Controllable High-fidelity Facial Performance Transfer

Feng Xu* Jinxiang Chai† Yilong Liu⋆ Xin Tong*
*Microsoft Research †Texas A&M University ⋆Tsinghua University

Figure 1: *Our system transfers high-fidelity facial animation data from a source facial model to a target facial model while allowing the user to control and edit the retargeted animation sequence in the spacetime domain.*

Abstract

Recent technological advances in facial capture have made it possible to acquire high-fidelity 3D facial performance data with stunningly high spatial-temporal resolution. Current methods for facial expression transfer, however, are often limited to large-scale facial deformation. This paper introduces a novel facial expression transfer and editing technique for high-fidelity facial performance data. The key idea of our approach is to decompose high-fidelity facial performances into high-level facial feature lines, large-scale facial deformation and fine-scale motion details and transfer them appropriately to reconstruct the retargeted facial animation in an efficient optimization framework. The system also allows the user to quickly modify and control the retargeted facial sequences in the spatial-temporal domain. We demonstrate the power of our approach by transferring and editing high-fidelity facial animation data from high-resolution source models to a wide range of target models, including both human faces and non-human faces such as "monster" and "dog".

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—animation

Keywords: Facial animation, deformation transfer and editing, blendshape interpolation, motion retargeting

Links: DL PDF

1 Introduction

One of the most successful approaches for creating realistic facial animation is the use of prerecorded facial data. However, in most applications, the digital character that needs to be animated is often not a digital replica of the performer. This inevitably requires solving the problem of transferring the recorded facial performances to the desired facial model. The problem is challenging because the target facial model generally has a size and proportions different from the original actor.

For decades researchers in computer graphics and animation have explored a number of approaches for 3D facial performance transfer, including geometric mapping, blendshape mapping and data-driven methods [Pighin and Lewis 2005]. However, current methods for facial performance transfer are mainly focused on large-scale facial deformation. Due to recent technological advances in facial capture [Beeler et al. 2011; Huang et al. 2011], which have made it possible to acquire high-fidelity 3D facial performances with stunningly high spatial-temporal resolution, there is an increasing demand for transferring high-fidelity facial performances of a high-resolution source model to a desired target model.

Another challenge for facial performance cloning is the lack of control over the retargeted animation. In practice, no facial retargeting system is perfect as the facial proportions of the source model could be significantly different from the desired target model. Therefore, additional manual work by an animator is typically required to correct or adjust the retargeted animation. Even if the retargeting algorithm is perfect, the results of facial retargeting do not necessarily meet the requirements of the final animation. For example, the desired performance is not what the actor performed; the required expression is not present in the motion capture data, or it needs to be exaggerated [Havaldar 2006]. However, thus far, very few facial retargeting methods allow for adjustment and correction of the retargeted facial performances.

In this paper, we present a new solution for transferring and editing high-fidelity facial animation. Our approach takes a source reference expression, a source facial animation sequence and a target

reference expression as input and outputs a retargeted animation sequence that is "analogous" to the source animation. The system is appealing for facial performance transfer because it allows the user to control both large-scale deformation and fine-scale facial details of the retargeted animation in the spatial-temporal domain. To achieve this goal, we decompose high-fidelity facial performances into high-level facial feature deformation, large-scale mesh deformation and fine-scale facial details, and transfer them appropriately to reconstruct the retargeted facial animation in an efficient optimization framework. We demonstrate the power of our approach by transferring and editing high-fidelity facial animation data from high-resolution source models to a wide range of target models, including both human faces and non-human faces such as "monster" and "dog" (Figure 1).

Our work is made possible by a number of technical contributions:

- An efficient multi-scale representation that decomposes high-fidelity facial expression into high-level facial feature deformation, large-scale mesh deformation and fine-scale facial details. The new representation facilitates an efficient transfer of high-fidelity facial performances.
- A large-scale facial performance transfer and editing method that allows the user to adjust and correct the retargeted facial performance data in the spatial-temporal domain.
- A fine-scale facial detail transfer and editing method that allows the user to control the retargeted facial details (*e.g.*, wrinkles) in the spatial-temporal domain.

2 Background

Our system transfers and edits high-fidelity 3D facial performance data from a source model to a target facial model. We therefore focus our discussion on facial retargeting and editing. A full review of facial performance transfer systems is beyond the scope of this paper, and we refer to [Pighin and Lewis 2005] for a more detailed discussion.

2.1 Facial Expression Retargeting

One popular solution to facial expression retargeting is blendshape mapping [Buck et al. 2000; Chai et al. 2003; Chuang 2004; Weise et al. 2009; Seol et al. 2011; Kholgade et al. 2011]. Given a set of blendshapes for the source model and one for the target character that corresponds to the same expressions, then once the subject is mapped onto the source blendshapes, it can be mapped onto the target character by simply reusing the same set of blending weights. Recently, Seol et al. [2011] presented a blendshape-based facial animation retargeting system that is carefully designed to support the animator's workflow. Kholgade et al. [2011] developed a layered composition model of expressions for retargeting facial performances from the source model to a target character with dissimilar facial structure. More recently, blendshape mapping has been combined with facial tracking to animate a target character model [Bhat et al. 2013; Bouaziz et al. 2013; Li et al. 2013]. Blendshape mapping, however, often requires skilled artists to manually create the facial rig to ensure the quality of output animation, but this is a slow, labor-intensive and costly process.

One way to reduce manual effort for facial retargeting is to apply geometric mapping techniques to transfer facial deformation data from one source mesh model to another one [Noh and Neumann 2001; Sumner and Popović 2004; Baran et al. 2009; Li et al. 2010]. Notably, Sumner and Popovic [2004] proposed the deformation gradient representation for novel deformation transfer, which computes the set of transformations induced by the deformation of the source mesh and maps the transformations through the correspondences from the source to the target in an efficient optimization framework. Li et al. [2010] extended deformation gradient techniques to transfer a complete set of blendshape models from the source to the target by solving for an optimal reconstruction of the training examples from the target model. Current geometric mapping techniques, however, are mainly focused on transferring large-scale deformation rather than high-fidelity facial animation as targeted in our work. In addition, they do not allow adjustment and correction of the retargeted facial animation, thereby significantly limiting their applications in animation production.

Several researchers [Vlasic et al. 2005; Cao et al. 2013] have extended the blendshape representation to multilinear models for facial expression transfer. For example, Vlasic et al. [2005] estimated a multilinear model of human faces by examining geometric variations between 3D face scans and used the constructed multi-linear model to transfer expression parameters between different subjects. Cao et al. [2013] combined a multilinear representation with real-time facial tracking techniques to animate a target avatar model in real time. However, it is not clear how multilinear models can be extended to transfer high-fidelity facial performance data and how they can be applied to retarget facial expression data to non-humanoid characters (*e.g.*, dogs) because they are constructed from low-resolution human face scans.

Our work significantly extends previous work on facial retargeting by developing a multi-scale facial representation that facilitates high-fidelity facial expression transfer. Unlike previous work, we decompose high-fidelity facial performances into high-level facial features, large-scale deformation and fine-scale details and transfer them appropriately in an efficient optimization framework. Another distinction is that we allow the animator to control both large-scale deformation and fine-scale details of the transferred animation in the spatial-temporal domain. As shown in our results, this significantly improves the quality of output animation.

Our idea of decomposing high-fidelity facial performances into a multi-scale representation is relevant to recent efforts on multi-scale capture and editing of facial geometry and motion [Bickel et al. 2007; Bickel et al. 2008; Bermano et al. 2014]. In particular, Bickel et al. [2008] introduced a multi-scale representation for editing highly detailed facial expression data based on a combination of linear shell deformations and fine-scale pose space deformations. Our goal, however, is different because we focus on transferring the whole sequence of facial expression data from one source model to another one, rather than on facial capture and editing as targeted in their work. This motivates us to develop a new representation to achieve our goal. Additionally, unlike their large-scale deformation representation, which is based on linear shell mesh deformation, our representation combines the power of high-level facial features, blendshape representation and deformation gradients. Another difference is that we use a Laplacian coating of vertex displacements, rather than pose space deformation, to encode fine-scale facial details.

