the deformation problem VTC-based deformation transfer.

The VTCs enable us to express the deformation transfer between complete 3D meshes as the mapping of transformations between local vector sets. Therefore, for each vertex, we have a \mathbf{Q} for it to be applied to the corresponding vertex on the target. Due to the physical difference between the source

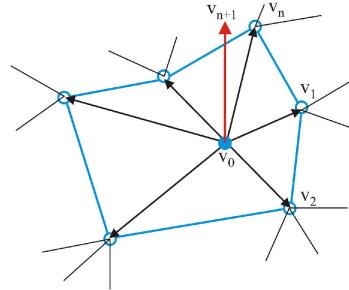


Fig. 2. Vertex-tent-coordinates (VTCs) encode the geometry of a vertex and its one-ring neighborhood. Black arrows and red arrow represent the μ and ν components of the VTCs respectively.

face and the target one, it is inevitable to generate a lot of artifacts in the result if we apply \mathbf{Q} directly. In order to make sure that the transformations of vertices in each local one-ring neighborhood comply with each other, we enforce the following consistency constraint during expression transfer:

$$\mathbf{Q}_t^j \mathbf{v}_i = \mathbf{Q}_t^k \mathbf{v}_i, \quad i \in \{1, \dots, N\}, j, k \in p(\mathbf{v}_i) . \quad (3)$$

Here, $p(\mathbf{v}_i)$ is the set of indices of all vertices in the one-ring around vertex \mathbf{v}_i and $\mathbf{Q}_t^1, \mathbf{Q}_t^2, \dots, \mathbf{Q}_t^N$ are the transforms of the N vertices of the target mesh. VTC-based deformation transfer is now performed by minimizing the difference between the source transformations $\mathbf{Q}_s^1, \dots, \mathbf{Q}_s^N$ and the corresponding target transformations $\mathbf{Q}_t^1, \mathbf{Q}_t^2, \dots, \mathbf{Q}_t^N$ under the above constraints in terms of the target transformations:

$$\begin{aligned} & \min_{\mathbf{Q}_t^1, \dots, \mathbf{Q}_t^N} \sum_{m=1}^N \|\mathbf{Q}_s^m - \mathbf{Q}_t^m\|_F^2 \\ & \text{subject to} \\ & \mathbf{Q}_t^j \mathbf{v}_i = \mathbf{Q}_t^k \mathbf{v}_i, \quad i \in \{1, \dots, N\}, j, k \in p(\mathbf{v}_i) . \end{aligned} \quad (4)$$

In the equation above, $\|\cdot\|_F$ is the Frobenius norm. If one substitutes Eq. (2) into Eq. (4), one obtains a formulation of the problem in terms of the coordinates of the target mesh in the target expression. Solving for these coordinates in a least-squares-sense corresponds to solving a simple system of linear equations, which we henceforth refer to as VTC deformation equation. The solution of this linear system can quickly be computed by means of LU decomposition [31].

Specifically, for a vertex \mathbf{v}_0 on the target, the linear system for it is described as below:

$$\mathbf{A} \begin{bmatrix} \mathbf{v}_0 \\ \mathbf{v}_1 \\ \vdots \\ \mathbf{v}_{n+1} \end{bmatrix} = \mathbf{S} \quad (5)$$

where

$$\mathbf{A} = \left[\begin{bmatrix} -1 & -1 & \dots & -1 \\ 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 \end{bmatrix} \mathbf{w}_t^T (\mathbf{w}_t \mathbf{w}_t^T)^{-1} \right]^T \quad (6)$$

$$\mathbf{S} = [\mathbf{w}'_s \mathbf{w}_s^T (\mathbf{w}_s \mathbf{w}_s^T)^{-1}]^T \quad (7)$$

V. 3D FACIAL EXPRESSION ANALOGY

We now describe how to apply our generic facial expression analogy method, in order to map an expression from a 3D source face mesh to a 3D target face mesh. The workflow of 3D facial expression analogy is illustrated in Fig. 3. The source neutral, the source expressive and the target neutral faces are given as 3D triangle meshes. The user specifies corresponding feature points on the source neutral and target neutral meshes. Thereafter, both models are automatically aligned by the method described in [32] whose main idea can be summarized as follows. Firstly, we compute the centroid coordinates of the source and the target with the aid of the feature points. Then, we derive the relative coordinates of the feature points with respect to the centroids. Thirdly, the rotation and scale value between the source and the target is computed in least-squares sense by means of singular value decomposition. The translation between the source and the target can be obtained from the distance of their centroids. By applying the computed rotation, scale and translation value, the target face is aligned to the source one.

Before we can apply our VTC-based deformation transfer approach to map the source expression onto the target, we have to establish dense per-vertex and per-triangle correspondences between the input and the output mesh. To this end, relying on the powerful Graphics Processing Unit (GPU), we have developed a very efficient GPU-assisted method. Since the source and target geometries differ in both triangle count and topology, we first establish per-vertex correspondences and correct topological differences in a postprocessing step automatically. Our correspondence finding method comprises the following main steps:

- (1). The aligned source neutral and target neutral models are projected onto cylinders.
- (2). We make use of the GPU to transform the cylindrical target mesh into an image, the so-called *mesh image*.
- (3). Using the parameterized source neutral model as reference geometry, we resample the geometry of the target mesh based on its corresponding mesh image.
- (4). Redundant vertices are deleted and the topologies of all input meshes are updated accordingly.

