Spacetime Expression Cloning for Blendshapes

YEONGHO SEOL
KAIST and Weta Digital
J.P. LEWIS
Weta Digital
JAEWOO SEO and BYUNGKUK CHOI
KAIST
KEN ANJYO
OLM Digital and JST CREST
and
JUNYONG NOH
KAIST

The goal of a practical facial animation retargeting system is to reproduce the character of a source animation on a target face while providing room for additional creative control by the animator. This article presents a novel spacetime facial animation retargeting method for blendshape face models. Our approach starts from the basic principle that the source and target movements should be similar. By interpreting *movement* as the derivative of position with time, and adding suitable boundary conditions, we formulate the retargeting problem as a Poisson equation. Specified (e.g., neutral) expressions at the beginning and end of the animation as well as any user-specified constraints in the middle of the animation serve as boundary conditions. In addition, a model-specific prior is constructed to represent the plausible expression space of the target face during retargeting. A Bayesian formulation is then employed to produce target animation that is consistent with the source movements while satisfying the prior constraints. Since the preservation of temporal derivatives is the primary goal of the optimization, the retargeted motion preserves the rhythm and character of the source movement and is free of temporal jitter. More importantly, our approach provides spacetime editing for the popular blendshape representation of facial models, exhibiting smooth and controlled propagation of user edits across surrounding frames.

Categories and Subject Descriptors: I.3.7 [**Computer Graphics**]: Three-Dimensional Graphics and Realism—*Animation*

General Terms: Algorithms, Experimentation

Additional Key Words and Phrases: Face, retargeting, movement, spacetime, editing

ACM Reference Format:

Seol, Y., Lewis, J. P., Seo J., Choi, B., Anjyo, K., and Noh, J. 2012. Spacetime expression cloning for blendshapes. ACM Trans. Graph. 31, 2, Article 14 (April 2012), 12 pages.
DOI = 10.1145/2159516.2159519
http://doi.acm.org/10.1145/2159516.2159519

1. INTRODUCTION

Creating realistic facial animation is an important topic in computer graphics with immediate applications in movies and computer animations. As humans are apt to recognize subtle artifacts in facial animations, it has never been easy to create realistic facial animation using keyframing even for skilled animators. Instead of relying on manual keyframing, realistic animations increasingly use motion capture of human actors. This solves the problem of obtaining realistic facial motion, but introduces the problem of *retargeting* the motion to the desired facial model. The target facial model generally has differing proportions than the original actor, and frequently is not even human. However, due to an anthropomorphic principle, the target face must be human-like in order to convey acting to a human audience. Recent movies such as *King Kong* and *Avatar* are well-known examples of facial retargeting.

The goal of a practical facial animation retargeting system is to: (1) consistently reproduce a source animation on a target face while (2) providing room for additional editing by an animator. The first goal has been the main subject of previous retargeting approaches, and many successful results have been reported [Sumner and Popovic 2004; Bickel et al. 2007; Weise et al. 2009]. In contrast, the second goal has been neglected, even though additional editing is indispensable for professional-quality animation [Havaldar 2006]. Although retargeting methods based on a blendshape model [Joshi et al. 2003; Chuang and Bregler 2005; Li et al. 2010] might be considered for the second goal, automatic retargeting results that have keyframes at every frame are very difficult to edit. Motivated by these needs, we present a novel retargeting

This work was supported by KOCCA/MCST (R2010050008-00000002 and R2011050109-00000001).
Authors' addresses: Y. Seol, KAIST and Weta Digital; email: seolyeongho@kaist.ac.kr; J. P. Lewis, Weta Digital; J. Seo, B. Choi, KAIST; K. Anjyo, OLM Digital and JST CREST; J. Noh (corresponding author), KAIST; email: junyongnoh@kaist.ac.kr.

DOI 10.1145/2159516.2159519
http://doi.acm.org/10.1145/2159516.2159519

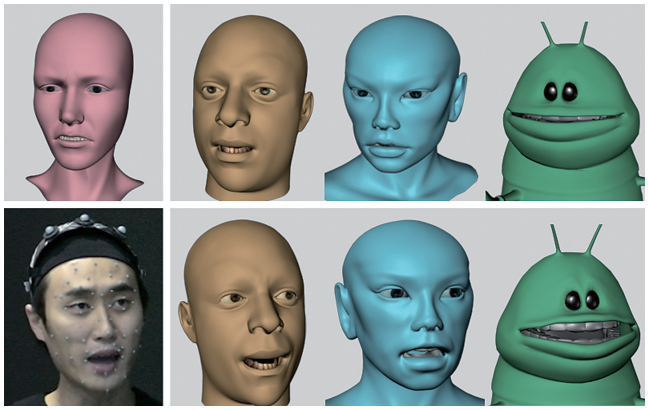
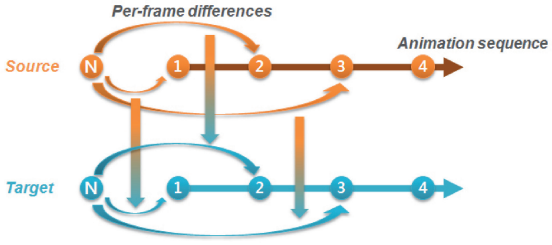
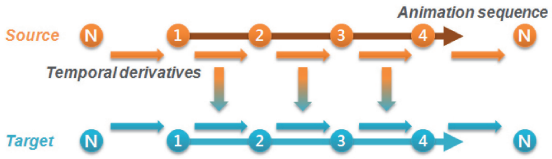