2.2 Facial Animation Editing

Our idea of combining facial retargeting with editing is inspired by recent success on facial expression editing, including both spatial facial editing [Joshi et al. 2003; Zhang et al. 2004; Lau et al. 2007; Meyer and Anderson 2007; Bickel et al. 2008; Lau et al. 2009; Lewis and Anjyo 2010; Tena et al. 2011] and spatial-temporal expression editing [Li and Deng 2008; Seol et al. 2012; Akhter et al. 2012]. Recent research on spatial facial editing has been focused on utilizing various forms of data-driven models to constrain the solution space of facial editing. For example, Zhang and col-

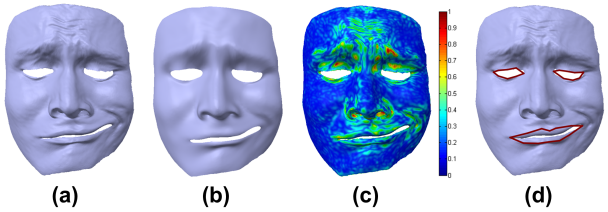

Figure 2: *Multiscale facial performance data representation: (a) the source mesh model; (b) the smoothed source mesh model; (c) fine-scale geometric details encoded by Laplacian coating (the colors visualize the magnitudes of the coating vectors along their normal directions.); (d) facial feature lines around the eyes and mouth.*

leagues [2004] performed facial expression editing by representing the solution as a weighted combination of database examples. Lau et al. [2007; 2009] constructed statistical models (mixture of factor analyzers) from prerecorded facial data and utilized them to edit facial expressions in a Maximum A Posteriori (MAP) framework. Tena et al. [2011] introduced a region-based PCA model that allows click-and-drag interaction for facial animation.

There has been little research done on editing facial animation in the temporal domain. Li and Deng [2008] explored how to use an orthogonal blendshape face model with constrained weight propagation to edit a facial motion capture sequence across the entire sequence. Seol and colleagues [2012] introduced a novel spacetime facial animation retargeting method for blendshape face models that reproduces the character of a source animation on a target face while providing room for additional creative control by the animator. Akhter and colleagues [2012] introduced bilinear spatiotemporal basis models for editing low-resolution facial performance data in the spacetime domain.

Our work is most closely related to the work of [Seol et al. 2012] because both allow for correction and adjustment of the retargeted motion in the spatial-temporal domain. Our work, however, differs from theirs in several ways. First and foremost, they concentrate on large-scale facial deformation while our goal is to retarget both large-scale facial deformation and fine-scale facial details such as wrinkles. With our framework, the animator is not restricted to adjusting large scale deformation. She can also edit fine-scale facial details such as by copying-and-pasting, enhancing or removing any wrinkle on the face. In addition, our spatial-temporal editing technique for large-scale deformation is also different from theirs. We optimize both blendshape weights and bases of the retargeted animation data to match the constraints specified by the user. In contrast, their spatial-temporal editing is strictly focused on modifying blendshape weights only. As shown in our results, editing both blendshape bases and weights provides the user better control over output animation and thereby improves the quality of output animation.

3 Problem Formulation

Given a reference expression of the source model S_0 and a reference expression of the target model T_0, our goal of controllable high-fidelity facial performance transfer is to transform a sequence of facial expression data of the source model S_1,..., S_n into an expression sequence of the target model T_1,...,T_n that is not only "analogous" to the source expression data but also consistent with constraints specified by the user. We assume that the reference source and target mesh models, as well as the source mesh sequence, have the same mesh topology.

One popular solution to this problem is to apply deformation transfer techniques (*e.g.*, [Noh and Neumann 2001; Sumner and Popović 2004]) to transfer each frame of the source expression sequence to the target model one by one. However, frame-by-frame deformation transfer often fails to preserve spatial-temporal correlation embedded in the source animation sequence. In addition, it does not allow for adjustment and correction of the retargeted animation in the spatial-temporal domain. Another limitation of current methods is that they are mainly focused on large-scale facial deformation rather than high-fidelity facial performance data as targeted in our work.

We propose a high-fidelity facial performance transfer method that addresses all the aforementioned challenges. The key idea of our approach is to decompose high-fidelity facial performance data into high-level facial feature deformation, large-scale geometric deformation and fine-scale facial details and transfer them appropriately to reconstruct the retargeted facial animation in an efficient optimization framework. Our system is advantageous for high-fidelity facial expression transfer because it not only considers spatial-temporal correlation in the source expression data but also allows the user to control and edit the retargeted facial expression data in the spatial-temporal domain.

Preprocessing. Our facial performance transfer process requires the reference source model S_0 to have the same mesh topology as the reference target model T_0. To achieve this goal, we first build the correspondences between the two reference meshes in a semi-automatic manner. Briefly, we manually identify a small number of corresponding points on the reference meshes and use them to deform the reference source mesh to the reference target mesh via the non-rigid Iterative Closest Points (ICP) technique described in [Huang et al. 2011]. The corresponding points of the remaining vertices of the source mesh are automatically obtained by searching for the closest points on the target mesh. To ensure that the reference target mesh has the same topology as the reference source mesh, we resample the target mesh using the topology of the source mesh in a manner similar to [Huang et al. 2011].

Multi-scale facial performance decomposition. We now describe our key idea on how to decompose facial performance data into a multi-scale representation that facilitates transferring and editing of high-fidelity facial performance data. Our first idea is to decompose each mesh model of the source expression data $S_t, t = 0, ..., n$ into two components: a smoothed, low-resolution mesh model $\bar{S}_t$ and fine-scale geometric details $\hat{S}_t$ (see Figure 2).

We obtain the smoothed source models via Laplacian filters [Taubin 1995]. For all the results shown in the paper, we iteratively filter the source meshes about 40 times to generate the smoothed source meshes. The fine-scale geometric details of the source models are encoded by a Laplacian coating of vertex displacements as described in [Sorkine et al. 2004]:

$$\hat{S}_t = \{\xi_t^v | \xi_t^v = \delta_v - \bar{\delta}_v, v \in S_t\}, \quad (1)$$

where v is a vertex of the source mesh S_t. And δ_v and $\bar{\delta}_v$ are the Laplacian coordinates of the vertex in S_t and $\bar{S}_t$ respectively. Note that we can reconstruct the original source mesh S_t from the smoothed source mesh $\bar{S}_t$ and fine-scale geometric details $\hat{S}_t$ as follows:

$$S_t = L^{-1}(L(\bar{S}_t) + \hat{S}_t), \quad (2)$$

where the operator L calculates the Laplacian coordinates of vertices on a mesh model and the inverse operator L^{-1} reconstructs the original mesh from Laplacian coordinates of the vertices by solving a Poisson equation [Sorkine et al. 2004].

In practice, geometric details $\hat{S}_t$ of the expression data $S_t, t = 1, ..., n$ are often a combination of large-scale deformation of geometric details of the reference model $\hat{S}_0$ and fine-scale temporal movement $\hat{D}_t$ of geometric details of the reference model $\hat{S}_0$[1]. We, therefore, can obtain fine-scale motion $\hat{D}_t$ by simply removing large-scale deformation of geometric details of the reference model $\hat{S}_0$ from geometric details of the expression data $\hat{S}_t$. In our formulation, geometric details are invariant to the underlying large-scale deformation because they are encoded by Laplacian coating of vertex displacements. Therefore, we can extract fine-scale temporal movement $\hat{D}_t$ based on the difference of the Laplacian coating between the reference model $\hat{S}_t$ and the deformed model $\hat{S}_0$:

$$\hat{D}_t = \hat{S}_t - \hat{S}_0. \tag{3}$$

Combining Equations (2) and (3), we have

$$S_t = L^{-1}(L(\bar{S}_t) + \hat{S}_0 + \hat{D}_t). \tag{4}$$

In addition, the facial features around the eyes and mouth are especially important for high quality facial animation [Bhat et al. 2013]. For example, commercial applications such as Live Driver [ImageMetrics] have obtained impressive results for facial puppetry by tracking only those facial features around those regions. This motivates us to extract high-level facial features around the eyes and mouth from the source expression data and incorporate them appropriately into our facial performance transfer framework. Specifically, we encode our high-level facial features using a set of connected edges or feature lines around the mouth and eye regions. Figure 2(d) shows the feature lines used in our implementation.