Once the geometry correspondences have been established, we can employ the VTC-based deformation transfer method to map the source expression onto the target face. Results of our method are shown in Fig. 8 and Fig. 10.

A. Correspondence Finding

1) *Cylindrical Projection*: After face model alignment, the source neutral and the target neutral meshes are projected onto a cylinder. For a vertex $p = [x_o, y_o, z_o]^T$, its cylindrical coordinate after projection is (u_o, v_o) , where $u_o = \arccos(x_o/r)$, and $r = \sqrt{x_o^2 + z_o^2}$. An original mesh and its corresponding cylindrical projection are shown in Fig. 4.

2) *Mesh Image*: We want to resample the geometry of the target mesh such that its topology corresponds to the topology of the source mesh. In the process of resampling, vertex coordinates of the target mesh have to be interpolated. Computing interpolated vertex coordinates on the CPU is fairly inefficient.

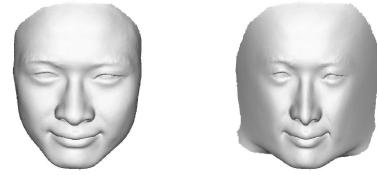


Fig. 4. The original 3D mesh (left) and its cylindrical projection (right).

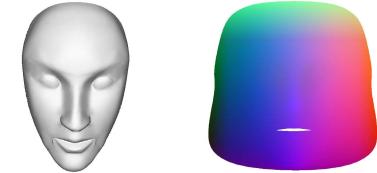


Fig. 5. A target mesh (left) and its corresponding mesh image (right)

Fortunately, current GPU support 32bit/16bit floating point format texture. Therefore we can exploit the GPU to perform geometry interpolation very efficiently. Motivated by [33], we transform the target cylindrical mesh into a high-resolution texture image. In this *mesh image*, the r , g and b color channels store the corresponding x , y , z coordinates of each vertex. The mesh image can be straightforwardly generated on the GPU by rendering the cylindrical target mesh with the vertex coordinates used as vertex colors. The hardware interpolation of the GPU leads to a dense representation of the surface geometry in the texture domain. A target mesh and its corresponding mesh image are shown in Fig. 5.

3) *Parameterization*: In a preprocessing step, the user has labeled corresponding feature points on the source and the target mesh. We make use of these feature points to define a parameterization of the source mesh. To this end, in the cylindrical mesh we triangulate the feature points as well as the four boundary vertices of the bounding box A , B , C and D , as it is illustrated in Fig. 6. The mesh obtained by this triangulation is henceforth referred to as *parameterization mesh*.

Based on this parameterization of the source mesh, we resample the target mesh in the following way: for each vertex p of the source mesh we determine in which triangle $\mathbf{t} = [l_{m1}, l_{m2}, l_{m3}]$ of the parameterization mesh it lies, and compute its barycentric coordinates $[x_b, y_b, z_b]$ with respect to \mathbf{t} . The corresponding vertex coordinate on the target mesh is computed by sampling from the target mesh image \mathbf{I}_t . The location l_{vertex} of a vertex in \mathbf{I}_t that corresponds to p evaluates to

$$\begin{aligned} l_{vertex} &= [l_{m1} \quad l_{m2} \quad l_{m3}] [x_b \quad y_b \quad z_b]^T \\ &= l_{m1}x_b + l_{m2}y_b + l_{m3}z_b \end{aligned} \quad (8)$$

where $l_i = [u_i, v_i]^T$ ($i = 1, 2, 3$) are the locations of the vertices of the parameterization triangle in the mesh image, which are equal to the locations of the corresponding marked feature points in the mesh image. Finally, the corresponding vertex' 3D coordinate $[x_{vertex}, y_{vertex}, z_{vertex}]^T$ is retrieved by sampling the r , g , b value at pixel l_{vertex} . In case $[x_{vertex}, y_{vertex}, z_{vertex}]^T = 0$, the vertex p is considered to have no corresponding vertex on the target.

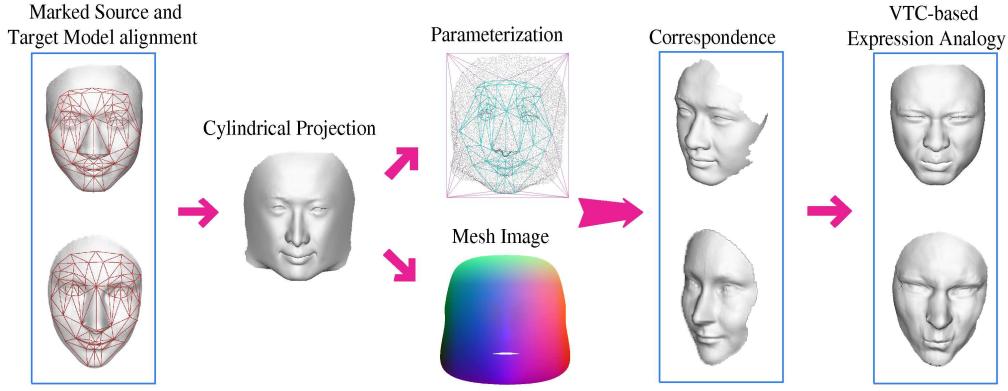


Fig. 3. The workflow chart of 3D expression analogy.