Fig. 1. Facial animations are faithfully retargeted to various models with a different number of blendshapes using our method. The brown face model is used with permission of Jason Osipa.



(a) *The per-frame transfer*



(b) *The velocity domain transfer*

Fig. 2. Whereas previous approaches conducted retargeting independently at every frame (a), movement matching transfers the temporal derivative of source animation (b). N denotes a neutral pose face.

system that provides both consistent reproduction of source animation on a target face and efficient spacetime editing.

Our novel approach to facial animation retargeting starts from the proposition that *the movement of the source and target faces should be similar*. Defining *movement* as the time derivative of position and matching movement between source and target face points leads to a velocity domain boundary value problem. Unlike previous methods where retargeting is conducted independently at every frame (Figure 2(a)), our approach of matching the velocity of the source and target (Figure 2(b)) has obvious advantages. First, the influence of additional modifications by the animator propagates smoothly through surrounding frames, minimizing user interaction and increasing productivity. Second, when constraints on expressiveness are imposed (which is usually the case), least squares matching in the velocity domain may preserve the overall movement better than per-frame matching (Figure 3).

A target face-specific prior model is integrated into the movement matching formulation as a constraint on expressiveness. By applying Principal Component Analysis (PCA) to the existing facial expressions of the target blendshape model, we build a prior

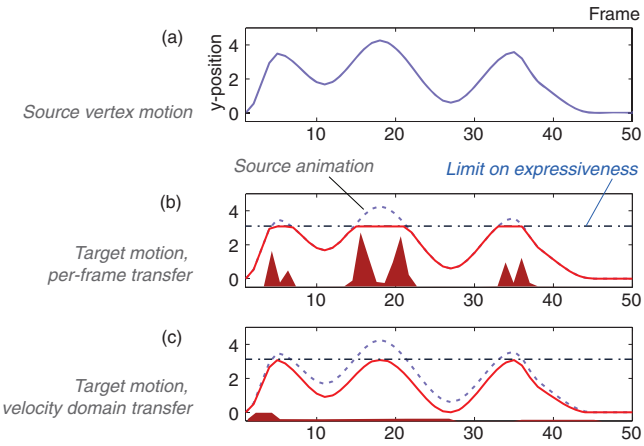


Fig. 3. Least squares matching of derivatives transfers motion better than per-frame matching when constraints are present. (a) y-position of a point on the source face. (b) per-frame matching attempts to transfer the motion exactly. When the target model cannot reproduce the desired motion (due to a constrained weight range (hard constraint) or a prior (soft constraint)) the transferred motion does not resemble the source. The shaded area indicates the squared error in derivative between source and target. (c) the least-squares objective prefers many small errors over a few large errors, thus movement matching preserves the overall shape of the motion.

model that reflects the probability of the retargeted results being in a plausible space of expressions for the target face. We use a Maximum A Posteriori (MAP) formulation to produce an animation consistent with the source movement while satisfying the prior imposed on the target model. Using this prior our formulation constructs a simple and efficient linear system to solve for blendshape weights. We show that our velocity domain transfer works better with the prior (or any other constraints) than per-frame transfer does, leading to more temporally coherent and plausible results.

Our formulation specifically targets the popular blendshape parameterization of facial models. Blendshapes are a simple linear combination of basis shapes, however, the basis is not orthogonal since the individual shapes are constructed to have interpretable roles such as "raise the right eyebrow." While blendshapes resemble PCA models in being a linear combination of basis vectors, the individual eigenvectors in a PCA model generally lack any semantic interpretation due to their orthogonality [Lewis and Anjyo 2010]. Thanks to their combination of direct geometric authoring and a meaningful parameterization, the blendshape approach has been the predominant choice for realistic computer graphics characters.

The movement matching criterion resembles various gradient domain image and mesh editing techniques that involve a Poisson equation (e.g., Pérez et al. [2003]), but the derivative here is with respect to time rather than space and the solution is expressed in terms of a blendshape model, leading to a different mathematical formulation than in previous gradient domain approaches. The idea of minimizing the difference in velocity has long been used for smooth blending of different body motions [Lee et al. 2002; Kovar and Gleicher 2003], and the literature on editing motion capture includes a variety of approaches that are distantly related to ours as they involve global optimization of temporal relationships.