Multi-scale facial performance transfer. Similarly, we can decompose the reference target model T_0 into a smoothed target model $\bar{T}_0$ and fine-scale geometric details $\hat{T}_0$ and reconstruct the expression data of the target model as follows:

$$T_t = L^{-1}(L(\bar{T}_t) + \hat{T}_0 + \hat{D'}_t), \tag{5}$$

where $\bar{T}_t$ represents the smoothed mesh of the target expression at frame t, which can be obtained by large-scale deformation of the smoothed reference target model. And $\hat{D'}_t$ represents fine-scale facial movement of the reference target model.

Equation (5) shows that high-fidelity facial performance transfer requires solving for both large-scale deformation and fine-scale temporal movement of the reference target model. In the next two sections, we provide a detailed explanation of how to transfer large-scale deformation data and fine-scale facial movement of the source model to the target model. In addition, we discuss how to extend the transfer method to allow the user to quickly edit and control the retargeted facial animation in the spatial-temporal domain.

4 Large-scale Facial Performance Transfer

In this section, we describe an efficient optimization algorithm for transferring large-scale deformation sequences of the source model to the target model. In addition, we discuss how to incorporate high-level facial features (*e.g.*, facial feature lines) into the optimization process and how to extend the facial transfer process by generating retargeted facial performances that are consistent with user-defined constraints.

[1]Note that we omit the contributions of fine-scale temporal movement $\hat{D}_t$ on the low-resolution reference mesh $\bar{S}_0$ because in practice we cannot capture fine-scale motion details using a low-resolution mesh.

4.1 Blendshape Representation

We choose to formulate the large-scale deformation transfer process in the blendshape interpolation framework. Mathematically, we represent the smoothed source mesh $\bar{S}_t, t = 1, ..., n$ as a weighted combination of blendshape bases $\mathbf{b}_i, i = 1, ..., K$:

$$\bar{S}_t = [\mathbf{b}_1 ... \mathbf{b}_K]\mathbf{w}_t, \quad \mathbf{w}_t \geq \mathbf{0}, \tag{6}$$

where the vector $\mathbf{w}_t$ represents K nonnegative weights to model the smoothed or low-resolution face mesh at frame t. One major benefit of blendshape interpolations is that it decouples spatial details $\mathbf{b}_i, i = 1, ..., K$ from temporal details $\mathbf{w}_t, t = 1, ..., T$.

We follow the method in [Huang et al. 2011] to automatically select a minimum set of facial bases from the rigidly aligned source sequence by minimizing the reconstruction error associated with the selected facial bases:

$$\arg\min_{K,\{\mathbf{w}_t\},\{\mathbf{b}_i\}} \sum_{t=1}^{n} \|[\mathbf{b}_1 ... \mathbf{b}_K]\mathbf{w}_t - \bar{S}_t\|^2, \quad \mathbf{w}_t \geq \mathbf{0}, \tag{7}$$

where the distance $\|[\mathbf{b}_1 ... \mathbf{b}_K]\mathbf{w}_t - \bar{S}_t\|^2$ is evaluated by the sum of squared distances of corresponding vertices of the two meshes.

To simplify the optimization problem, we choose to constrain the space of $\mathbf{b}_i, i = 1, ..., K$ in a discrete space determined by the source facial sequence $\bar{S}_t, t = 1, ..., n$. We adopt a greedy strategy to find the optimal solution by initializing the set of face bases with $\mathbf{b}_1 = \bar{S}_0$ and then incrementally expanding the set by minimizing the objective function described in Equation (7). We evaluate the objective function by computing the unknown weights via an efficient non-negative least squares solver (NNLS) described in [James and Twigg 2005]. We terminate the process when the error defined in Equation (7) falls below a user-specified threshold ϵ.

We adopt the same representation for the smoothed expression sequence of the target model. Therefore, we can model the smoothed target mesh $\bar{T}_t, t = 1, ..., n$ as a weighted combination of blendshape bases $\mathbf{b}'_i, i = 1, ..., K$:

$$\bar{T}_t = [\mathbf{b}'_1 ... \mathbf{b}'_K]\mathbf{w}'_t, \quad \mathbf{w}_t \geq \mathbf{0}. \tag{8}$$

Large-scale deformation transfer, therefore, requires transferring both facial bases and blendshape weights of the source model to the target model.

4.2 Feature-preserving Large-scale Facial Deformation Transfer

We transfer each facial base of the source model $\mathbf{b}_i, i = 1, ..., K$ to the target model so that the target face bases $\mathbf{b}'_i, i = 1, ..., K$ share the same set of representative expression bases as the source face bases. We formulate the transfer process in an optimization framework by adapting the deformation transfer method in [Sumner and Popović 2004] to construct a target face base $\mathbf{b}'_i$ from the corresponding source face base $\mathbf{b}_i$. Because people are extremely sensitive to mouth and eye movement in facial animation perception, we further extend the optimization framework to incorporate high-level facial features into the transfer process (Figure 3).

We reconstruct facial bases of the target model $\mathbf{b}'_i$ by minimizing the following objective function ($E_{b'_i}$):

$$\left\| \bar{D'}_i^f(\mathbf{b}'_i, \bar{T}_0) - \bar{D}_i^f(\mathbf{b}_i, \bar{S}_0) \right\|^2 + \alpha \left\| \bar{D'}_i^l(\mathbf{b}'_i, \bar{T}_0) - \bar{D}_i^l(\mathbf{b}_i, \bar{S}_0) \right\|^2, \tag{9}$$

where the first term ensures that the 3D geometric deformation of the source model is consistent with the deformation of the target model. The second term evaluates how well the deformation of

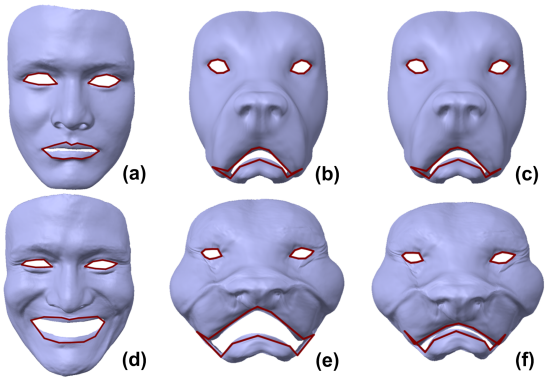

Figure 3: *Feature-preserving large-scale facial deformation transfer: (a) reference expression of the source model; (b)&(c) reference expression of the target model; (d) one base expression of the source model; (e) transferred expression with feature line constraints; (f) transferred expression without feature line constraints.*

feature lines on the source face base matches the deformation of the corresponding feature lines on the target face base. The weight α is experimentally set to 100.

We adopt the deformation gradients representation [Sumner and Popović 2004] to model geometric deformation of the face bases with respect to their reference model. Specifically, we have

$$\bar{D}_i^f(\mathbf{b}_i, \bar{S}_0) = \{D_i^f | D_i^f = (\mathbf{e}0_i^f, \mathbf{e}1_i^f, \mathbf{e}2_i^f)(\mathbf{e}0_0^f, \mathbf{e}1_0^f, \mathbf{e}2_0^f)^{-1}, f \in \bar{S}\}, \tag{10}$$

where D_i^f represents the affine transformation between the f-th triangle of the i-th face base $\mathbf{b}_i$ and its corresponding triangle in the reference mesh $\bar{S}_0$. The first two vectors $\mathbf{e}0^f$ and $\mathbf{e}1^f$ are the vectors aligned with two triangle edges and the third vector $\mathbf{e}2^f = \mathbf{e}0^f \times \mathbf{e}1^f$ is a vector perpendicular to the corresponding triangle plane.