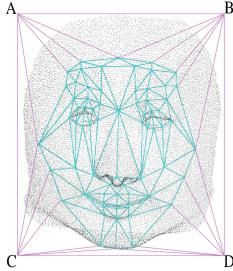


Fig. 6. Parameterization of source mesh by triangulating the marked feature points and the bounding box vertices A , B , C , and D in the cylindrical representation.

A, B, C, D are determined by the bounding box of the cylindrical coordinates of the 3D mesh. Since the target face is aligned to the source one, the barycentric coordinates would keep constant even though the target head motion happened. Consequently, the expression analogy result wouldn't be influenced.

4) *Topology Correction*: Also, in order to perform the deformation transfer, it is necessary to build the topology correspondence between the source and the target meshes, which lead to the VTC correspondence naturally for each vertex. Actually some of the vertices in the source mesh have not been assigned to a partner vertex in the target mesh. In terms of mesh correspondence, these non-matched source vertices are redundant. In order to make sure that the topologies of the source mesh and the resampled target mesh are identical, the redundant vertices and the adjacent triangles are removed from the source geometry. An example of topology correction is shown in Fig. 7. This process ensures that the VTCs for each vertex of the source and the target have been assigned a correspondence.

VI. 2D FACIAL EXPRESSION ANALOGY

Our generic expression transfer method can also be employed to map a facial expression from a source image of one person to a target image of another person. The luminance (Y in YUV color space) variations of the pixels reflect the changes of subtle expression details in the face [6]. In our approach, the face image is regarded as the 3D surface of a



Fig. 7. By means of topology correction between source mesh (left) and target mesh (right), the topologies of the triangles in the source mesh are reorganized and redundant vertices are deleted (middle).

height field whose height values are based on the luminance values. The transfer of these subtle expression details is modeled by the luminance transformation between the source neutral and source expression face image. This transformation can be computed and applied on the target face to obtain the same subtle expression details by our VTC-based method.

The inputs to our 2D facial analogy framework are therefore a photograph of a source neutral face, S , a photograph of a source expressive face, S' , and a photograph of a target neutral face, T . The output of the method is an image of the target face T' , in which the target subject mimics the expression depicted in S' . To this end, we need to transfer the change in the source face's shape between S and S' onto the target face image. However, for 2D expression analogy shape transfer alone is not sufficient. We also have to make sure that the differences in the source face's textural surface appearance between S and S' are correctly mapped onto the target face image. These variations in surface appearance across different facial expressions are mainly caused by local lighting changes due to skin deformation, e.g. in the vicinity of wrinkles. To produce the correct face shape and the correct surface colors in T' we combine our VTC-based deformation scheme with a geometric image warping approach. While the warping method generates the correct shape of the target face in the novel expression, our VTC-based deformation scheme is used to correctly reproduce the changed surface appearance. In contrast to previous methods, like expression ratio images [6], our method can faithfully transfer facial expressions even if the lighting conditions between the source and the target image are significantly different.

Before we can apply our VTC-based deformation transfer

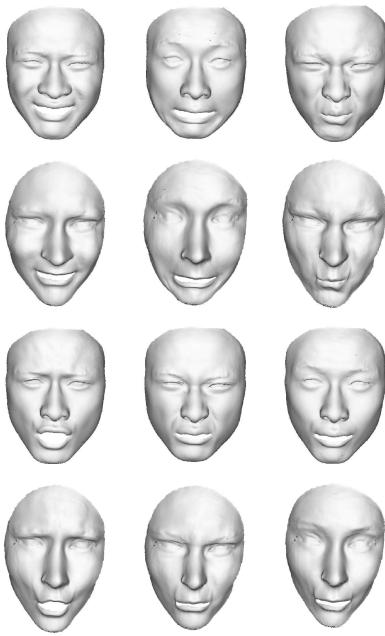


Fig. 8. Example results of 3D facial expression analogy: the expressions in the source face model (the first and third row) are realistically transferred to the different target face model (the second and fourth row) through VTC-based expression analogy. The density of the meshes: Row 1 & 3: 23725 vertices, 46853 triangles; Row 2 & 4: 19142 vertices, 37551 triangles.

approach to map the correct surface appearance to the target image, we need to transform all input images into 3D meshes. Moreover, in the 2D case we are confronted with the problem that the source and the target face are usually shown in different poses, and that their respective sizes in the images may differ.

Considering all the aforementioned issues, we suggest the following sequence of steps to perform 2D facial expression analogy:

- (1). Label the face feature points in S , S' and T .
- (2). Align the images S , S' and T based on the selected feature points by rotation, scale and translation.
- (3). Compute the motion vector of the feature points between S and S' . Perform geometric image-warping on the image T by using the motion vectors of the feature points between S and S' as warping constraints.
- (4). Transform S , S' and T into corresponding image grids, i.e. 3D triangle meshes, quadrangle meshes, etc. Build the correspondence between the source and target face 3D image grids.
- (5). VTC-based expression mapping is carried out to compute the pixels' luminance values in the warped target image. The final image T' is obtained by converting the luminance values back to RGB color space.

