Emotion-Preserving Blendshape Update With Real-Time Face Tracking

Zhibo Wang, Jingwang Ling, Chengzeng Feng, Ming Lu, and Feng Xu

Abstract—Blendshape representations are widely used in facial animation. Consistent semantics must be maintained for all the blendshapes to build the blendshapes of one character. However, this is difficult for real characters because the face shape of the same semantics varies significantly across identities. Previous studies have handled this issue by asking users to perform a set of predefined expressions with specified semantics. We observe that facial emotions can be used to define semantics. Herein, we propose a real-time technique that directly updates blendshapes without predefined expressions. Its aim is to preserve semantics based on the emotion information extracted from an arbitrary facial motion sequence. In addition, we have designed corresponding algorithms to efficiently update blendshapes with large- and middle-scale face shapes and fine-scale facial details, such as wrinkles, in a real-time face tracking system. The experimental results indicate that using a commodity RGBD sensor, we can achieve real-time online blendshape updates with well-preserved semantics and user-specific facial features and details.

Index Terms—Facial animation, real-time tracking, blendshape animation

1 Introduction

Blendshape representation is widely used in facial animation in both academia and industry. The ideal blendshapes of different characters require consistent semantics for each blendshape bases, which is critical for high-quality facial animations (e.g., expression retargeting). Without consistent semantics, the same set of blendshape coefficients does not indicate the same semantic expression across different characters, introducing problems when generating animation. Therefore, the blendshapes of a virtual avatar must be designed carefully, including via tedious manual modeling and editing. Apart from the avatars, the generation of the blendshapes of real humans has attracted increasing attention in recent years because of the importance associated with tracking and animating personalized real humans. However, the blendshapes of real humans have an additional requirement, i.e., the personalities of different identities have to be matched. Thus, additional difficulties are associated with the generation of blendshapes.

Techniques have been proposed to generate personalized blendshapes of real humans. Some studies focused on applying high-quality face tracking or re-enactment [1], [2], [3], [4], [5], and these techniques optimized user-specific blendshapes to achieve optimal matching with respect to the input depth, color, and facial landmark data in a motion sequence. Thus, the blendshapes cover the motion space in a sequence very well. However, there is an inherent ambiguity, i.e., tuning the blendshapes and the coefficients could both fit the input. Therefore, the semantics of the blendshapes may be incorrectly altered when using some incorrect blendshape coefficients. The fitting error could still be very low in such cases because the two types of errors can compensate each other. Thus, the blendshape semantics may get mixed [6]. To solve this problem, other techniques record predefined expressions with known/initial blendshape coefficients [6], [7], [8]. Here, correct or reasonable constraints can be obtained using the blendshape coefficients. Thus, the ambiguity can be solved. However, such predefined expressions require additional effort from experienced artists, increasing the complexity associated with the blendshape generation pipeline.

Thus, we propose a method that bridges the two techniques to achieve online semantic-preserving blendshape generation for real humans. Our key observation is that the incorrect updating of blendshapes can be avoided by constraining the emotions of the reconstructed faces instead of directly constraining the blendshape coefficients, solving the aforementioned ambiguity. Unlike the predefined blendshape coefficients, emotion information can be extracted directly from the color images to constrain the blendshape coefficients in blendshape generation.

Specifically, we train the mapping from blendshape coefficients to emotions. The "ground truth" emotions can be extracted from the recorded frames. Thus, the blendshape coefficients are constrained to match the "ground truth" emotions through mapping. Hence, the updated blendshapes maintain their original semantics. Here, we do not use an emotion label because emotion labels are considerably sparse for completely constraining the blendshape coefficients. Instead, we employ an emotion feature, i.e., the intermediate feature obtained using an emotion classification network. This emotion feature is considered to be more sensitive to subtle emotion differences and contributes more to constraining blendshape coefficients.

In addition to the semantics, we decompose the facial geometry into large, middle, and fine scales. These three scales

• *The authors are with the BNRist and Department of Electronic Engineering, Tsinghua University, Beijing 100084, China. E-mail: {wzb17,lingjw16,fcz18, lu-m13}@mails.tsinghua.edu.cn, feng-xu@tsinghua.edu.cn.*

Manuscript received 31 Mar. 2020; revised 13 Oct. 2020; accepted 19 Oct. 2020. Date of publication 26 Oct. 2020; date of current version 2 May 2022.
(Corresponding author: Feng Xu.)
Recommended for acceptance by R. Zhang.
Digital Object Identifier no. 10.1109/TVCG.2020.3033838

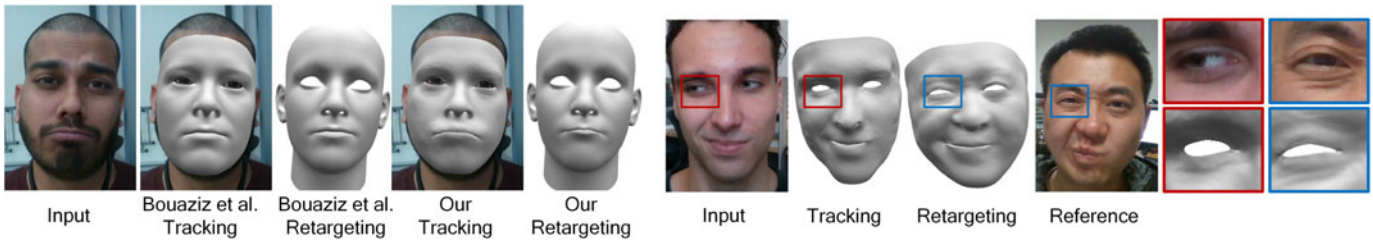

Fig. 1. The proposed blendshape update technique can maintain the semantics of an expression and capture user-specific facial features. On the left, we compare our updated blendshapes with those of [1] by fitting an input image and transferring the coefficients to a template. Our blendshapes appropriately reconstruct the semantics of the expression using the subtle mouth pose, and the tracked expression is correctly transferred to the template. On the right, we transfer an expression from one real human to another. The user-specific wrinkles of the source and target are captured by our blendshapes.

are subsequently reconstructed in an efficient online manner for blendshape update to capture the user-specific facial detail, thereby achieving high-fidelity blendshape generation. We propose a blendshape update strategy to update the three scales on the blendshapes, which is compatible with the commonly used blendshape-based facial animation systems. In addition, we consider and develop depth fusion techniques to integrate multiframe information for suppressing the depth noise and refining template-to-depth correspondences. We also develop a real-time "Shape from Shading" (SfS) component to achieve fine-scale tracking and updating with color inputs. Based on these components, our technique can preserve the blendshape semantics and recover the user-specific expression features and wrinkles, as shown in Fig. 1. Our primary contributions are summarized as follows.

- We present an online blendshape update system that provides high-fidelity real-time face tracking.
- We present an emotion control method that preserves the emotion semantics of blendshape bases.
- We present an online three-scale blendshape update strategy that updates blendshapes with user-specific facial details.

2 Related Work

2.1 3D Face Modeling

Many methods have been proposed to reconstruct the 3D models of human faces [9]. These methods can be classified as parametric-based and parametric-free methods. The parametric-based methods initially learn low-dimensional statistical models from the preprocessed datasets. Then, they can reconstruct the 3D face by finding the optimal parameters of the models. [10] is a pioneering work related to 3D morphable models. It learns the linear models of identity and albedo based on the scanned faces. [11] constructs a large dataset of different identities and expressions and learns a high-order linear model from the constructed dataset. [12] learns the skinned linear model [13] from both static and dynamic facial scans. [14] uses a combination of local PCA submodels to achieve improved expressiveness. [15] improves robustness by placing an anatomical constraint on the local blendshape face model. Many deep parametric models have been proposed relative to deep neural networks (DNNs). For example, [16] learned an identity and appearance model from in-the-wild video clips, whereas [17] used a graph convolution network (GCN) to learn the parametric model from a previously reported dataset [12]. DNNs are highly nonlinear; thus, they can better learn lower-dimensional models when compared with linear models. Instead of training the GCN based on the positions of vertices, [18] trains the GCN based on an "As-Consistent-As-Possible" representation. Thus, the nonlinearity of the GCN is further improved. Even though many deep models have been proposed, traditional linear models have been employed in this study because such models continue to dominate the current face tracking and animation systems owing to their ease of use and intuitiveness in case of artists.