The deformation of feature lines on the source face base is defined by deformation gradients $\bar{D}_i^l(\mathbf{b}_i, \bar{S}_0)$:

$$\{D_i^l | D_i^l = (\mathbf{e}0_i^l, \mathbf{e}1_{i0}^l, \mathbf{e}2_i^l)(\mathbf{e}0_0^l, \mathbf{e}1_{i0}^l, \mathbf{e}2_0^l)^{-1}, l \in \bar{L}\}, \tag{11}$$

where D_i^l represents the affine transformation between the l-th feature line on the i-th face base $\mathbf{b}_i$ and its corresponding feature line on the reference model $\bar{S}_0$. Here $\mathbf{e}0^l$ is the vector aligned with the l-th feature line. The second and third vectors $\mathbf{e}1_{i0}^l = \mathbf{e}0_i^l \times \mathbf{e}0_0^l$ and $\mathbf{e}2^l = \mathbf{e}0^l \times \mathbf{e}1_{i0}^l$ are two virtual vectors used to define D_i^l.

We solve the optimization via a linear system solver as in [Sumner and Popović 2004]. With the reconstructed target face bases $\mathbf{b}'_i, i = 1, ..., K$ and the weight vectors of the source sequence $\mathbf{w}_t$, we can reconstruct the smoothed target mesh in each frame by a weighted combination of target face bases: $\bar{T}_t = [\mathbf{b}'_1 ... \mathbf{b}'_K]\mathbf{w}'_t$.

Figure 3 shows a side-by-side comparison for target face bases generated by our method with and without facial feature line constraints. With facial feature lines, both facial deformation and high-level facial features around the mouth and eyes are faithfully transferred to the target model. In contrast, direct transfer of face bases without considering facial features lines fails to preserve facial features around the mouth and eyes, thereby producing an unsatisfactory result.

4.3 Transfer with User Control

Although the facial performance transfer method described in the previous section can automatically generate an expression sequence of the target face, the result does not necessarily meet the requirements of the final animation. In this section, we discuss how to extend the facial transfer method to allow the user to edit and control the retargeted facial animation in the spatial-temporal domain. In the current system, the user edits the large-scale deformation of the retargeted animation at any frame via kinematic editing operations such as point and distance constraints. The system automatically propagates the effects of editing to the entire sequence or any user-defined temporal window (the supplemental video shows a live demo of large-scale deformation transfer with various user editing operations.).

We develop an efficient optimization algorithm that computes both face bases and weight vectors of the target sequence from user-defined animation constraints:

$$\arg\min_{\{\mathbf{b}'_i\},\{\mathbf{w}'_t\}} \lambda_b E_b + \lambda_w E_w + \lambda_e E_e. \tag{12}$$

The first term $E_b = \sum_{i=1}^{K} E_{b'_i}$ measures the deformation transfer error of all target face bases, where $E_{b'_i}$ is defined in Equation (9).

The second term E_w ensures that the temporal behavior of the target expression sequence is consistent with that of the source expression sequence. In our implementation, we minimize the weight gradients of the source and target models in a user-specified temporal window $[t_s, t_e]$, resulting in the following objective function term:

$$E_w = \sum_{t=t_s}^{t_e} \left\| d(\mathbf{w}'_t) - d(\mathbf{w}_t) \right\|^2, \tag{13}$$

where the functions $d(\mathbf{w}_t)$ and $d(\mathbf{w}'_t)$ compute the velocities of the source and target weight curves at frame t, respectively.

The third term E_e measures how well the retargeted facial performance matches the user-specified constraints. The current system allows the user to interactively edit facial expressions at any frame by specifying one or a combination of two types of constraints: *point constraint* and *distance constraint*. The user can use *fixed constraints* to select points on the face mesh that should remain unchanged. This constraint must be used together with any of the other ones, thereby allowing for local control of the editing process.

Point constraint allows the user to change the positions of individual vertices on the mesh. This enables the user to have detailed control over the deformed facial mesh. For facial editing using point constraints, we have

$$E_e = \sum_{t \in P(t)} \sum_{v'_t \in P(v',t)} \left\| v'_t - v^*_t \right\|^2, \tag{14}$$

where $P(t)$ is the set of all frames that requires facial editing and $P(v', t)$ is the set of all mesh vertices that are edited by the user in frame t. v'_t is the final position of an edited vertex, which is defined by a weighted combination of corresponding vertices from all the face bases. v^*_t is the target position of v'_t specified by the user.

Distance constraint allows the user to select two facial vertices on the face mesh and edit the retargeted animation interactively by adjusting the distance between the two vertices. This is particularly useful for controlling high-level facial features such as the width and height of the eyes or mouth. For distance editing, we have

$$E_e = \sum_{t \in P(t)} \sum_{(v'_i, v'_j) \in P(v',t)} \left\| (v'_i - v'_j) - d^*(i,j) \right\|^2, \tag{15}$$

where v'_i and v'_j are a pair of vertices selected on the face mesh and $d^*(i, j)$ is the target offset vector of two vertices defined along

the direction of the original vertex pair but with a user-specified distance.

The weights λ_b, λ_w and λ_e control the importance of the three terms. Note that without user edits ($\lambda_e = 0$), this optimization is equivalent to the feature-preserving facial performance transfer algorithm described in Section 4.2. We have found that direct optimization of the cost function is not efficient, particularly when the temporal window is large. The system often becomes very slow and the optimization is also prone to fall into a local minimum.

To solve for the blendshape weights and face bases efficiently, we introduce an iterative optimization algorithm to decompose the large nonlinear optimization problem into two small optimization problems that can be solved efficiently. In each iteration, we keep one group of the unknowns constant and search for the optimal update for the other group of unknowns. Briefly, we initialize weights of the target model $\mathbf{w}'_t$ using the source weights $\mathbf{w}_t$. We then iteratively update the blendshape weights and face bases until the error difference of two successive steps drops below a user-specified threshold (5% of the current error in our implementation): (i). keep the weight vectors unchanged and update the face bases via a linear solver; (ii). fix the face bases and update the weights with a non-negative linear solver [James and Twigg 2005]. To support interactive user editing, we further speed up the face base optimization by performing the optimization on low-resolution face meshes first and then upsampling the result to the original-resolution meshes to refine the result.

By reconstructing face bases and weights simultaneously, our transfer method can efficiently propagate facial edits in both the facial pose and temporal domains. The user can choose different weights for λ_b and λ_w to control the edit propagation in both domains. On one hand, she can keep the face bases unchanged by setting λ_b to 0 and focus on updating the weights in a small temporal widow. On the other hand, she can fix the weights by setting λ_w to 0 and modify the face bases to propagate the edit across the entire sequence. As a result, any frame that has an expression similar to the editing frame will produce similar facial editing results no matter how far they are along the timeline. An appropriate tradeoff between the two will propagate the edit to both domains. Figure 4 shows the transfer results corresponding to different weights of λ_b and λ_w.

5 Fine-scale Facial Performance Transfer

This section focuses on transferring and editing fine-scale facial movement. We first describe an algorithm for automatically transferring the detailed motion of the source model to the target model. We then extend the system to allow the user to edit the retargeted fine-scale facial details in the spatial and temporal domain.