We would like to point out that step 1 consists of two substeps. The first substep automatically locates feature points by means of an active appearance model based tracking scheme (AAM). In the second substep, the user optionally adjusts the feature points' locations by a very few manual interactions. Apart from this, step 2, 3, 4, 5 are carried out automatically.

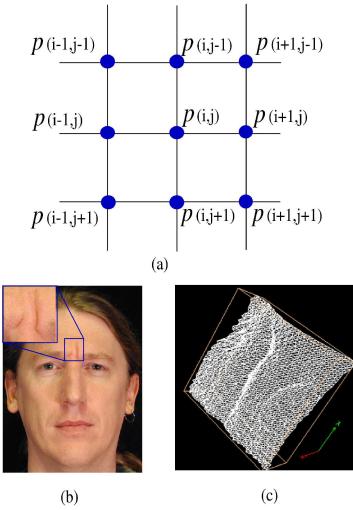


Fig. 9. (a) Triangulation of image pixels. (b) An enlarged region of an input image and the corresponding image grid (c).

A. 2D Images as Quadrangle Meshes

In order to make the face image data accessible to our VTC-based deformation scheme, we need to transform them into a 3D surface. The image can either be transformed into a triangle mesh or a quadrangle mesh. We found the latter method to be more convenient, as no explicit triangulation needs to be carried out. As opposed to 3D expression mapping, we don't intend to use the VTC-based deformation transfer to model the overall change of the target face's geometry. Instead, we employ it to accurately transfer the changes in the source face's textural appearance between S and S' onto the target face. Hence, our face representation needs to enable us to appropriately formulate per-pixel appearance variations. We thus propose the following method to transform each input image into a 3D image grid: First, the image is transformed into YUV color space. Now, as opposed to the triangulation manipulation in [28], the image pixels are treated as vertices on a quadrangle mesh, as it is illustrated in Fig. 9(a). Based on this mesh, we assign to each pixel $p(i, j)$ at image coordinates i and j a corresponding 3D vertex at 3D position $(i, j, l_{i,j})$ whose z coordinate equals the corresponding pixel's luminance value $l_{i,j}$. Fig. 9(b),(c) show the image grid.

B. Geometric Image Warping

We employ a geometric image warping technique to transfer the global change of the face's shape between the source neutral and the source expressive image onto the target image. To this end, we triangulate the locations of the marked feature points in the target neutral image T . The target neutral image is warped [34] by applying the motion of the feature points between S and S' to the corresponding feature points in T . We exploit the texture mapping capability of the GPU to render the complete warped target image very quickly.

C. Correspondence Finding

The geometric warping applies the coarse changes in the source face's shape between S and S' onto the target neutral

face. However, the warped target face does not yet look convincing since, so far, the changes in surface texture have not been considered. We make use of our VTC-based deformation transfer approach to compute the correct pixel values of the face in T' . To this end, we need to establish per-pixel correspondences between the source neutral and the (aligned) target neutral face image. This is the same as establishing per-vertex correspondences between the respective image grids. We thus resample the source image grid such that its topology becomes the same as the one of the target image grid. In other words, for each pixel in target image, i.e. each vertex in the target image grid, we find the corresponding pixel in the source image, i.e. the corresponding vertex in the source image grid.

D. Appearance Transfer

With these dense correspondences between source and target at hand, we can compute the pixel values in T' . This is straightforwardly achieved by formulating (4) in terms of the luminance components of all the involved image grids, and solving for the luminance values of deformed target image. The final color of the target expressive image is obtained by transforming the *YUV* colors back into *RGB* space.

In comparison to the algorithm proposed in [6] our method has a couple of intriguing advantages. First, we do not need to apply an adaptive Gaussian filter to correct artifacts caused by image registration errors. The pixel values in T' are computed globally and thus local misregistrations have a much smaller influence on the overall visual quality. Moreover, because the warping on the target image is carried out in advance, it is unnecessary to re-compute the i and j components of the vertices $(i, j, l_{i,j})$ of the target image repeatedly. The linear system only needs to be solved in the luminance component $l_{i,j}$ of the image grid.

In our implementation, the positions of the feature points close to the face contour are adjusted a bit to lie on the inside of the edge. This way, we can avoid that the contour of the source face is mapped onto the target face by mistake.

VII. RESULTS AND DISCUSSION

Our VTC-based expression analogy method produces compelling results on both 2D and 3D data. To validate our approach we compared it to well-known related techniques from the literature, namely Expression Cloning [14] and deformation transfer [15] in the 3D case, and expression ratio images (ERI) [6] in the 2D case.

Our test input data for 3D facial expression analogy comprise of a sequence of 300 frames that shows the dynamic face geometry of a performing human actor. The data were kindly provided to us by the authors of [9] who captured the footage with their real-time face scanner. We implemented expression cloning, deformation transfer and our VTC-based method to transfer this expression sequence onto three face scans of other test persons. It is our intent to clearly show the geometric detail in the 3D target animations. In particular in order to make the subtle deformation in the results better visible, we show our 3D results without textures. In principle, however, the application of a texture would be straightforward. A comparison of the