Compared with the parametric-based methods, the 3D geometry of a human face can be directly reconstructed using parametric-free methods. [19], [20], [21] used multiview geometry to reconstruct a detailed 3D face from multiple images captured from different views. [22] fused the frames captured by a depth sensor to reconstruct a 3D model; however, this method cannot reconstruct the geometry details owing to the noisy depth data. The proposed technique uses a consumer-level depth camera similar to that used in [22]. In the proposed technique, the profile of each user is dynamically updated and the expression of the user is tracked in real time instead of building the user's profile by tedious scanning. Warping-field-based deformation helps to achieve good template-to-depth correspondences; therefore, depth noises are filtered when more frames are considered.

2.2 3D Face Tracking

Traditional 3D face tracking methods can be roughly classified as parametric-based, parametric-free, and hybrid methods. The parametric-based methods, such as [7], reconstruct the large-scale face motion in each frame by directly optimizing the coefficients in the parametric models. High-quality face tracking can be achieved using some parametric-free methods. For example, [23], [24] captured high-fidelity face geometry using a multiview stereo and face motion using a propagation technique. The remaining methods combine these two solutions by adding parametric-free deformation to a parametric-based face model. On top of a parametric-based face model, [25], [26], [27] added a fine-scale layer based on which the high-frequency fine-scale geometry of a face can be described. [25], [27] reconstructed the fine-scale details using SfS. In addition, [26] used a boosted regressor trained via high-quality face scans to obtain visually convincing wrinkles, and [28] added middle-scale deformation between large- and fine-scale deformations to capture the geometric semantics of the user. Recently, DNNs have been widely used in the face tracking field. For example, [29] trained a deep convolutional autoencoder that can reconstruct a 3D face from a single in-the-wild

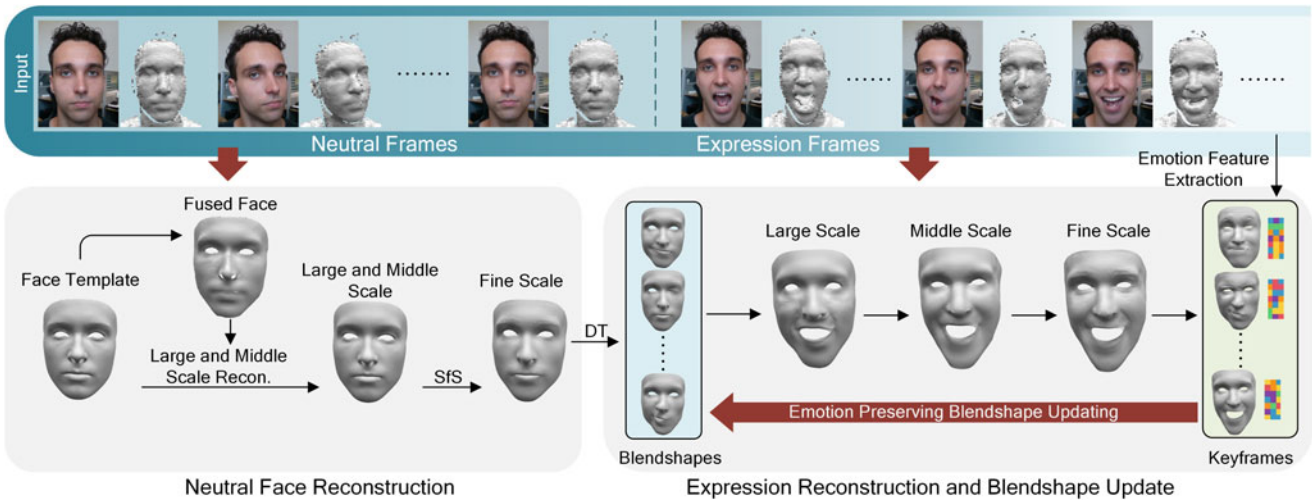

Fig. 2. Pipeline of the proposed method. The proposed method includes a neutral face reconstruction stage and an online tracking and blendshape updating stage. The first stage reconstructs the neutral face of the user with a short segment of the color and depth sequence, which covers different head poses. Then, deformation transfer is applied to obtain the initial blendshapes for the user. In the second stage, expression tracking and emotion-preserving blendshape updating are performed in real time. The face reconstruction algorithm used in these two stages is described in Section 5. The blendshape update algorithm is introduced in Section 6.

facial image, and [30] directly regressed the coefficients in parametric models from RGB images at more than 250 Hz.

2.3 3D Facial Rigging

The objective of facial rigging is to reconstruct the blendshapes from training poses. Based on the neutral face of the user, [31] can automatically generate the blendshapes of the user via deformation transfer (DT) according to a set of generic models. [6] extended the formulation of [31] to more training poses instead of using only the neutral face of the user. Several user training poses can be scanned and jointly optimized to improve the blendshapes of the user. Instead of improving the original blendshapes, [2] added an orthogonal motion space to realize better face fitting, which can be obtained via online tracking. [5] added middle- and fine-scale facial geometry to the blendshapes of the user using a regression model. Thus, the user's facial rig can be recovered from a video sequence; however, this method is offline and requires a considerable amount of time. [1] used a Laplacian matrix to represent the middle-scale deformation and developed the user's facial rig online; however, the detail scale of the facial rig could not be captured in this method. [8] achieved high quality. However, a predefined expression had to be recorded and the method was operated offline. Compared to such existing methods, the proposed method can build a detailed rig for users in real time.

3 Overview

In Fig. 2, we present the frameworks of the proposed technique. As can be seen, the proposed technique involves two stages. In the first stage, the user is asked to rotate their face with a neutral expression; we reconstruct the neutral face from this sequence. The blendshape model is obtained from the neutral face using DT [31]. In the second stage, we simultaneously reconstruct different user expressions in real time and apply an online blendshape update algorithm. Subsequently, we introduce the two main algorithms used in the proposed method, i.e., a high-fidelity face tracking algorithm used in both stages and an emotion-constrained blendshape update algorithm used in only the second stage. First, we introduce the three-scale face geometry representation in Section 4. Then, the objective functions and optimization strategies of the face reconstruction algorithm are presented in Section 5. We describe the emotion-constrained blendshape update algorithm in Section 6.

4 Preliminaries

As mentioned previously, the face geometry is decomposed into large, middle, and fine scales in the proposed technique. The large-scale shape and expression changes are represented by two linear statistical models, and the middle scale is a node-driven warping field that generates global nonlinear and local linear facial motions to provide greater flexibility for deformations on top of the large-scale motions. The fine scale is a per-vertex displacement to fit the normal information obtained using SfS, which represents the wrinkles and folds on faces.

Large Scale. In the proposed technique, linear morphable models are used to describe the large-scale deformation of faces. Here, an identity PCA model $B_{\mathrm{id}} \in \mathbb{R}^{3N \times M_{\mathrm{id}}}$ is applied to represent neutral faces with different identities. We obtain a face with neutral expression as $b_n = b_\mu + B_{\mathrm{id}}\boldsymbol{\alpha}$ given a set of coefficients $\boldsymbol{\alpha} \in \mathbb{R}^{M_{\mathrm{id}}}$ and the mean face $b_\mu \in \mathbb{R}^{3N}$. Similarly, a linear blendshape model is applied to represent expressions. Here, b_n denotes the user-specific neutral face, and $B_{\mathrm{exp}} \in \mathbb{R}^{3N \times M_{\mathrm{exp}}}$ denotes the blendshape model; thus, a face with large-scale deformation can be formulated as $b_{\boldsymbol{\beta}} = b_n + B_{\mathrm{exp}}\boldsymbol{\beta}$, where $\boldsymbol{\beta} \in \mathbb{R}^{M_{\mathrm{exp}}}$ represents the blendshape coefficients. Our objective is to generate B_{exp} for each user rather than directly using the template B_{exp} or DT to obtain B_{exp} without considering the user-specific dynamics.