5.1 Automatic Transfer of Fine-scale Details

We transfer detailed movement of each source frame to the target by using the coating transfer method described in [Sorkine et al. 2004]. Briefly, for each vertex v' on the target mesh $\bar{T}_t$, we first compute a rotation R_v to align its local reference frame with the local frame of its corresponding vertex v on the source mesh $\bar{S}_t$. The local reference frame of each vertex is determined by its normal and a tangent vector, which is generated by projecting an edge originating from the vertex to its tangent plane. Next, we transfer the fine-scale motion $\hat{D}_t$ of the source model to the target model by rotating the Laplacian coating difference of each vertex on the source mesh with the computed rotation R_v:

$$\hat{D'}_t = \left\{ R_v(\xi_t^v - \xi_0^v) | v \in \bar{S}_t \right\}. \quad (16)$$

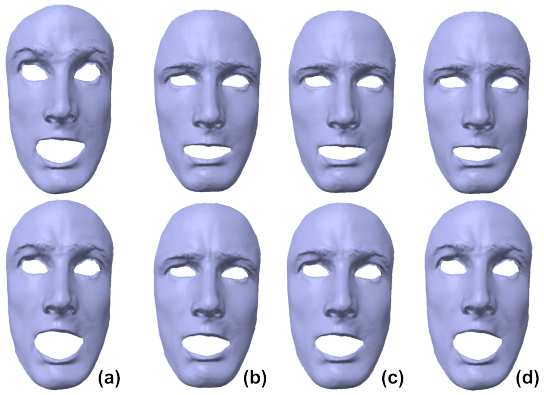

Figure 4: *Propagation of user edits in both the expression and temporal domains. In the first row, a frame to be edited by the user is shown in (a) and the edited results obtained by updating the bases only (b), jointly updating the bases and weights (c), and updating the weights only (d). Note that all editing results are consistent with constraints specified by the user. In the second row, (a) shows a frame with similar expression to the edited frame but is out of the temporal window; (b) shows that updating the bases only propagates the edit to similar expressions and (d) shows that updating only the weights does not change frames outside of the temporal window; result obtained by jointly optimizing the bases and weights is shown in (c).*

In the last step, we reconstruct the target mesh T_t from the fine-scale motions $\hat{D'}_t$, geometric details of the reference target model $\hat{T}_0$, and large-scale transfer result $\bar{T}_t$ obtained from Section 4 according to Equation (5).

5.2 Fine-detail Transfer with User Control

Our system allows the user to edit and control fine-scale facial performance transfer results in the spatial-temporal domain. The user starts with selecting a key frame for facial detail editing. After the user selects fine-scale facial details such as dynamic wrinkles at the key frame, she can perform a set of spatial editing operations in the selected region, including enhancing, diminishing and removing the selected facial details, copying and pasting the selected details to any facial region, and deforming the wrinkle shapes. The system automatically propagates the user edit at the key frame to the rest of the frames (the supplemental video shows a live demo of fine-detail transfer with user control.).

Dynamic facial detail selection. The user can select any fine-detail region to edit by painting operations. She can also sketch a 3D curve along a wrinkle and quickly select the region containing the wrinkle by identifying all vertices whose distance to the curve is smaller than a user-specified threshold (see Figure 5(b)). Given the selected region, the system automatically identifies the starting and ending frames of dynamic details by thresholding the Laplacian coating values of corresponding vertices. In particular, we extract a subsequence $[t_s, t_e]$ for which the Laplacian coating in the selected region $P(v', t)$ in each frame $t \in [t_s, t_e]$ satisfies $\frac{\sum_{v' \in P(v',t)} \xi_t^v}{\sum_{v' \in P(v',t_k)} \xi_{t_k}^v} \geq \mu$, where μ is a user-specified threshold (0.23 in our implementation).

Spatial fine detail editing. We now discuss how to edit selected fine details at key frame t_k. Given the selected region or wrinkle, the user can enhance, diminish or remove fine details by simply scaling the Laplacian coating of each vertex in the selected region.

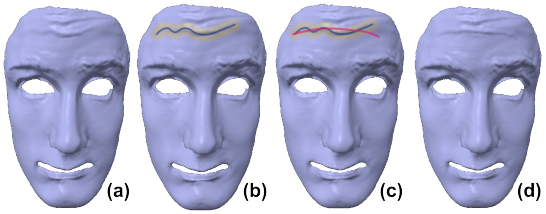

Figure 5: *Wrinkle region selection and shape editing. (a) Direct transfer result before editing. (b) The user paints the wrinkle to select the region for editing. The blue line is drawn by the user while the yellow region is the wrinkle region automatically extracted. (c) The user draws the target shape for the selected wrinkle. (d) The wrinkle after editing.*

After that, we apply the scaled details back to the base mesh to reconstruct the retargeted mesh according to Equation (5). In practice, there might be some discontinuities at the boundary of the selected region. However, these discontinuities are often very small and we can distribute them within a smoothing window around the discontinuity. We address this issue via linear interpolations between the edited and original coating values.

In addition, the user can copy and paste the selected facial details or wrinkles to another facial region on the face mesh at the key frame. For this purpose, the user aligns the source region to a target region by interactively adjusting scale, rotation and translation of the source region. The system automatically computes the closest points of all vertices of the target region in the source region and transfers the coating details from the source region to the target region via the coating transfer technique.

We have also developed a tool for interactive wrinkle shape editing. After the user draws a target curve on the face mesh (see Figure 5(c)), the system automatically deforms the wrinkle region to match the target curve drawn by the user. This requires solving the following least squares problem:

$$\arg\min_{v'} \sum_{v'_c \in P_C(v',t)} \left\| v'_c - v^*_c \right\|^2 + \lambda_l \sum_{v'_i \in P(v',t)} \left\| L(v'_i) - L(v^*_i) \right\|^2, \tag{17}$$

where v'_c and v^*_c are corresponding points on the selected wrinkle curve and the target curve drawn by the user. v^*_i is a vertex in the selected region containing the wrinkle and v'_i is its corresponding new position after deformation. $L(\cdot)$ is the Laplacian coordinates of a vertex.

Intuitively, the first term measures how well the deformed wrinkle matches the target curve drawn by the user. Evaluating the first term requires one-to-one correspondence between the selected wrinkle curve and the drawn target curve. We achieve the goal by parameterizing both curves using arc length. The second term is the Laplacian term, which preserves the fine details of the selected region and is mainly used to regularize the solution space. The weight λ_l is set to 0.2 in our implementation. We obtain the editing result by removing the details in the original wrinkle region and then applying the deformed wrinkle details to the target region via the copy and paste operation described above. Figure 5(d) shows the wrinkle editing result.

Fine-scale facial editing propagation. Once the transferred details at key frame t_k are refined, the system automatically propagates the edit to the remaining frames of the temporal window $[t_s, t_e]$. For the frames that are outside of the temporal window, we keep their facial motion details unchanged. The fine detail editing propagation problem can be formulated as the following least squares problem:

$$\arg\min_{\hat{D'}_{t_s},\ldots,\hat{D'}_{t_e}} \lambda_e E_e + \lambda_t E_t + \lambda_s E_s, \tag{18}$$

where the first term E_e measures the difference between the transferred motion detail $\hat{D'}_{t_k}$ and the edited motion detail $\hat{D}^*_{t_k}$ obtained from the last step:

$$E_e = \left\| \hat{D'}_{t_k} - \hat{D}^*_{t_k} \right\|^2. \tag{19}$$

The second term E_t represents the difference of temporal and spatial gradients of the transferred motion details between the source sequence and the target sequence:

$$\begin{aligned} E_t = & \sum_{t \in [t_s, t_e]} \left\| d_t(\hat{D'}_t) - d_t(R(\hat{D}_t)) \right\|^2 \\ & + \sum_{t \in [t_s, t_e]} \left\| d_s(\hat{D'}_t) - d_s(R(\hat{D}_t)) \right\|^2, \end{aligned} \tag{20}$$

where $R = \{R_v | v \in \bar{S}_t\}$ is the local frame rotations of all target mesh vertices as defined in Equation (16). $d_t(\hat{D'}_t)$ is the temporal gradient of motion details, which measures the difference of fine details $\hat{D'}_t$ between two consecutive frames and $d_s(\hat{D'}_t)$ is the spatial gradient of motion details, which measures the difference of fine details between adjacent vertices.

The last term E_s ensures that the transferred details on the boundary frames (t_s and t_e) are unchanged, resulting in the following objective function:

$$E_s = \left\| \hat{D'}_{t_s} - R(\hat{D}_{t_s}) \right\|^2 + \left\| \hat{D'}_{t_e} - R(\hat{D}_{t_e}) \right\|^2. \tag{21}$$

We reconstruct the final facial details with a linear solver and obtain the final transfer result according to Equation (5). For all the results in the paper, the weights λ_e, λ_t, and λ_s are set to 1.0, 0.1 and 1.0 respectively. Note that without user edits, this optimization is equivalent to the automatic transfer algorithm described above.