obtained results is shown in Fig. 10. As previously mentioned, basic facial expression cloning leads to unwanted geometry artifacts on the target face, e.g. close to the cheeks. In contrary, mesh-based deformation transfer [15] and our VTC-based method lead to visually more pleasing results. To evaluate the quality of the results obtained by the latter two methods, we mapped the expression to the source neutral face itself and calculate the per-vertex error (detailed error calculations are included in appendix). Due to different 3D surface encoding strategy and deformation transfer algorithm, even mapping the expression to itself, there are still different error values between different approaches. In Fig. 11, we utilize different colors to represent different error quantity levels. The depicted errors are in the range of < 0.1% (green), [0.1%, 0.5%) (blue) and [0.5%, 2.0%) (red). One can see that our VTC-based method leads to a high reconstruction accuracy in those parts of the face that carry most of the expressive detail, such as the vicinity of the eyes and the mouth. Furthermore, in Fig. 8 we show that VTC-based expression analogy convincingly maps source expressions to target subjects with widely different physiognomies. We would like to point out that during the expression analogy, the feature points enforce the structural coherence between the neutral face and the expression face. If the feature points corresponding to the cheek-bones are not located precisely, the cheek-bones are not preserved very well. Therefore, care has to be taken in general that the feature points are located appropriately. In the accompanying video¹, we show two complete animation sequences created with our approach that demonstrate the very natural look of the animated faces.

In Fig. 12, several results are shown that we obtained with our 2D facial expression analogy algorithm. They illustrate that we can convincingly transfer facial expressions between photographs of people with different gender and different physiognomies. Note that our image-based expression transfer algorithm nicely reproduces even subtle appearance details on the target expressive images, such as wrinkles on the forehead, that do not occur in the target neutral images. Moreover, we can make an image blink at you (second row in Fig. 12). We also show examples for 2D expression cloning, in which an expression is mapped in which the person opens the mouth (Fig. 13). Although the shape of the mouth and the wrinkles in the face are faithfully reproduced in the result images, the teeth are not. However, we don't consider this to be a principal limitation of our method as any image-based method suffers from it and it would be straightforward to apply a 2D teeth template to handle such a situation. We left this as a topic of future work, as this is not a core issue of our approach.

For validation, we compared our method to expression ratio images (ERI) in Fig. 14. In the ERI results, appearance details in the target expressive faces are sometimes blurred which is due to registration errors between the input images. Since our VTC-based method solves for the target pixel values globally, this problem is hardly noticeable in our results. In Fig. 15,

¹Accompanying video is compressed by 3ivx D4 4.5.1 codec and it can be played normally with Quicktime player 6.x (free). In case it can NOT be played, please download the codec software from <http://www.3ivx.com/download/>

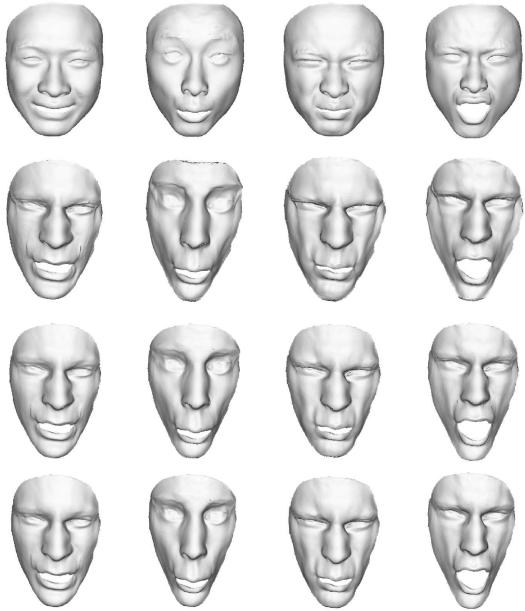


Fig. 10. Comparison of two 3D facial analogy approaches: Source meshes (top row), results obtained with basic expression cloning(2nd row), results obtained with deformation transfer (3rd row), and results obtained with our VTC-based 3D expression analogy algorithm. Row 1: 23725 vertices, 46853 triangles; Row 2-4: 16981 vertices, 33226 triangles.

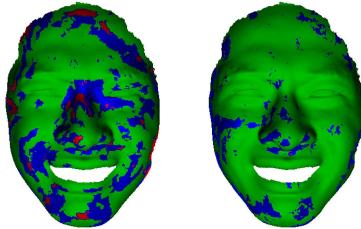
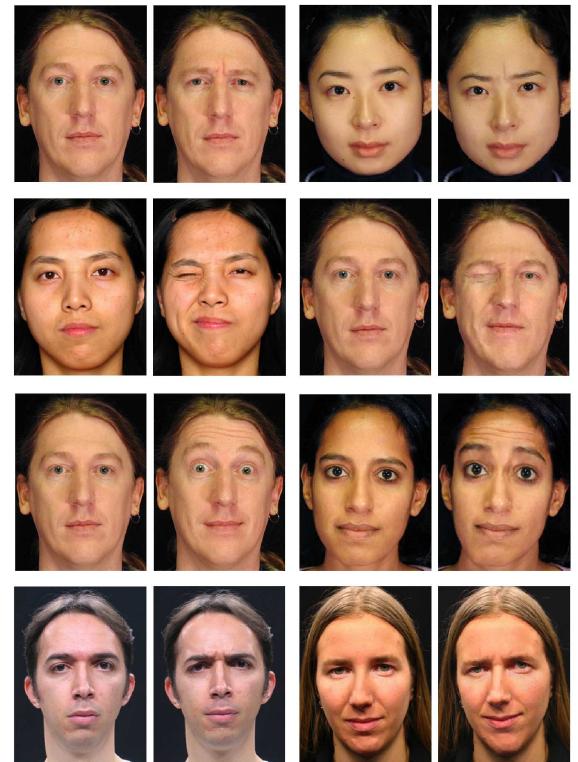