Middle Scale. The middle-scale deformation is represented by a warping field $\mathcal{W}$. Specifically, $\mathcal{W} = \{[\mathbf{p}_j \in \mathbb{R}^3, T_j = \{R_j, \mathbf{t}_j\} \in SE(3)]\}$. Here, j denotes the index of a node in deformation graph $\mathcal{G}$. $\mathbf{p}_j$ is its 3D position, and T_j involves

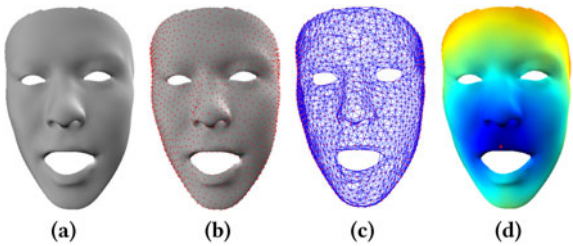

Fig. 3. The node graph and geodesic distance on a 3D surface. On a template face (a), we uniformly sample the nodes (b). In face tracking, the node structure (c) is determined by the geodesic distance between nodes. The geodesic distance between the red node and other vertices in the mesh is color-coded in (d).

the rotation R_j and translation $\mathbf{t}_j$ of the node. We use the notation $\mathcal{T}_i$ to represent the transformation of the vertex $\mathbf{v}_i$ on the face, which is a linear combination of the transformations of its nearest four nodes

$$\mathcal{T}_i\mathbf{v}_i = \sum_{j\in\mathcal{N}_v^i} w_{i,j}T_j\mathbf{v}_i, \tag{1}$$

where $\mathcal{N}_v^i$ contains the four nearest nodes of $\mathbf{v}_i$. We use the uniform distributed nodes on the mesh generated by the blue noise sampling algorithm [32]. The node graph is constructed by connecting the eight nearest neighbors of each node. We sample 1,200 nodes on a face in the proposed technique to fulfill the deformation ability of the warping field. Here, weights $w_{i,j}$ are determined by the geodesic distance $d(\mathbf{v}_i, \mathbf{p}_j)$ and can be defined as $w_{i,j} = \exp(-d(\mathbf{v}_i, \mathbf{p}_j)^2/r^2)$, where r is a constant. The weights $w_{i,j}$ and the node graph are precomputed on the neutral face. Details about the warping field are presented in Fig. 3.

Fine Scale. The fine-scale deformation is represented by the displacements of all the vertices on a face mesh. The displacement is obtained via SfS. Here, the shading irradiance can be expressed as follows:

$$S(i) = \boldsymbol{\rho}_i\mathbf{l}^T H(\mathbf{n}_i), \tag{2}$$

where i is the index of the vertex $\mathbf{v}_i$, ρ_i is the albedo of the vertex, and $\mathbf{n}_i$ is the normal of the vertex. $\mathbf{l}$ is the spherical harmonics (SH) coefficient vector [33], and H is the SH basis function of the first two orders. We employ a PCA albedo model $\boldsymbol{\rho} = \boldsymbol{\rho}_n + B_{\text{alb}}\boldsymbol{\gamma}$, where $\boldsymbol{\rho}_n \in \mathbb{R}^{3N}$ is the mean albedo, $B_{\text{alb}} \in \mathbb{R}^{3N\times M_{\text{alb}}}$ is the PCA albedo basis, and $\boldsymbol{\gamma} \in \mathbb{R}^{M_{\text{alb}}}$ represents the albedo coefficients. N is the number of vertices.

5 Face Reconstruction

In this section, we describe how to use the aforementioned three-scale motion model to reconstruct the high-fidelity neutral shape b_0 and all the expressions b_ws in the sequence with the given template topology.

5.1 Optimization Model

The general problem associated with face reconstruction is to solve the parameters in the three scales based on a recorded color and depth image.

Large and Middle Scale. The large- and middle-scale reconstructions can be formulated as follows:

$$\arg\min_{R,\mathbf{t},\boldsymbol{\alpha},\boldsymbol{\beta},\mathcal{W}} E_d + \lambda_{\text{lm}}E_{\text{lm}} + \lambda_{\text{id}}E_{\text{id}} + \lambda_{L1}E_{L1} + \lambda_s E_s, \tag{3}$$

where E_d and E_{lm} are the data terms that constrain the deformed face to fit the input depth points and the detected facial landmarks, respectively. E_{id}, E_{L1}, and E_s are the regularization terms that regularize the identity, expression coefficients, and warping field, respectively. Here, the global rigid transformation of the face involves rotation R and translation $\mathbf{t}$.

We use the point-to-plane loss [34] to define E_d as follows:

$$E_d = \sum_{(\mathbf{v}_i,\mathbf{d}_i)\in\mathcal{P}_d} \left(\mathbf{n}_i^T(R\mathbf{v}_i + \mathbf{t} - \mathbf{d}_i)\right)^2, \tag{4}$$

where $\mathbf{d}_i$ is the closest depth point of $\mathbf{v}_i$ and $\mathbf{n}_i$ is the normal direction of $\mathbf{d}_i$. The set $\mathcal{P}_d$ contains all pairs of correspondences $(\mathbf{v}_i, \mathbf{d}_i)$. Here, $\mathbf{v}_i$ is determined by the three-scale representation parameters that are to be solved in Eq. (3). $\{\boldsymbol{\alpha}, \boldsymbol{\beta}, \mathcal{W}\}$.

On the template face, we predefine some vertices corresponding to the detected 2D landmarks [35]. Here, the fitting error can be defined as follows:

$$E_{\text{lm}} = \sum_{(\mathbf{v}_i,\boldsymbol{f}_i)\in\mathcal{P}_f} \|\Pi(R\mathbf{v}_i + \mathbf{t}) - \mathbf{f}_i\|^2, \tag{5}$$

where $\mathbf{f}_i$ represents the detected landmarks and Π denotes the 3D to 2D projection. Set $\mathcal{P}_f$ includes all the visible pairs.

Following the literature [36], we employ the regularization term E_{id} to constrain the identity coefficients $\boldsymbol{\alpha}$ as follows:

$$E_{\text{id}} = \sum_i^{M_{\text{id}}} \left(\frac{\alpha_i}{\sigma_i^{\text{id}}}\right)^2, \tag{6}$$

where σ_{id} is the eigenvalue of the ith PCA basis.

The sparse term can be given as follows:

$$E_{L1} = \|\boldsymbol{\beta}\|_1, \tag{7}$$

which ensures that only a few action units (AU) are activated for each frame to ensure clear semantics.

To regularize the warping field, we use the As-Rigid-As-Possible (ARAP) term E_s [37] as follows:

$$E_s = \sum_{i=1}^{K}\sum_{j\in\mathcal{N}_p^i} \|T_i\mathbf{p}_i - T_j\mathbf{p}_i\|^2, \tag{8}$$

where $\mathcal{N}_p^i$ contains the eight nearest neighbors of the ith node.

The variables $R, \mathbf{t}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \mathcal{W}$ are separately updated during each iteration. During optimization, we first solve the rigid transformation of R, $\mathbf{t}$ using a Gaussian Newton (GN) solver implemented on a GPU. Then, with fixed R, $\mathbf{t}$, we only optimize the identity coefficients $\boldsymbol{\alpha}$ or expression coefficients $\boldsymbol{\beta}$ in the first few iterations using a conjugate gradient (CG) solver. After the error converges, we begin to deform the mesh using the warping field $\mathcal{W}$, which is also optimized by the GN solver in real time.

Fine Scale. Fine-scale reconstruction is performed after the large- and middle-scale reconstructions; thus, a normal map can be solved to further update the geometry. The fine-scale reconstruction is formulated as follows:

$$\arg\min_{\boldsymbol{n},\boldsymbol{\gamma},\mathbf{l}} E_{\text{shad}} + \lambda_{\text{alb}} E_{\text{alb}}. \tag{9}$$

The first term is given as follows:

$$E_{\text{shad}} = \sum_{i=1}^{N} \|S(\mathbf{v}_i) - I(\Pi(R\mathbf{v}_i + \mathbf{t}))\|^2, \tag{10}$$

where $I(\Pi(R\mathbf{v}_i + \mathbf{t}))$ is the corresponding pixel of $\mathbf{v}_i$ in the color image.

The second term is a regularization term and can be given as follows:

$$E_{\text{alb}} = \sum_{i=1}^{M_{\text{alb}}} \left(\frac{\gamma_i}{\sigma_i^{\text{alb}}}\right)^2, \tag{11}$$

where σ_i^{alb} is the eigenvalue of the ith PCA basis for albedo.

In practice, albedo, normal, and environment lighting are iteratively optimized. After the normal is optimized, we use the existing method [38] to compute the positions of the vertices in the face template.

5.2 Solving for Neutral Shape

Here, we discuss how to solve the neutral face shape of a user (denoted as b_0). Because a single depth map contains strong noise and cannot appropriately cover the face profile, we employ a fusion technique to integrate multiview images to obtain better quality. Here, we ask the user to maintain a neutral expression and turn the face left and right in front of a camera to record a sequence.