6 Experimental Results

Result Name	Mesh Resolution	Sequence Length	Number of Basis	Computing Time
Matt to Monster	80K	363	10	1min
Abu to Old Man	80K	534	12	1.2min
Robert to Monster	80K	199	6	0.7min
Abu to Dog1	80K	1373	28	2.9min
Max to Rejane	30K	200	5	0.3min
Abu to Dog2	80K	308	10	1.1min
Abu to Baby	80K	1373	30	3min
Abu to Nouillet	80K	522	10	0.8min
Man to Woman	1180K	348	10	8.2min

Table 1: *Statistics of facial performance transfer results.*

Result Name	Editing type	Refine Time	Num of Edited Frames
Abu to Dog1	Lare scale	91sec	23
Max to Rejane	Lare scale	2sec	1
Abu to Dog2	Fine scale	4sec	1
Abu to Baby	Fine scale	99sec	24
Man to Woman	Fine scale	5sec	1

Table 2: *Statistics of edited facial performance transfer results.*

Performance. We implemented our method in C++ on a PC with an Intel i7-2600 3.40GHZ eight core CPU and 8GB memory. For

an input mesh with 80K vertices, our algorithm takes about one second for mesh decomposition, three seconds for large-scale deformation transfer, and another two seconds for detailed motion transfer. Our large scale facial performance transfer algorithm only needs to transfer the motions of the source face bases to the target. Compared to a naive method that transfers the motions of all frames in the source sequence to the target, our method significantly reduces the computational cost while achieving similar quality. A comparison of these two methods is shown in the supplemental video. For transfer result editing and refinement, our method generally takes 0.5 seconds to refine the mesh in the edited frame (in both the large-scale transfer result editing and facial detail editing) and another $1-7$ seconds to propagate the edit to the other frames, depending on the number of frames in the temporal window. For a sequence that consists of $1,300$ frames, our method takes 4.5 seconds to propagate a large-scale edit to the whole sequence, and takes 6.8 seconds to propagate a fine-scale edit to the target. With this fast editing algorithm, our system allows the user to interactively edit the large-scale deformation and fine details of the transfer results.

Results. We validate our method by transferring a set of high fidelity facial performances captured from real subjects [Huang et al. 2011; Beeler et al. 2011] to the target faces with variant geometric shapes and details and editing the transfer results with our editing system. Tables 1 and 2 list statistics of all transfer and editing results shown in the paper and accompanying video, including mesh resolution, sequence length, number of blendshape bases, as well as the computation times. Except the"Man to Woman" sequence, which was obtained from [Beeler et al. 2011], all the other sequences were captured by [Huang et al. 2011]. Our results are best seen in video form.

Figure 6 illustrates a facial performance transfer result, where we directly transfer Matt's facial performance to a Monster without editing. Both the large-scale facial expressions and detailed wrinkle motions in the source facial performance are faithfully transferred to the target with convincing results. Please see the accompanying video for a full sequence of this transfer result ($1'08''$) and more direct transfer results generated by our method.

With large-scale editing, the animator can easily modify and refine the transfer results. Figure 7 illustrates the *Max to Rejane* transfer results before and after user editing. In this result, the animator uses a photo of the target subject (Rejane) as reference and edits the mouth and eye region in one frame. Our system automatically propagates the edit in both the spatial domain and temporal domain. The editing result not only preserves the user edit, but also generates smooth transitions along time. The full result sequence can be found in the accompanying video ($3'02''$). In the *Abu to Dog1* result shown in the accompanying video ($2'22''$), we transfer Abu's facial performance to a dog. Due to the large shape difference between the two faces, the automatic transfer algorithm generates some artifacts in the mouth region. With sparse user editing, our algorithm refines the transfer results in the mouth region and well preserves the facial performance transfer results in the other region.

Figure 8 shows the *Abu to Dog2* result ($3'33''$) with edited facial details, in which the angry expression of the dog is successfully exaggerated with the enhanced wrinkle details in the eye corner region. To this end, the animator selected a wrinkle beneath the right eye and pasted it to the eye corner region with the detail copy and paste operation. Our algorithm automatically refines the facial details in neighboring frames and generates convincing dynamic wrinkles in the result sequence. In the *Abu to Baby* result ($4'23''$), we weaken the details transferred to the target baby face to make the transfer results look more natural. In the *Man to Woman* result ($3'56''$), we weaken one wrinkle on the forehead of the target character and the system automatically propagates the edit in the temporal domain (see Figure 9). Note that the original facial data of the old man was obtained from [Beeler et al. 2011].

7 Conclusion

We present a controllable facial performance transfer method for retargeting high-fidelity facial performance data from one subject to another one. Different from previous facial performance retargeting methods, our method allows the user to control the transfer process in the spatial-temporal domain with minimal user interaction. To achieve this goal, we decouple high-fidelity facial expression data into large-scale facial deformation and fine-scale facial motion, as well as high-level facial feature lines. We combine a blendshape representation with deformation gradient techniques for transferring large-scale facial deformation, which not only reduces the computational cost for spatial-temporal facial editing but also significantly improves the flexibility for facial editing. In addition, we present an efficient method for transferring and editing fine-scale facial details. We demonstrate the power of our approach by transferring high-fidelity facial capture data from high-resolution source models to a wide variety of target models, including both human faces and non-human faces such as "monster" and "dog".

Similar to most facial performance transfer systems [Noh and Neumann 2001; Sumner and Popović 2004; Li et al. 2010], the quality of a retargeted animation highly depends on the accuracy of the correspondences between the source and target models. Currently, we take a semi-automatic approach for building dense correspondences between the source and target models. In practice, building accurate vertex-to-vertex correspondences between two mesh models is challenging. One possible solution to improve registration accuracy is to refine the correspondences based on user input on the retargeted motion. This could be resolved by combining our optimization with published techniques such as the face scan registration method of Huang et al. [2011].

Our facial retargeting scheme combines the power of geometric mapping and blendshape mapping. At one end, geometric mapping techniques can automatically transfer the facial performance from a source model to a target but does not allow the user to correct and improve the transferred performances. At the other end, the user can control the retargeted animation by modifying blendshape bases for the target model. However, this often requires skilled artists to manually create the facial rig for blendshape bases. Our work reaches a good balance between manual effort and animation quality and it only requires minimal user interaction to adjust and correct the retargeted facial expression in the spacetime domain.

Acknowledgement

The authors would like to thank Xing Zhao for her help in modeling and editing the "Dog" sequence. The authors also want to thank Stephen Lin for paper proofreading. The second author (Jinxiang Chai) was supported in part by the National Science Foundation under Grants No. IIS-1065384 and IIS-1055046.

References

AKHTER, I., SIMON, T., KHAN, S., MATTHEWS, I., AND SHEIKH, Y. 2012. Bilinear spatiotemporal basis models. *ACM Trans. Graph. 31*, 2 (Apr.), 17:1–17:12.

BARAN, I., VLASIC, D., GRINSPUN, E., AND POPOVIĆ, J. 2009. Semantic deformation transfer. *ACM Trans. Graph. 28*, 3 (July), 36:1–36:6.