Fig. 11. Error characteristic and distribution: Deformation transfer based method (left) and VTC-based method (right). Different color represent different error quantity level: low (Green), middle (Blue) and high (Red)

we mapped the source expression back to the source neutral image using both ERI and our algorithm in order to assess their respective robustness. In the ideal case, the resulting image exactly matches the source expressive image. While the result produced by our algorithm closely matches the source expressive image, Fig. 15(a), the ERI result exhibits clearly noticeable artifacts, e.g. close to the eyebrow. We'd also like to point out that our method faithfully transfers expressions even if the lighting conditions in the source and target images are different. In the accompanying video we show a few more image-based results, and also demonstrate that we can make an image mimic a complete input video sequence of an actor.

We have measured the average CPU times of our 3D and 2D approaches on a Pentium IV 3.0 GHz with 512 MB of memory. Please note that for computational efficiency we always perform a LU decomposition in a preprocessing step and only measure the time needed to evaluate the right-hand side of our equation systems and perform back-substitution. With our

Fig. 12. VTC-based 2D expression analogy: The left image pair in each row of faces shows a source neutral image (l) and a source expressive image (r). The right image pair in each row shows a target neutral image (l) of another person that we made convincingly mimic (r) the expression of the source subject in the same row.

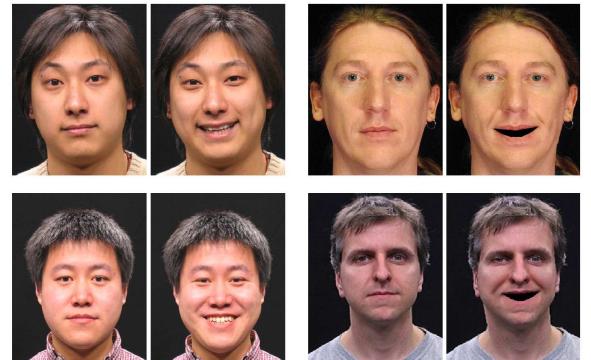


Fig. 13. Results of VTC-based 2D expression analogy when the person in the target expression image opens the mouth. Our method cannot correctly reproduce the appearance of the teeth but it would be straightforward to implant a 2D teeth template to handle this situation.

3D models comprising of roughly 19,000 vertices and 37,500 triangles it takes around 1.1 s to transfer one expression. Moreover, our correspondence finding method only takes 1.5 s which is significantly faster than the hour magnitude needed by the correspondence finding of deformation transfer on the same data set. In the 2D case, our method transfers an expression between images of 311×419 pixels in around 1.3 s while ERI takes 6.3 s on the same input data.

To summarize, our generic facial analogy approach provides



Fig. 14. Comparison of two 2D facial analogy methods: In each row the source expression (left column) is mapped onto an image of another person by means of expression ratio images (ERI) (middle column), as well as by means of our VTC-based approach (right column). While our method reproduces face details sharper than that of ERI results, which are blurred in some regions, e.g. around the cheek in the third row.

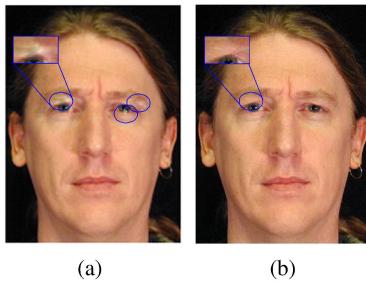


Fig. 15. Robustness check: When mapping the source expression back onto the source neutral image, the original source expressive image should be reproduced as good as possible. However, the results obtained with ERI exhibit clearly visible artifacts, e.g. close to the eyebrow (a). Our method, in contrast, reproduces this image region very accurately which demonstrates its higher robustness (b).

animation professionals with a powerful tool to animate both 2D and 3D images from example expressions.

VIII. CONCLUSION AND FUTURE WORK

We have presented a novel generic facial expression analogy technique. It enables us to convincingly transfer facial expressions between 3D face models, as well as between 2D face images. In both the 3D and the 2D cases our animated faces look very natural and correctly mimic even subtle expression details in the source faces. Moreover, our method is faster than most related methods from the literature and also outperforms them in terms of the achieved visual quality.

In future, we plan to explore if we can further reduce the computation time by combining our method with a machine learning approach. We would like to improve the computing efficiency by capitalizing on concepts presented in [26]. Furthermore, we intend to work on an fully automatic scheme to detect facial features on both images and 3D meshes more

precisely. It seems also promising to apply our framework to more general image transformation operations, for instance body pose analogy or more complex image warping operations.

IX. ACKNOWLEDGEMENT

This research was partly supported by the China Postdoctoral Science Foundation (20060401040), Microsoft Joint Laboratory Research Foundation and the National Science Foundation of China (60502006). The authors also would like to thank Yi Chai, Natascha Sauber, Edilson de Aguiar, Kuangyu Shi, Wenxiang Ying, and Michael Neff for their assistance. Thanks Li Zhang and Xianfeng Gu for their 3D face data. Many thanks to anonymous reviewers and Associate Editor Prof. Horace H.S.Ip for their works and dedications.