For each frame in the sequence, we perform the aforementioned optimization in large and middle scales. $\boldsymbol{\beta}$ can be prefixed because the user is having a neutral expression. After optimization, each vertex will associate with the depth point (i.e., the closest point) in the frame. Further, we fuse the depth points of the online recorded frames to obtain increasingly complete and better geometry with less noise by back-warping the depth points to the first frame (referred to as the canonical frame) based on the solved large- and middle-scale motions. The large- and middle-scale optimizations are performed again to fit the fused depth points to obtain improved reconstruction. Finally, the color image of the canonical frame can be used to realize fine-scale optimization to add details to the reconstruction.

The most straightforward solution is to model the motion between frames as rigid motion and use the rigid motion to fuse the depth points. However, the face shape may slightly change during rotation; thus, we run large- and middle-scale optimizations first to model the change and fuse the depth points with improved correspondences. We then run the optimization again to obtain better reconstruction.

5.3 Solving for Each Expression

Here, we reconstruct the facial expressions (denoted b_ws) in the recorded sequence to realize blendshape updates. We run the optimization described in Section 5.1 to reconstruct each frame in real time. $\boldsymbol{\alpha}$ is fixed with the values obtained in Section 5.2.

As mentioned previously, not all the b_ws will be used in the blendshape update described in Section 6.3). Here, some keyframes are selected for blendshape update. We use the reconstructed blendshape coefficient β^t to perform this update. A greater β value indicates stronger expression, which is beneficial for blendshape update. Thus, if $|\beta^t|$ is greater than the threshold δ, frame t will be selected as a keyframe.

6 Emotion-Constrained Blendshape Update

In this section, we discuss how to use the reconstructed neutral face b_0 and facial expressions b_ws to update the blendshapes with wrinkle-scale facial details while preserving emotional semantics. In Sections 6.1 and 6.2, we propose an emotion mapping that can be used to preserve the emotions of blendshape bases in blendshape update. [1] updates the blendshapes in Laplacian space. However, this method cannot be used to achieve a blendshape model with wrinkles. In Section 6.3, we present a new blendshape update strategy combined with emotion mapping that can fuse fine-scale details into a blendshape model.

6.1 Emotion Feature Extraction

Here, we discuss how to extract the emotion feature from a face image. Following the literature [39], wherein emotion features are extracted for face fitting, we first train an emotion classification network [40] on the *AffectNet* dataset [41], which contains 420,299 images with manually annotated emotion labels. Then, we use the feature of the second-last layer of the network as the emotion feature. With this network, we obtain a 128-dimensional emotion feature for each frame t as $\boldsymbol{f}_e^t$.

6.2 Emotion Mapping

With ideal blendshapes, a set of blendshape coefficients defines a unique expression and corresponds to a unique emotion feature. Here, we discuss how to train the mapping from the blendshape coefficients to the emotion feature. We collected 522,096 facial images with "ground truth" blendshape coefficients and emotion features for 40 characters with different genders and ethnicities. These images were recorded with the emotions in *AffectNet* and other facial motions. The "ground truth" blendshapes and blendshape coefficients were obtained using the commercial FaceShift software. We also extracted the emotion feature using the network described in Section 6.1 as the ground truth. We trained emotion mapping with three fully connected layers using the $L2$ loss as the loss function. The latent feature size of each layer was 1,024. The obtained mapping is denoted as $\mathcal{M}$ and maps a set of blendshape coefficients $\boldsymbol{\beta}$ to an emotion feature as $\boldsymbol{f}_e = \mathcal{M}(\boldsymbol{\beta})$.

6.3 Blendshape Update

After obtaining the high-fidelity neutral shape b_0, DT is first applied to obtain a set of initial blendshapes B_{exp} from the blendshapes of a template. Meanwhile, as the tracking results b_ws are also obtained, we first select some keyframes from b_ws according to the selection scheme discussed in

Section 5.3. Then, blendshape update can be performed by optimizing the vertex positions and blendshape coefficients separately to minimize the energy E_{BS}

$$\arg\min_{B^*_{\exp},\boldsymbol{\beta}^t} E_{BS} = \sum_{t=k_0}^{k_m} \left(E_d^t + \lambda_e E_e^t + \lambda_{\text{reg}} E_{\text{reg}}\right), \tag{12}$$

where t is the frame index of a single keyframe and the first term can be expressed as follows:

$$E_d^t = \|b_0 + B^*_{\exp}\boldsymbol{\beta}^t - b_w^t\|_2^2, \tag{13}$$

which represents the fitting error to the tracking result b_w^t.

The second term is given as follows:

$$E_e^t = \|\mathcal{M}(\boldsymbol{\beta}^t) - \boldsymbol{f}_e^t\|_2^2, \tag{14}$$

which attempts to constrain that the final blendshape coefficients should match the emotion information extracted from the recorded images. The term E_e^t is the key to this optimization. It uses the prior emotion information to solve the ambiguity associated with the minimization of the first term E_d^t. The third term can be given as follows:

$$E_{\text{reg}} = \|(B^*_{\exp} - B_{\exp}) \odot D\|_F^2, \tag{15}$$

which further preserves the semantic meaning of the blendshapes using predefined masks. Here, each blendshape only controls a local active region on a face; thus, this term can be applied to ensure that the active region does not change after the blendshape update. Specifically, we employ a matrix $D \in \mathbb{R}^{3N\times M_{\exp}}$ to encode the information of the $M_{\exp}$ masks. The operator $\odot$ represents elementwise multiplication. $D(3i : 3i+2, j) = 1$ indicates that the ith vertex in the jth blendshape is not in the active region, whereas $D(3i : 3i+2, j) = 0$ indicates that it is in the active region. The D values change smoothly from 0 to 1 in the boundary region.

E_{BS} can be minimized by optimizing $\boldsymbol{\beta}$ and $B^*_{\exp}$ separately in each iteration. Here, we first fix $B^*_{\exp}$ as $B_{\exp}$ to solve for $\boldsymbol{\beta}^t$. This optimization is very fast because $\boldsymbol{\beta}^t$ only contains 51 dimensions. Then, we fix $\boldsymbol{\beta}^t$ to solve for $B^*_{\exp}$. The new objective function will be computed and stored in fixed graphics memory when adding a new keyframe. The time cost associated with this is constant because we can add a new $E_d^{k_{m+1}} + \lambda_{\text{reg}} E_{\text{reg}}$ to the previous E_{BS}. Although there are $3N \times M_{\exp}$ variables in this optimization, each dimension of each vertex is independent of the others; thus, the optimization can be treated as $3N$ independent processes. The number of variables in each optimization is only $M_{\exp}$; therefore, we can solve them efficiently using a Gauss-Seidel solver. The functions can be constructed and solved in parallel on the GPU. This optimization process can be drastically accelerated using these strategies.

7 Results

In this section, we first introduce the implementation details of our system. Subsequently, we evaluate the proposed real-time blendshape update on both the semantic preservation and detail generation tasks by comparing it with the state-of-the-art solutions. In addition to the overall system, we evaluate the novel techniques with respect to some components of the proposed method, including neutral reconstruction and emotion control. To demonstrate the power of the proposed technique, we perform facial retargeting among real humans whose personalized blendshapes can be obtained using the proposed technique. Finally, we discuss the limitations of our technique.

7.1 Implementation Details

In the current implementation, the face template is obtained from the basel face model [42] with $N = 34508$ vertices and $M_{\text{id}} = 199$ PCA bases for identity, and the $M_{\exp} = 51$ blendshapes are obtained from FaceShift. Here, we selected the parameters empirically as follows: $\lambda_e = 0.02$, $\lambda_{\text{reg}} = m \times 0.01$, $\lambda_{\text{id}} = 0.1$, $\lambda_{L1} = 4$, $\lambda_s = 100$, $\lambda_{\text{alb}} = 0.2$, $r = 20$, and $\delta = 0.4$. We set λ_{lm} to 15 in large-scale tracking and changed it to 4 when optimizing the warping field. The system was primarily implemented on a single NVIDIA GeForce RTX 2080 Ti GPU using the OpenGL and NVIDIA CUDA APIs. Some parts of the algorithm were run on a 3.4 GHz eight-core Xeon E3-1231 with 16 GB of memory. For each input frame, feature point detection required 12 ms on the CPU and emotion feature extraction required 1.7 ms. The large-scale deformation could be estimated in 9 ms, and the warping field could be optimized in 15.2 ms. In addition, the reconstruction of the fine-scale details required 21.4 ms on the GPU. The blendshape update ran in parallel with the tracking. Here, approximately 1.26 s was required for the optimization of Eq. (12). Once the optimization was complete and 40 new keyframes were detected, the optimization was repeated to further update the blendshapes. The tasks on the CPU were multithreaded, and the tasks involved the operation of GPU on different streams. This system operated at 30 Hz for neutral face reconstruction (typically requiring 5-10 s) and 20 Hz for real-time tracking. The input RGB-D sequences were captured using Intel RealSense SR300, and the resolutions of the color and depth images were 640×480.