Figure 6: *Results of direct transfer. The results are generated by directly transferring Matt's expressions (first row) to Monster (second row).*

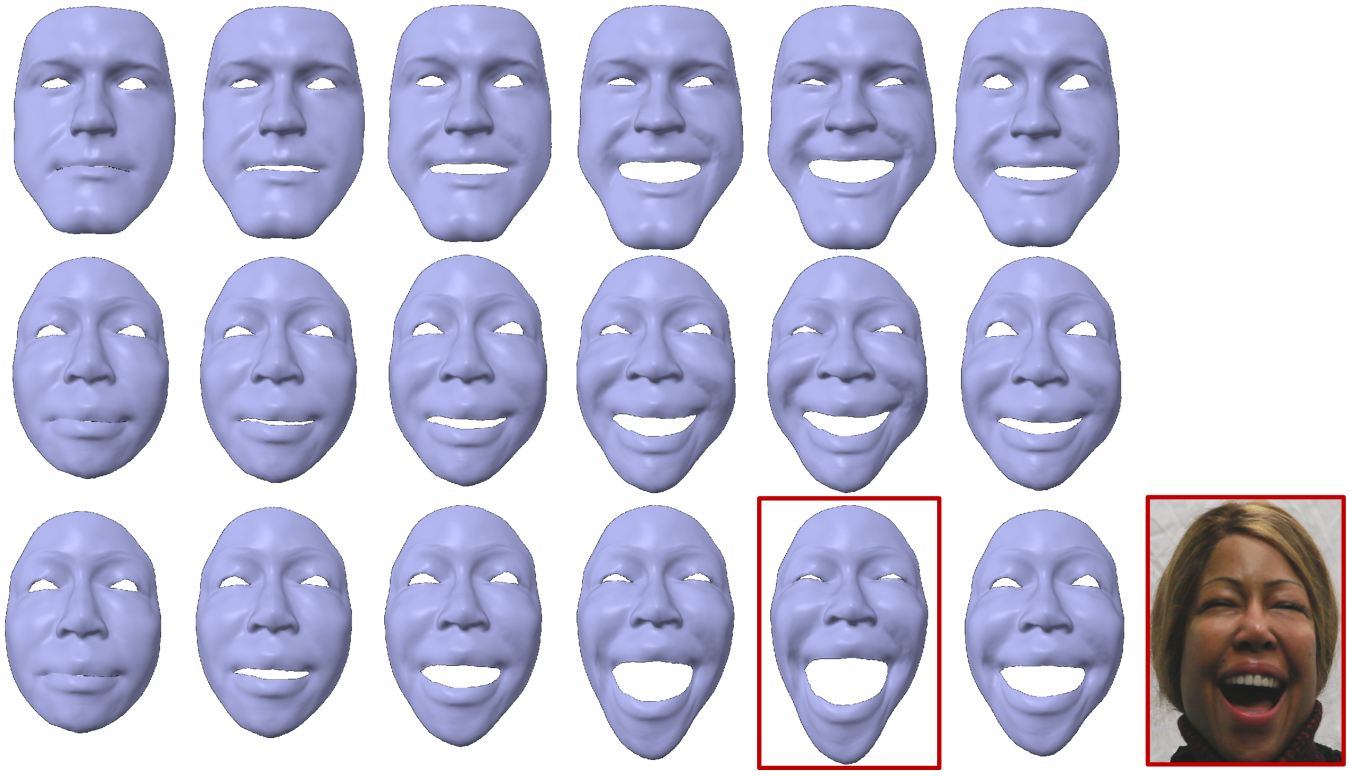

Figure 7: *Results of large-scale editing with an image as reference. First row: Max's source expression. Second row: automatic transfer result on Rejane. Third row: edited result. The mesh model in the red rectangle is edited by the user with the right image as reference. The other models are automatically edited by the blendshape-based optimization.*

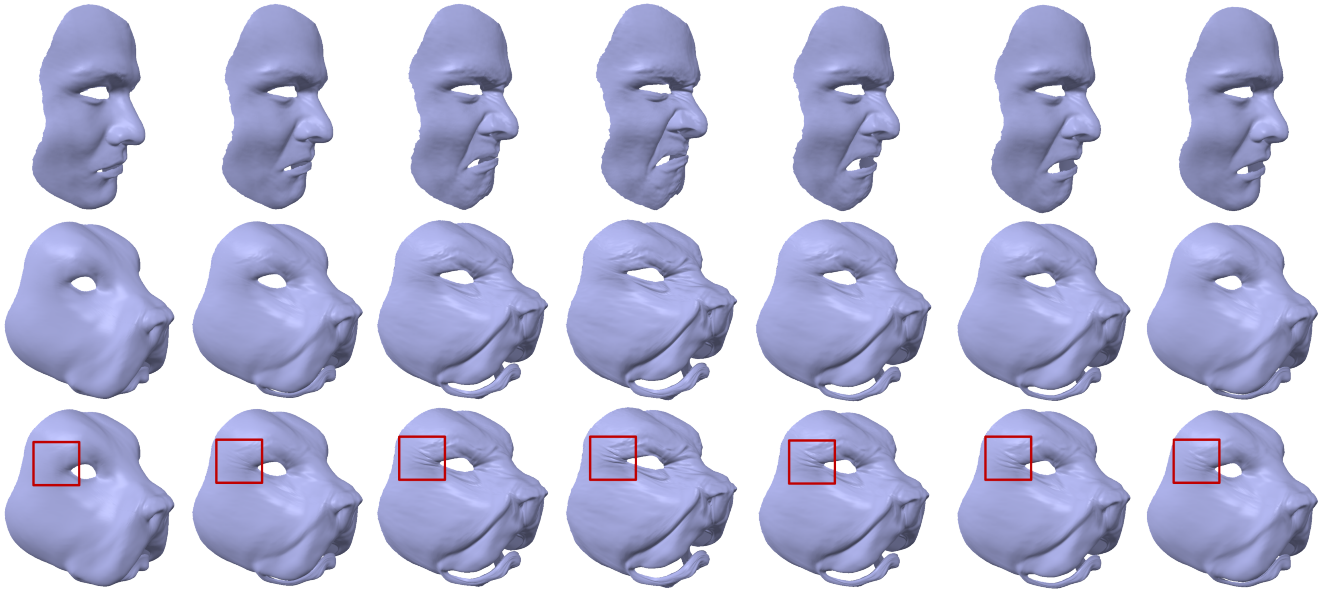

Figure 8: *Results of detail editing. The results are generated by transferring Abu's expressions (first row) to a dog face. The second row shows the original facial performance transfer results. In the third row, the wrinkles in the eye corner region are generated by our detail editing tool. The user performed three copy and paste operations on the frame shown in the fifth column and the system automatically propagated the edit in the temporal domain.*

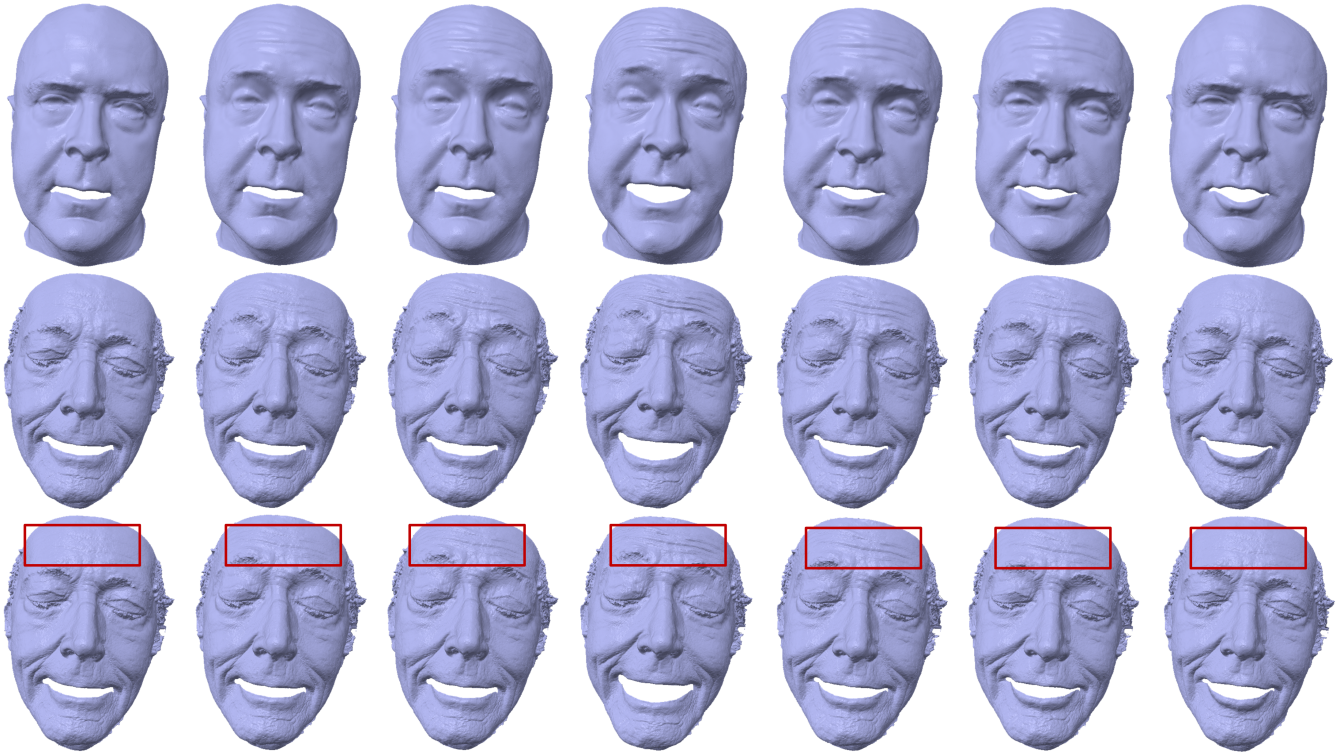

Figure 9: *Results of detail editing. The results are generated by transferring an old man's expressions (first row) to an old woman face. The second row shows the original facial performance transfer results. In the third row, the wrinkles on the forehead region are generated by our detail editing tool. The user weakened the wrinkles on the forehead on the frame shown in the fifth column and the system automatically propagated the edit in the temporal domain.*