APPENDIX

To calculate the per-vertex error, we regard the motion vector of each vertex between the source neutral and source expressive faces as ground truth. We map the expression to the source neutral face itself and then evaluate the error by the following formula:

$$E = \left| \frac{L_2(V_{TE} - V_{TN})}{L_2(V_{SE} - V_{SN})} - 1 \right|$$

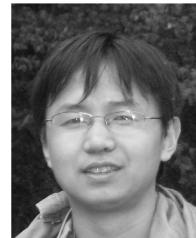
$$= \left| \frac{L_2(V_{TE} - V_{SN})}{L_2(V_{SE} - V_{SN})} - 1 \right|$$

Where $V_{SN}, V_{SE}, V_{TN}, V_{TE}$ represent the coordinates of any vertex in source neutral, source expressive, target neutral and target expressive face respectively. Here $V_{TN} = V_{SN}$ for the source neutral model is just the target neutral model. $L_2(\cdot)$ represents the Euclidean distance.

REFERENCES

- [1] S. B. Göktürk, C. Tomasi, B. Girod, and J.-Y. Bouguet, "Model-based face tracking for view-independent facial expression recognition," in *Proc. 5th IEEE International Conference on Automatic Face and Gesture Recognition*. IEEE Computer Society, 2002, pp. 287–293.
- [2] K. Waters, "A muscle model for animating three dimensional facial expression," *Computer Graphics*, vol. 21, no. 4, pp. 17–24, 1987.
- [3] Y. Lee, D. Terzopoulos, and K. Walters, "Realistic modelling for facial animation," in *SIGGRAPH '95: Proc. of the 22nd annual conference on Computer graphics and interactive techniques*. New York, NY, USA: ACM Press, 1995, pp. 55–62.
- [4] H. Pyun, Y. Kim, W. Chae, H. W. Kang, and S. Y. Shin, "An example-based approach for facial expression cloning," in *SCA '03: Proc. of the 2003 ACM SIGGRAPH/Eurographics symposium on computer animation*, 2003, pp. 167–176.
- [5] K. Kähler, J. Haber, and H. P. Seidel, "Geometry-based muscle modeling for facial animation," in *Proc. Graphics Interface*, 2001, pp. 37–46.
- [6] Z. Liu, Y. Shan, and Z. Zhang, "Expressive expression mapping with ratio images," in *SIGGRAPH '01: Proc. of the 28th annual conference on Computer graphics and interactive techniques*. New York, NY, USA: ACM Press, 2001, pp. 271–276.
- [7] Q. Zhang, 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.
- [8] E. Cosatto and H. P. Graf, "Photo-realistic talking-heads from image samples," *IEEE Transactions on Multimedia*, vol. 2, no. 3, pp. 152–163, 2000.
- [9] L. Zhang, N. Snavely, 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.