7.2 Blendshape Update

First, we qualitatively compared the blendshapes obtained from different solutions by fitting the facial sequences and retargeting the tracked expressions to the template face. Here, we tested the blendshapes of different users with various expressions. Some selected results are presented in Fig. 4. From (a) to (d), we observe that the proposed technique generates more correct mouth and cheek motions to represent the happy emotion from subtle to strong. However, the compared solutions could not correctly capture this emotion. In (e) and (f), we generated more correct shapes of the nasolabial fold and the upper lip pose to better represent the expressions. Our two solutions generated more facial details, e.g., wrinkles on head. Readers are referred to the teaser (the left part; this example shows that the proposed technique reconstructed the subtle pout in a better manner) and accompanying video for additional results.

We then quantitatively evaluated blendshape update on semantic preservation. The major advantages associated with the emotion constraint are that the semantics of blendshapes are maintained and that the expressions can be correctly reconstructed. Therefore, we used the blendshapes obtained via different solutions to fit expression sequences and

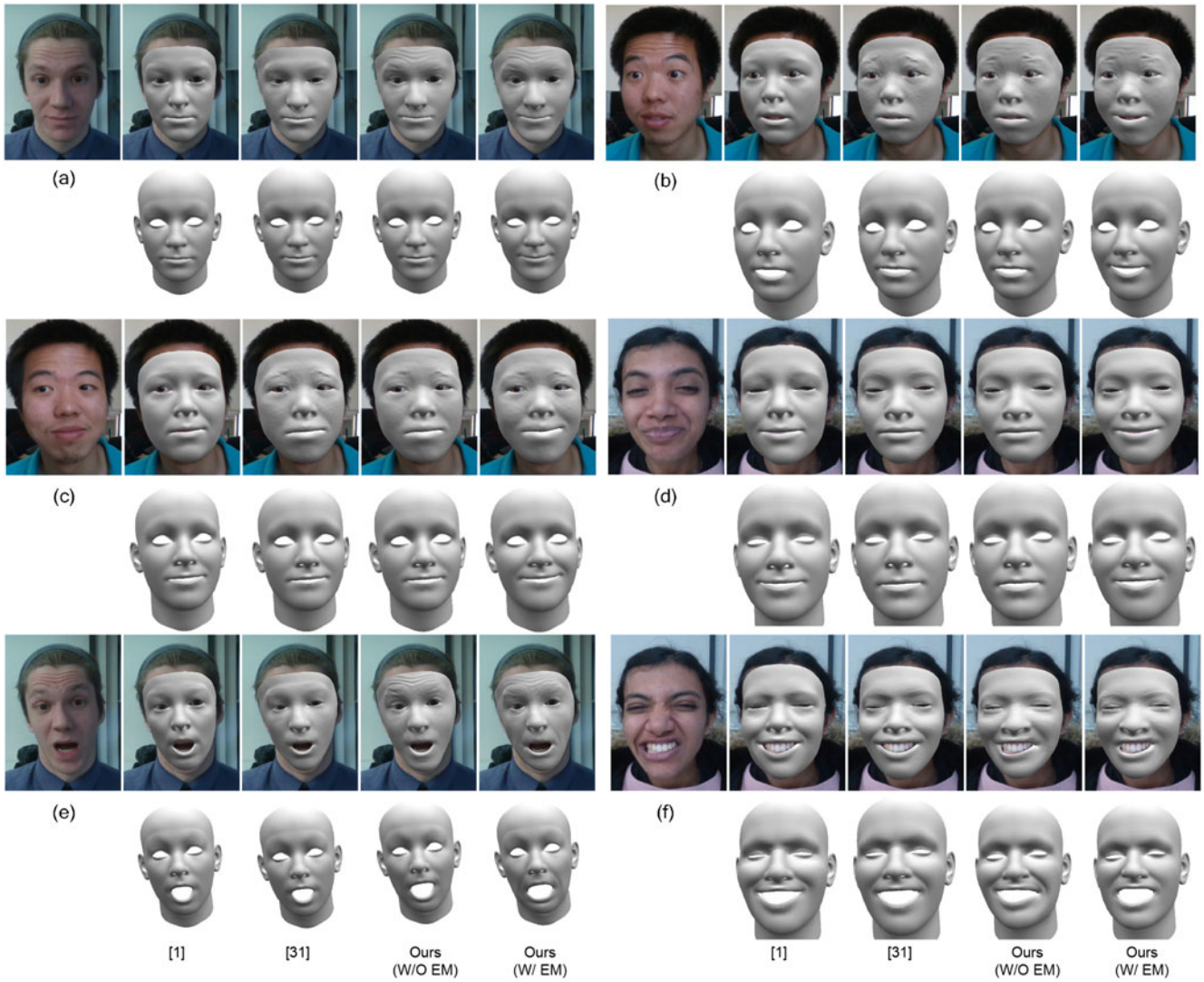

Fig. 4. Tracking and retargeting results on the selected frames (left to right). Each result contains the input image, the tracking and retargeting results with the blendshapes obtained by [1] and via deformation transfer, proposed technique without emotion constraint, and proposed technique with emotion constraint.

determine whether the reconstructed coefficients can closely map to the "ground truth" emotion feature. The curves of a sequence with 500 frames are shown in Fig. 5. As can be seen, our blendshapes can fit the sequence with the most correct emotion features when compared with those observed when using the state-of-the-art and other solutions. Table 1 shows the average results on eight sequences of eight different users. Updating the blendshapes without the emotion constraint pollutes the semantics, indicating the importance of constraining the blendshape update with correct emotions. We did not employ the emotion constraint in this fitting to quantitatively evaluate the blendshapes. We compare the updated blendshapes of different methods to the "ground truth" generated by FaceShift in Fig. 7 to better evaluate the effect of this emotion constraint. As shown, the updated blendshapes generated using the proposed technique exhibited minimum errors. By comparing the error maps in Figs. 7c and 7d, blendshape update without emotion constraint resulted in large errors, indicating the importance of our emotion constraint.

Further, we evaluated the effect of the three-scale deformation in our blendshape update algorithm. We used the blendshapes of different solutions to fit the expression sequences by optimizing the blendshape coefficients to demonstrate the effect of the middle-scale deformation in

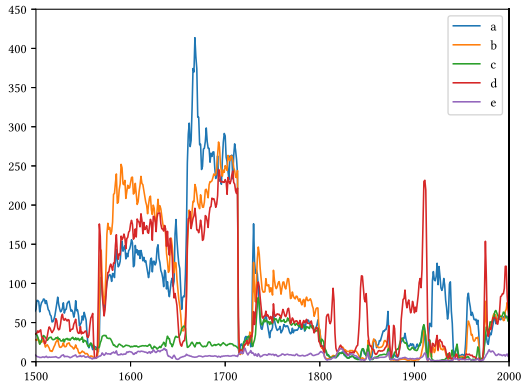

Fig. 5. $L2$ distance to the "ground truth" emotion feature for tracking results with different blendshapes. Blendshapes obtained by (a) [1], (b) deformation transfer [31], (c) proposed technique without middle-scale reconstruction, (d) proposed technique without emotion constraint, and (e) our full pipeline.

TABLE 1
Average $L2$ Distance to the "Ground Truth" Emotion Feature on Eight Motion Sequences of Eight Users

[1]	DT	W/O Emotion	W/ Emotion
63.64	39.87	59.06	11.77

blendshape updating. Here, we computed the errors between input depth maps and the reconstructed facial models. Fig. 6 shows the error curves of a sequence with 500 frames. Blendshape updating with middle-scale reconstruction considerably reduced the depth fitting errors. By comparing Figs. 6d and 6e, we can state that the addition of an emotion constraint did not increase the fitting errors. We directly compared the blendshape bases of different methods to demonstrate that our blendshape update can effectively construct fine-scale details. As shown in Fig. 8, our blendshape bases successfully captured the user-specific fine-scale features shown in the reference image and outperformed the remaining methods. In the first row, our updated frown basis captured the wrinkles on the forehead. In the second row, the motion of the upper lip stretches the muscles below the left eye and wrinkles are generated in this region when using the proposed technique. The input sequence for the blendshape update only involved ordinary facial expressions. We did not ask the user to perform any expression similar to the blendshape bases. To demonstrate more fine-scale features, we present additional retracking results obtained using the updated blendshapes together with the three-scale reconstruction results in the supplemental materials, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TVCG.2020.3033838.