BEELER, T., HAHN, F., BRADLEY, D., BICKEL, B., BEARDSLEY, P., GOTSMAN, C., SUMNER, R. W., AND GROSS, M. 2011. High-quality passive facial performance capture using anchor frames. *ACM Trans. Graph. 30*, 4, 75:1–75:10.

BERMANO, A. H., BRADLEY, D., BEELER, T., ZUND, F., NOWROUZEZAHRAI, D., BARAN, I., SORKINE-HORNUNG, O., PFISTER, H., SUMNER, R. W., BICKEL, B., AND GROSS, M. 2014. Facial performance enhancement using dynamic shape space analysis. *ACM Trans. Graph. 33*, 2 (Apr.), 13:1–13:12.

BHAT, K. S., GOLDENTHAL, R., YE, Y., MALLET, R., AND KOPERWAS, M. 2013. High fidelity facial animation capture and retargeting with contours. In *Proceedings of the 12th ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, ACM, New York, NY, USA, SCA '13, 7–14.

BICKEL, B., BOTSCH, M., ANGST, R., MATUSIK, W., OTADUY, M., PFISTER, H., AND GROSS, M. 2007. Multi-scale capture of facial geometry and motion. *ACM Trans. Graph. 26*, 3 (July).

BICKEL, B., LANG, M., BOTSCH, M., OTADUY, M. A., AND GROSS, M. 2008. Pose-space animation and transfer of facial details. In *Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, SCA '08, 57–66.

BOUAZIZ, S., WANG, Y., AND PAULY, M. 2013. Online modeling for realtime facial animation. *ACM Trans. Graph. 32*, 4 (July), 40:1–40:10.

BUCK, I., FINKELSTEIN, A., JACOBS, C., KLEIN, A., SALESIN, D., SEIMS, J., SZELISKI, R., AND TOYAMA, K. 2000. Performance-driven hand-drawn animation. In *Proceedings of NPAR 2000*.

CAO, C., WENG, Y., LIN, S., AND ZHOU, K. 2013. 3d shape regression for real-time facial animation. *ACM Trans. Graph. 32*, 4 (July), 41:1–41:10.

CHAI, J., XIAO, J., AND HODGINS, J. 2003. Vision-based control of 3D facial animation. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 193-206.

CHUANG, E., 2004. Analysis, synthesis, and retargeting of facial expressions. PhD thesis, Stanford University.

HAVALDAR, P. 2006. Performance driven facial animation. *ACM SIGGRAPH Course Notes*.

HUANG, H., CHAI, J., TONG, X., AND WU, H.-T. 2011. Leveraging motion capture and 3d scanning for high-fidelity facial performance acquisition. *ACM Trans. Graph. 30*, 4, 74:1–74:10.

IMAGE-METRICS. Live driver (face tracking). `http://www.image-metrics.com`.

JAMES, D. L., AND TWIGG, C. D. 2005. Skinning mesh animations. *ACM Transactions on Graphics 24*, 3, 399–407.

JOSHI, P., TIEN, W. C., DESBRUN, M., AND PIGHIN, F. 2003. Learning controls for blendshape based realistic facial animation. In *Proceedings of ACM SIGGRAPH/Eurographics Symposium on Computer Aimation*. 187–192.

KHOLGADE, N., MATTHEWS, I., AND SHEIKH, Y. 2011. Content retargeting using parameter-parallel facial layers. In *Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, ACM, New York, NY, USA, SCA '11, 195–204.

LAU, M., CHAI, J., XU, Y.-Q., AND SHUM, H.-Y. 2007. Face poser: interactive facial modeling using model priors. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*.

LAU, M., CHAI, J., XU, Y.-Q., AND SHUM, H. 2009. Face poser: Interactive modeling of 3d facial expressions using facial priors. *ACM Transactions on Graphics 29*, 1, 3:1–3:17.

LEWIS, J., AND ANJYO, K. 2010. Direct manipulation blendshapes. *IEEE Computer Graphics and Applications 30*, 4, 42–50.

LI, Q., AND DENG, Z. 2008. Orthogonal-blendshape-based editing system for facial motion capture data. *IEEE Comput. Graph. Appl. 28*, 6 (Nov.), 76–82.

LI, H., WEISE, T., AND PAULY, M. 2010. Example-based facial rigging. *ACM Transactions on Graphics 29*, 4, 32:1–32:6.

LI, H., YU, J., YE, Y., AND BREGLER, C. 2013. Realtime facial animation with on-the-fly correctives. *ACM Trans. Graph. 32*, 4 (July), 42:1–42:10.

MEYER, M., AND ANDERSON, J. 2007. Key point subspace acceleration and soft caching. In *ACM Transactions on Graphics*. 26(3): Article No. 74.

NOH, J., AND NEUMANN, U. 2001. Expression cloning. In *Proceedings of ACM SIGGRAPH 2001*. 277-288.

PIGHIN, F., AND LEWIS, J. 2005. Digital face cloning. *ACM SIGGRAPH Course Notes*.

SEOL, Y., SEO, J., KIM, P. H., LEWIS, J. P., AND NOH, J. 2011. Artist friendly facial animation retargeting. In *Proceedings of the 2011 SIGGRAPH Asia Conference*, ACM, New York, NY, USA, SA '11, 162:1–162:10.

SEOL, Y., LEWIS, J., SEO, J., CHOI, B., ANJYO, K., AND NOH, J. 2012. Spacetime expression cloning for blendshapes. *ACM Trans. Graph. 31*, 2 (Apr.), 14:1–14:12.

SORKINE, O., COHEN-OR, D., LIPMAN, Y., ALEXA, M., RÖSSL, C., AND SEIDEL, H. 2004. Laplacian surface editing. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, ACM, 175–184.

SUMNER, R. W., AND POPOVIĆ, J. 2004. Deformation transfer for triangle meshes. *ACM Trans. Graph. 23*, 3 (Aug.), 399–405.

TAUBIN, G. 1995. A signal processing approach to fair surface design. In *Proceedings of the 22Nd Annual Conference on Computer Graphics and Interactive Techniques*, ACM, New York, NY, USA, SIGGRAPH '95, 351–358.

TENA, J. R., DE LA TORRE, F., AND MATTHEWS, I. 2011. Interactive region-based linear 3d face models. *ACM Trans. Graph. 30*, 4 (July), 76:1–76:10.

VLASIC, D., BRAND, M., PFISTER, H., AND POPOVIĆ, J. 2005. Face transfer with multilinear models. *ACM Transactions on Graphics 24*, 3, 426–433.

WEISE, T., LI, H., VAN GOOL, L., AND PAULY, M. 2009. Face/off: live facial puppetry. In *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, ACM, New York, NY, USA, SCA '09, 7–16.

ZHANG, L., SNAVELY, N., CURLESS, B., AND SEITZ, S. 2004. Spacetime faces: high resolution capture for modeling and animation. *ACM Transactions on Graphics 23*, 3, 548–558.