- [10] Y. Wang, X. Huang, C.-S. Lee, S. Zhang, Z. Li, D. Samaras, D. Metaxas, A. Elgammal, and P. Huang, "High resolution acquisition, learning and transfer of dynamic 3-d facial expressions," *Computer Graphics Forum*, vol. 23, no. 3, pp. 677–686, 2004.
- [11] P. J. Phillips, H. Moon, S. A. Rizvi, and P. J. Rauss, "The feret evaluation methodology for face recognition algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 1090–1104, 2000.
- [12] L. Williams, "Performance-driven facial animation," in *SIGGRAPH '90: Proc. of the 17th annual conference on Computer graphics and interactive techniques*. New York, NY, USA: ACM Press, 1990, pp. 235–242.
- [13] F. Pighin, J. Hecker, 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.
- [14] J. Noh and U. Neumann, "Expression cloning," in *SIGGRAPH '01: Proc. of the 28th annual conference on Computer graphics and interactive techniques*, 2001, pp. 277–288.
- [15] R. W. Sumner and J. Popovic, "Deformation transfer for triangle meshes," vol. 23, no. 3. New York, NY, USA: ACM Press, 2004, pp. 399–405.
- [16] O. Sorkine, D. Cohen-Or, Y. Lipman, M. Alexa, C. Rössl, and H. P. Seidel, "Laplacian surface editing," in *Proc. of the Eurographics/ACM SIGGRAPH symposium on Geometry processing*. Eurographics Association, 2004, pp. 179–188.
- [17] F. I. Parke and K. Waters, *Computer facial animation*. Wellesley, Mass.: AK Peters, 1998.
- [18] S. E. Brennan, "Caricature generator: Ms visual studies," *MIT Cambridge*, 1982.
- [19] P. Litwinowicz and L. Williams, "Animating images with drawings," in *SIGGRAPH '94: Proc. of the 21st annual conference on Computer graphics and interactive techniques*. New York, NY, USA: ACM Press, 1994, pp. 409–412.
- [20] P. Ekman and W. V. Friesen, *Facial action coding system: a technique for the measurement of facial movement*. Palo Alto, CA.: Consulting Psychologists Press, 1978.
- [21] J. Ostermann, "Animation of synthetic faces in mpeg-4," in *Proc. Computer Animation*, 1998, pp. 49–51.
- [22] W. Gao, Y. Chen, R. Wang, S. Shan, and D. Jiang, "Learning and synthesizing mpeg-4 compatible 3-d face animation from video sequence," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 11, pp. 1119–1128, 2003.
- [23] B. Park, H. Chung, T. Nishita, and S. Y. Shin, "A feature-based approach to facial expression cloning," *Journal of Computer Animation and Virtual Worlds*, vol. 16, no. 3–4, pp. 291–303, 2005.
- [24] Y. Yu, K. Zhou, D. Xu, X. Shi, H. Bao, B. Guo, and H. Y. Shum, "Mesh editing with poisson-based gradient field manipulation," *ACM Transaction on Graphics*, vol. 23, no. 3, pp. 644–651, 2004.
- [25] R. Zayer, C. Rössl, Z. Karni, and H.-P. Seidel, "Harmonic guidance for surface deformation," *Computer Graphics Forum*, vol. 24, no. 3, pp. 601–609, 2005.
- [26] M. Botsch, R. Sumner, M. Pauly, and M. Gross, "Deformation transfer for detail-preserving surface editing," in *Proc. of Vision, modeling, and visualization (VMV)*, 2006, pp. 357–364.
- [27] V. Blanz, K. Scherbaum, T. Vetter, and H. P. Seidel, "Exchanging faces in images," *Computer Graphics Forum*, vol. 23, no. 3, pp. 669–676, 2004.
- [28] M. Song, H. Wang, J. Bu, C. Chen, and Z. Liu, "Subtle facial expression modeling with vector field decomposition," in *Proc. of International Conference on Image Processing (ICIP)*, 2006, pp. 2101–2104.
- [29] T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Active appearance models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, pp. 681–685, 2001.
- [30] A. Sheffer and V. Kraevoy, "Pyramid coordinates for morphing and deformation," in *Proc. of International symposium on 3D data processing, Visualization and Transmission*, 2004, pp. 68–75.
- [31] W. Gautschi, *Numerical Analysis*. Birkhauser, 1997.
- [32] J. H. Challis, "A procedure for determining rigid body transformation parameters," *Journal of Biomechanics*, vol. 28, no. 6, pp. 733–737, 1995.
- [33] X. Gu, S. J. Gortler, and H. Hoppe, "Geometry images," in *SIGGRAPH '02: Proc. of the 29th annual conference on Computer graphics and interactive techniques*. New York, NY, USA: ACM Press, 2002, pp. 355–361.
- [34] T. Beier and S. Neely, "Feature-based image metamorphosis," in *SIGGRAPH '92: Proc. of the 19th annual conference on Computer graphics and interactive techniques*. New York, NY, USA: ACM Press, 1992, pp. 35–42.



Mingli Song is a researcher in Microsoft Visual Perception Laboratory, Zhejiang University. He received the PhD degree in computer science from Zhejiang University, China, in 2006. His research interests include face modeling and facial expression analysis. He is a member of the IEEE.



Zhao Dong is currently working as a PhD student in the computer graphics group of the Max-Planck-Institut (MPI) Informatik, Saarbrücken, Germany. His research interests include Real-time Image Synthesis, Realistic Hardware-Supported Shading and Appearance Editing. He is a student member of the IEEE.



Christian Theobalt is a postdoctoral researcher and research team leader in the computer graphics group of the Max-Planck-Institut (MPI) Informatik, Germany. He received his MSc degree in Artificial Intelligence from the University of Edinburgh, Scotland, and his Diplom (MS) degree in Computer Science from Saarland University, Saarbrücken, Germany, in 2000 and 2001 respectively. From 2001 to 2005 he worked as a PhD student and research associate in Hans-Peter Seidel's computer graphics group at MPI Informatik. In 2005, he received his PhD (Dr.-Ing.) from Saarland University. His research interests include free-viewpoint and 3D video, marker-less optical motion capture, 3D computer vision, and image-based rendering. He is a member of the IEEE.



Huiqiong Wang is a PhD candidate in the department of information science and electronic engineering, Zhejiang University. Her research interests include computer vision, face modeling and facial expression analysis.



Zicheng Liu received the PhD degree in computer science from Princeton University, the MS degree in operational research from the Institute of Applied Mathematics, Chinese Academy of Sciences, Beijing, China, and the BS degree in mathematics from Huazhong Normal University, Wuhan, China. He is currently a researcher at Microsoft Research. Before joining Microsoft, he worked as a member of technical staff at Silicon Graphics focusing on trimmed NURBS tessellation for CAD model visualization. His research interests include 3D face modeling and facial animation, linked figure animation, multisensory speech enhancement, and multimedia signal processing. He was the cochair of the IEEE International Workshop on Multimedia Technologies in E-Learning and Collaboration in 2003. He is a senior member of the IEEE.



Hans-Peter Seidel is the scientific director and chair of the computer graphics group at the Max-Planck-Institut (MPI) Informatik and a professor of computer science at Saarland University. He has published some 200 technical papers in the field and has lectured widely. He has received grants from a wide range of organizations, including the German National Science Foundation (DFG), the German Federal Government (BMBF), the European Community (EU), NATO, and the German-Israel Foundation (GIF). In 2003 Seidel was awarded the 'Leibniz Preis', the most prestigious German research award, from the German Research Foundation (DFG). Seidel is the first computer graphics researcher to receive this award.