7.3 Component Evaluations

7.3.1 Neutral Reconstruction

We evaluated our neutral reconstruction process, and the results are shown in Fig. 9. As discussed in Section 5.2, in each frame, we deform the template to fit the input depth and build the correspondences between the template and the depth map. The newly recorded depth can be correctly fused with the developed correspondences. Further, with the increasing number of frames, the fused geometry becomes increasingly complete and the noise in the depth is filtered out. The fused geometry is shown in the first row of Fig. 9, the error map of which (with a prescanned face) is shown in the second row. Then, warping is conducted in the second time to deform the template for fitting the fused geometry. As the quality of the fused geometry increases, the shape of the template becomes increasingly similar to that of the input user. The warped face is shown in the third row of Fig. 9, and the errors are shown in the fourth row.

A good shape of the user can be generated via our neutral reconstruction process. We compare our result with those of [1] and [2], which are state-of-the-art online model updating techniques, and the KinectFusion method, which is a state-of-the-art of depth fusion method for static objects (Fig. 10). [1] could not achieve a very similar result to the "ground truth" because the shape was updated in a limited

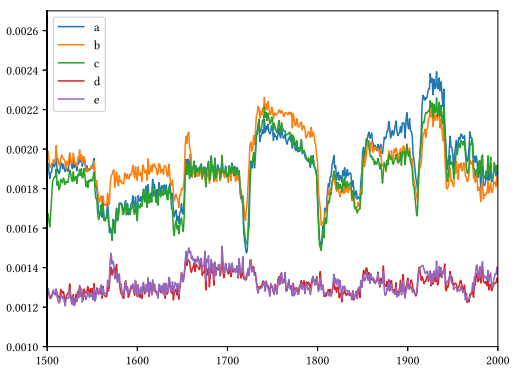

Fig. 6. Depth fitting errors (m) to the input depth maps for tracking results with different blendshapes. Blendshapes obtained by (a) [1], (b) deformation transfer [31], (c) proposed technique without middle-scale reconstruction, (d) proposed technique without emotion constraint, and (e) our full pipeline.

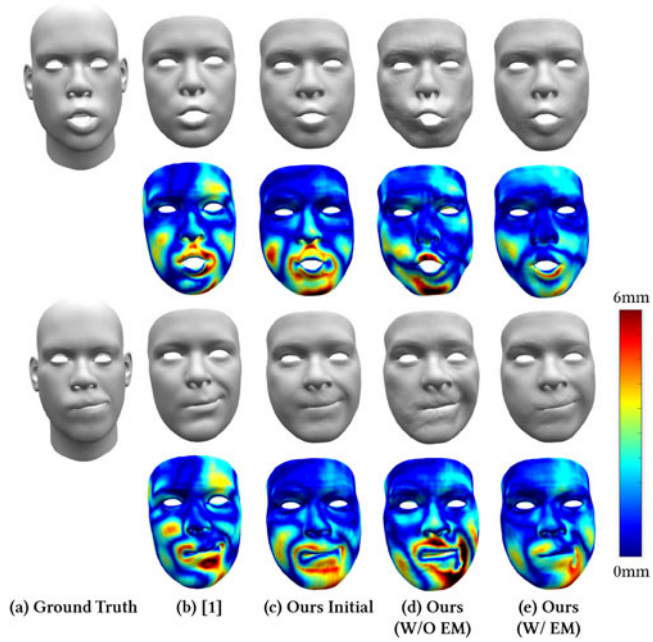

Fig. 7. Quantitative comparisons of blendshape update on real data: (a) "ground truth" blendshapes, (b) blendshapes obtained by [1], (c) proposed initial blendshapes obtained by deformation transfer, (d) blendshapes obtained using the proposed technique without emotion constraint, and (e) blendshapes obtained using the proposed technique with emotion constraint.

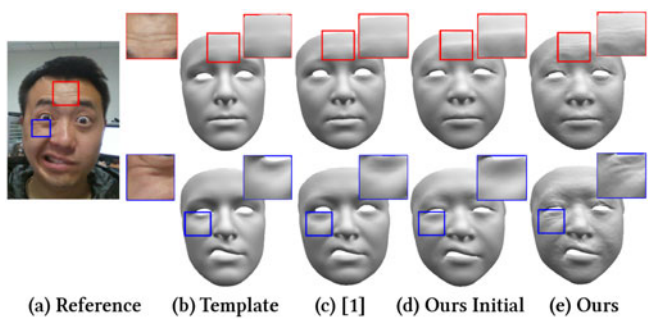

Fig. 8. Comparison of blendshape update on real data: (a) reference image of the user, (b) template blendshapes; (c) blendshapes obtained by [1], (d) our initial blendshapes obtained by deformation transfer, and (e) our updated blendshapes. The shown blendshape bases are activated in the reference image.

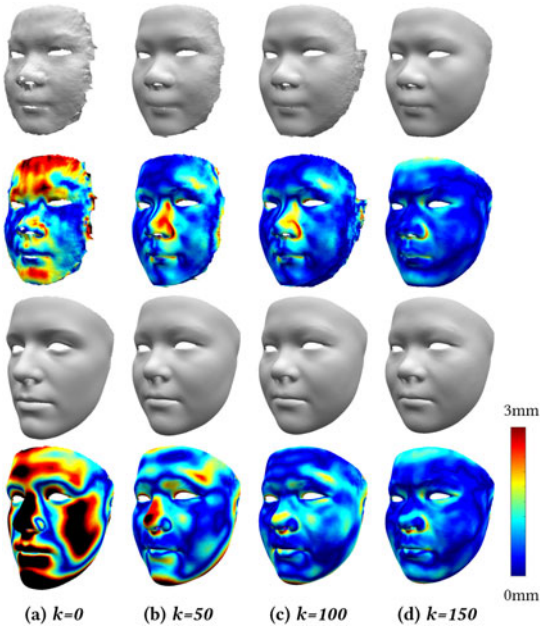

Fig. 9. Evaluation of our neutral face reconstruction process. The fused face b_f (first row) and deformed template (third row) improve with additional frames. The numeric errors compared to the prescanned "ground truth" are shown in the second and fourth rows, respectively. The input depth was recorded by RealSense SR300. The error for each vertex is color-coded, and k is the label of the current input frame.

deformation space. Notice that [1] did not require a static reconstruction step. The KinectFusion result looks good; however, it does not consider face priors. Thus, the sharp curvature change between the two lips is not reconstructed. [2] only allows rigid motion in the recording; therefore, nonrigid motions are not handled. The errors associated with this method are larger than those associated with the proposed technique. We do not use color images to generate details based on our result for a fair comparison with [2], which does not use color. However, our system can use color to generate realistic details, as shown in Figs. 4 and 13.

7.3.2 Emotion Control

For emotion control, we train the mapping from the blendshape coefficients to the emotion features. Further, we include this mapping in an emotion constraint in the facial expression tracking pipeline to estimate semantically correct blendshape

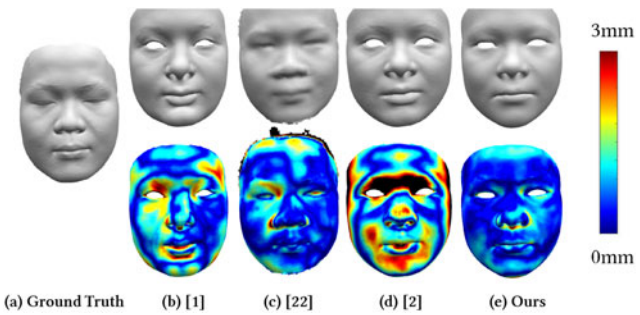

Fig. 10. Comparison of neutral reconstruction: (a) a scanned face, (b) the result of [1], (c) the result of KinectFusion [22], (d) the result of [2], and (e) our result. The inputs to the three methods were recorded by RealSense SR300, and the errors were calculated by treating the scanned face as the ground truth.

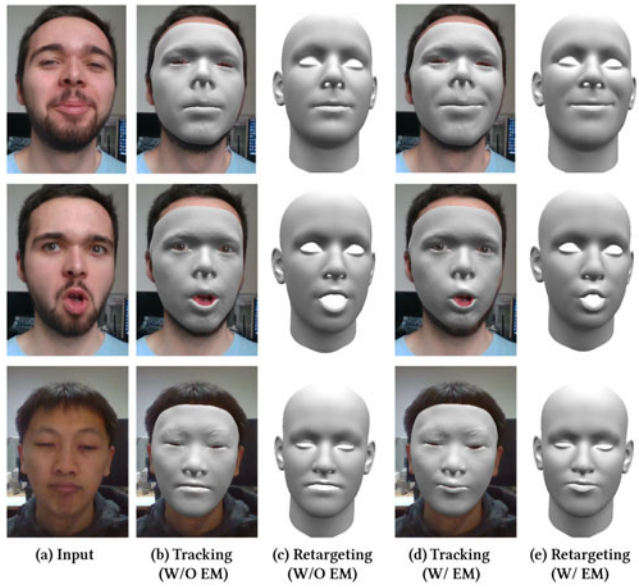

Fig. 11. Comparisons of face fitting with and without the emotion constraint: (a) input image, (b, c) fitting and retargeting results without the emotion constraint, and (d, e) fitting and retargeting results with the emotion constraint.

coefficients when the blendshapes are not appropriately established, which is important for realizing the proposed semantic-preserving blendshape update. We will evaluate both the mapping and the emotion constraint in this subsection. We evaluated the mapping according to the $L2$ distance between the mapping-obtained emotion feature and the "ground truth" emotion feature. Table 2 shows the $L2$ error on the training and testing datasets. Even though the average distance associated with the test is higher than that associated with training, it is still considerably less than the between-class distance of emotions and the within-class distance, indicating that the mapping is sufficient to preserve emotions.

Subsequently, we used mapping to construct an emotion constraint based on facial expression tracking. We used blendshapes B_{exp} to perform face fitting to our reconstructed faces b_w for evaluating whether the emotion

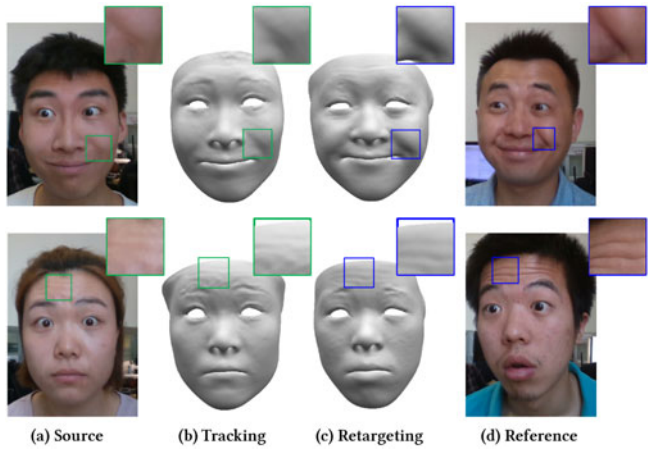

Fig. 12. Retargeting results of our personalized blendshapes (left to right): the source image, the tracked source expression, the retargeted target expression, and the reference image of the target with a similar expression (to evaluate the user-specific features in the retargeting result (not used in the experiment)).

Fig. 13. More retargeting results of our personalized blendshapes. The first column shows the source expressions. The following columns show the retargeting results, where the top images show the neutral expression of the target characters.

constraint maintains the semantics of the estimated blendshape coefficients. The effect of the emotion constraint is presented in Fig. 11. As can be seen, without this emotion constraint, the fitting provides incorrect blendshape coefficients, resulting in either wrong emotions (first row) or shapes (second and third rows). Better coefficients can be obtained using the emotion constraint, which can be observed in either the fitting or retargeting results. This can also be observed in the accompanying video.

7.4 Real Human Retargeting

We have generated user-specific blendshapes. Thus, we can achieve expression retargeting across real humans. Fig. 12

TABLE 2
$L2$ Distance to "Ground Truth" for Training and Testing Compared to the Average Within-Class Distance and Between-Class Distance of the Eight Types of Emotions

Within-class	Between-class	Train	Test
121.32	408.14	9.16	18.73

shows that we can successfully transfer expression semantics when the user-specific features are maintained by our blendshapes. In the upper row, as shown in the reference image, the target character can generate a strong nasolabial fold that is maintained after expression transfer. In contrast, the source character does not have this feature. Similarly, in the lower row, specific forehead wrinkles are generated correctly after the transfer. Additional results are shown in Fig. 13. As shown, different unique features are generated for different characters.

7.5 Discussions

We only performed comparison with one previous study [1] because we attempt to directly update blendshapes online with real-time tracking of the motion sequence. [2] did not update blendshapes but added a corrective field, [8] required the predefined expressions to be recorded, and [5] operated offline. These techniques have some advantages over the proposed technique. For example, [5] and [8] did not require depth data, and [8] effectively modeled face parts such as eyes. However, the proposed technique uses the emotion information extracted from images to preserve the semantics associated with blendshape update and generates high-fidelity blendshapes in real time.

Our emotion preservation technique did not present a significant contribution during our experiments. Humans exhibit similar large-scale motions for the same expressions. Under observation, human-specific features are inherently subtle. However, these subtle differences are sometimes important to identify expressions and identities. Such subtle factors can be considered to overcome uncanny valley and achieve high-fidelity and low-cost human animation.

The emotion constraint helps to constrain the blendshape coefficients; however, this does not always result in correct coefficients because of the error associated with the mapping and ambiguity of the features. The coefficients may be estimated incorrectly for some expressions with very subtle emotions (Fig. 14). However, our blendshape update process considers many frames together, and we select keyframes with extreme expressions; thus, frames with incorrect coefficients do not considerably affect our blendshape update.

Fig. 14. Blendshapes are updated with incorrect coefficients; thus, combined with the incorrect coefficients, the result appears OK (middle). However, combined with the "ground truth" coefficients, the result is incorrect with respect to the eyes and mouth (right).

Because we use SfS to generate details, high-frequency lighting may generate artifacts, as shown in the live demo in our accompanying video. Because the generation of these artifacts has some randomness and the same artifacts cannot always be generated in the same region with the same expression, our keyframe-based update is robust against the artifacts to a certain extent.

8 Conclusion

In this study, we have proposed a high-fidelity face blendshape update technique conducted online with real-time face tracking. In the proposed technique, the emotion information is employed to preserve the semantics of the blendshapes without the need to record predefined expressions. The system leverages the depth and color input and uses an efficient three-scale optimization process to update the blendshapes with user-specific facial features, including facial details, in real time. The experimental results indicate that the obtained blendshapes can better express facial emotions for face tracking and generate vivid facial animations for expression retargeting.

Acknowledgment

This work was supported by the National Key R&D Program of China under Grant 2018YFA0704000, the NSFC under Grant Nos. 61822111, 61727808, 61671268 and Beijing Natural Science Foundation under Grants JQ19015, L182052.

References

[1] S. Bouaziz, Y. Wang, and M. Pauly, "Online modeling for realtime facial animation," *ACM Trans. Graph.*, vol. 32, no. 4, pp. 1–10, 2013.
[2] H. Li, J. Yu, Y. Ye, and C. Bregler, "Realtime facial animation with on-the-fly correctives," *ACM Trans. Graph.*, vol. 32, no. 4, Jul. 2013, Art. no. 42.
[3] C. Cao, Q. Hou, and K. Zhou, "Displaced dynamic expression regression for real-time facial tracking and animation," *ACM Trans. Graph.*, vol. 33, no. 4, 2014, Art. no. 43.
[4] J. Thies, M. Zollhöfer, M. Nießner, L. Valgaerts, M. Stamminger, and C. Theobalt, "Real-time expression transfer for facial reenactment," *ACM Trans. Graph.*, vol. 34, no. 6, 2015, Art. no. 183.
[5] P. Garrido *et al.*, "Reconstruction of personalized 3D face rigs from monocular video," *ACM Trans. Graph.*, vol. 35, no. 3, 2016, Art. no. 28.
[6] H. Li, T. Weise, and M. Pauly, "Example-based facial rigging," *ACM Trans. Graph.*, vol. 29, no. 4, 2010, Art. no. 32.
[7] T. Weise, S. Bouaziz, H. Li, and M. Pauly, "Realtime performance-based facial animation," *ACM Trans. Graph.*, vol. 30, no. 4, 2011, Art. no. 77.
[8] A. E. Ichim, S. Bouaziz, and M. Pauly, "Dynamic 3D avatar creation from hand-held video input," *ACM Trans. Graph.*, vol. 34, no. 4, pp. 1–14, 2015.
[9] M. Zollh öfer *et al.*, "State of the art on monocular 3D face reconstruction, tracking, and applications," *Comput. Graph. Forum*, vol. 37, no. 2, pp. 523–550, 2018.
[10] V. Blanz *et al.*, "A morphable model for the synthesis of 3D faces," in *Proc. 26th Annu. Conf. Comput. Graph. Interactive Techn.*, 1999, pp. 187–194.
[11] C. Cao, Y. Weng, S. Zhou, Y. Tong, and K. Zhou, "FaceWarehouse: A 3D facial expression database for visual computing," *IEEE Trans. Vis. Comput. Graphics*, vol. 20, no. 3, pp. 413–425, Mar. 2014.
[12] T. Li, T. Bolkart, M. J. Black, H. Li, and J. Romero, "Learning a model of facial shape and expression from 4D scans," *ACM Trans. Graph.*, vol. 36, no. 6, 2017, Art. no. 194.
[13] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black, "SMPL: A skinned multi-person linear model," *ACM Trans. Graph.*, vol. 34, no. 6, 2015, Art. no. 248.

[14] J. R. Tena, F. De la Torre, and I. Matthews, "Interactive region-based linear 3D face models," in *Proc. ACM SIGGRAPH Papers*, 2011, pp. 1–10.
[15] C. Wu, D. Bradley, M. Gross, and T. Beeler, "An anatomically-constrained local deformation model for monocular face capture," *ACM Trans. Graph.*, vol. 35, no. 4, pp. 1–12, 2016.
[16] A. Tewari *et al.*, "FML: Face model learning from videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 10804–10814.
[17] A. Ranjan, T. Bolkart, S. Sanyal, and M. J. Black, "Generating 3D faces using convolutional mesh autoencoders," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 704–720.
[18] Z.-H. Jiang, Q. Wu, K. Chen, and J. Zhang, "Disentangled representation learning for 3D face shape," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 11957–11966.
[19] W.-C. Ma, T. Hawkins, C.-F. Chabert, M. Bolas, P. Peers, and P. Debevec, "A system for high-resolution face scanning based on polarized spherical illumination," in *Proc. ACM SIGGRAPH Sketches*, 2007, Art. no. 61–es. [Online]. Available: https://doi.org/10.1145/1278780.1278854
[20] T. Beeler, B. Bickel, P. Beardsley, B. Sumner, and M. Gross, "High-quality single-shot capture of facial geometry," *ACM Trans. Graph.*, vol. 29, no. 4, 2010, Art. no. 40.
[21] Y. Furukawa and J. Ponce, "Dense 3D motion capture for human faces," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 1674–1681.
[22] R. A. Newcombe *et al.*, "KinectFusion: Real-time dense surface mapping and tracking," in *Proc. 10th IEEE Int. Symp. Mixed Augmented Reality*, 2011, pp. 127–136.
[23] D. Bradley, W. Heidrich, T. Popa, and A. Sheffer, "High resolution passive facial performance capture," in *Proc. ACM SIGGRAPH Papers*, 2010, pp. 1–10.
[24] T. Beeler *et al.*, "High-quality passive facial performance capture using anchor frames," in *Proc. ACM SIGGRAPH Papers*, 2011, pp. 75:1–75:10.
[25] P. Garrido, L. Valgaerts, C. Wu, and C. Theobalt, "Reconstructing detailed dynamic face geometry from monocular video," *ACM Trans. Graph.*, vol. 32, no. 6, 2013, Art. no. 158.
[26] C. Cao, D. Bradley, K. Zhou, and T. Beeler, "Real-time high-fidelity facial performance capture," *ACM Trans. Graph.*, vol. 34, no. 4, 2015, Art. no. 46.
[27] L. Ma and Z. Deng, "Real-time hierarchical facial performance capture," in *Proc. ACM SIGGRAPH Symp. Interactive 3D Graph. Games*, 2019, pp. 1–10.
[28] Y. Li, L. Ma, H. Fan, and K. Mitchell, "Feature-preserving detailed 3D face reconstruction from a single image," in *Proc. 15th ACM SIGGRAPH Eur. Conf. Vis. Media Prod.*, 2018, pp. 1–9.
[29] A. Tewari *et al.*, "High-fidelity monocular face reconstruction based on an unsupervised model-based face autoencoder," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 357–370, Feb. 2020.
[30] A. Tewari *et al.*, "Self-supervised multi-level face model learning for monocular reconstruction at over 250 Hz," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 2549–2559.
[31] R. W. Sumner and J. Popović, "Deformation transfer for triangle meshes," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 399–405, 2004.
[32] Y. Xu, R. Hu, C. Gotsman, and L. Liu, "Blue noise sampling of surfaces," *Comput. Graph.*, vol. 36, no. 4, pp. 232–240, 2012.
[33] R. Ramamoorthi and P. Hanrahan, "A signal-processing framework for inverse rendering," in *Proc. 28th Annu. Conf. Comput. Graph. Interactive Techn.*, 2001, pp. 117–128.
[34] H. Li, B. Adams, L. J. Guibas, and M. Pauly, "Robust single-view geometry and motion reconstruction," *ACM Trans. Graph.*, vol. 28, no. 5, pp. 175:1–175:10, Dec. 2009. [Online]. Available: http://doi.acm.org/10.1145/1618452.1618521
[35] D. E. King, "Dlib-ml: A machine learning toolkit," *J. Mach. Learn. Res.*, vol. 10, pp. 1755–1758, 2009.
[36] J. Thies, M. Zollhofer, M. Stamminger, C. Theobalt, and M. Nießner, "Face2Face: Real-time face capture and reenactment of RGB videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2387–2395.
[37] K. Guo, F. Xu, T. Yu, X. Liu, Q. Dai, and Y. Liu, "Real-time geometry, albedo, and motion reconstruction using a single RGB-D camera," *ACM Trans. Graph.*, vol. 36, no. 3, 2017, Art. no. 32.
[38] C. Wu, M. Zollhofer, M. Niessner, M. Stamminger, S. Izadi, and C. Theobalt, "Real-time shading-based refinement for consumer depth cameras," *ACM Trans. Graph.*, vol. 33, no. 6, pp. 200:1–200:10, Nov. 2014. [Online]. Available: http://doi.acm.org/10.1145/2661229.2661232
[39] A. Chen, Z. Chen, G. Zhang, K. Mitchell, and J. Yu, "Photo-realistic facial details synthesis from single image," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 9428–9438.
[40] O. Arriaga, M. Valdenegro-Toro, and P. Plöger, "Real-time convolutional neural networks for emotion and gender classification," *Eur. Symp. Artif. Neural Netw.*, Bruges, Belgium, Apr. 24–26, 2019.
[41] A. Mollahosseini, B. Hasani, and M. H. Mahoor, "AffectNet: A database for facial expression, valence, and arousal computing in the wild," *IEEE Trans. Affective Comput.*, vol. 10, no. 1, pp. 18–31, First Quarter 2019.
[42] P. Paysan, R. Knothe, B. Amberg, S. Romdhani, and T. Vetter, "A 3D face model for pose and illumination invariant face recognition," in *Proc. 6th IEEE Int. Conf. Adv. Video Signal Based Surveillance*, 2009, pp. 296–301. [Online]. Available: https://doi.org/10.1109/AVSS.2009.58

Zhibo Wang received the BS degree in microelectronics science and technology from Nanjing University, Nanjing, China, in 2017. He is currently working toward the PhD degree with BNRist and School of Software, Tsinghua University, Beijing, China. His research interests include facial animation and face reconstruction.

Jingwang Ling is currently working toward the PhD degree in BNRist and School of Software, Tsinghua University, Beijing, China. His research interests include face reconstruction and animation.

Chengzeng Feng received the BS degree in software engineering from Northwestern Polytechnical University, Xi'an, China, in 2018. He is currently working toward the ME degree with the BNRist and School of Software, Tsinghua University, Beijing, China. His research interests include 3D face reconstruction and analysis.

Ming Lu received the PhD degree in information and communication engineering from Tsinghua University, Beijing, China, in 2019. He is currently a researcher with Intel Labs China. His research interests include computer vision and computer graphics. He is particularly interested in classification and detection, 3D face and body, and image restoration and synthesis.

Feng Xu received the BS degree in physics and the PhD degree in automation both from Tsinghua University, Beijing, China, in 2007 and 2012. He is currently an associate professor with the BNRist and School of Software, Tsinghua University. His research interests include facial animation, performance capture, and 3D reconstruction.