Figure 4 illustrates the overall movement matching process. Given a source animation, a target face with blendshapes, and their correspondences, a source-target calibration step matches the scale and proportion between the two (Section 3). Then, the source

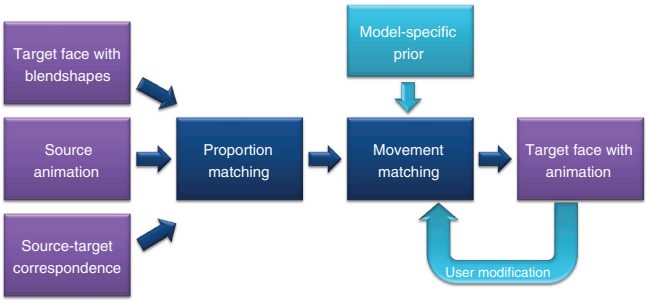


Fig. 4. The overall process of our retargeting method.

animation is retargeted to the target face based on movement matching (Section 4) and the target-specific prior model (Section 5). The spacetime editing capability of our method allows the animator to freely modify the results at any frame, without affecting temporal coherency (Section 6). Figure 1 shows examples of our results on various models.

2. RELATED WORK

Facial modeling and animation have a rich history of research since the pioneering work of Parke [1972]. In this section, we will review only work related to facial animation retargeting. For complete references on facial modeling and animation, see Parke and Waters [1996] and Deng and Noh [2007].

2.1 Facial Animation Retargeting

There is a strong body of existing research on facial animation retargeting. After initial work by Williams [1990], the use of sparse correspondences and interpolation schemes [Bickel et al. 2007; Ma et al. 2008] has been a popular choice to deform 3D facial geometry. These approaches often assume that the source and target faces are of the same shape or share the same mesh connectivity. The work by Noh and Neumann [2001] and Sumner and Popovic [2004] found dense correspondences between the source and target to transfer either motion vectors or deformation gradients. These approaches are more general in that transfer is possible across different mesh structures. However, the lack of intuitive high-level editing makes these techniques less suited for professional projects where editing subsequent to the retargeting is required.

Another noteworthy approach to facial animation retargeting is shape blending. These retargeting methods attempt to match high-dimensional vectors of source motion as linear combinations of a limited set of target shapes. The set of facial shapes can consist of facial muscle actions [Ekman and Friesen 1977; Choe et al. 2001; Reveret and Essa 2001], user-defined blendshapes [Chuang and Bregler 2002; Joshi et al. 2003; Pyun et al. 2003; Deng et al. 2006; Sagar and Grossman 2006], facial scan examples [Blanz and Vetter 1999; Zhang et al. 2004; Vlasic et al. 2005; Weise et al. 2009], or generic blendshape models [Li et al. 2010]. In Sumner et al. [2005], a shape blending approach was extended to the area of nonlinear blending to generate various deformations. Recently, Weise et al. [2009] presented a high-quality real-time facial puppetry and expression transfer system that employs person-specific facial shapes extracted by PCA. Although these mathematically extracted shapes recreated the source animation well, they are not meaningful shapes for artist-driven editing.

Although the retargeting methods surveyed before have different details, they share the common principle that retargeting is conducted independently at every frame (Figure 2(a)). The retargeted facial animation may be reasonable, but there are several drawbacks to this approach. In the presence of constraints on expressiveness the target motion may not resemble the source (Figure 3). In addition, modifying the retargeted animation can be tedious. Our movement matching method excels in these aspects.

2.2 Expressiveness of a Face Model

In practice, a blendshape model's range of motion rarely matches that of the source performance. While conventional blendshape models cannot exactly match a performance due to the limitations of manual sculpting, the mismatch between the model and the performance occurs even with PCA models [Blanz and Vetter 1999; Chuang and Bregler 2005; Weise et al. 2009] because the model is typically built from a training performance that only approximately contains the character's full range of movement. In these cases blendshape-based transfer has been observed to fit the source with large and mostly canceling positive and negative shape combinations, and applying the corresponding weights on the target face results in poor retargeting [Chuang and Bregler 2002]. In order to prevent these large extrapolations, methods such as Choe et al. [2001], Chuang and Bregler [2002], and Bregler et al. [2002] strictly constrain the blendshape weights to be in a certain range. The incorporation of a learned facial prior [Blanz and Vetter 1999; Lau et al. 2009] is a more general solution. Lau et al. [2009] modeled a prior from face data as a mixture of factor analyzers and used the prior for face editing. In many practical retargeting cases, however, the target is a fantasy character such as an alien and the facial scan data required to construct a target morphable model or suitable prior is not available. In these cases, standard practice is to use an artist-created blendshape model that spans the plausible face space for the character. In our method, the existing blendshape models are analyzed and used to estimate a plausible face space.

2.3 Animation Editing and Propagation

For corrections or adjustments on the retargeted animation, typically additional manual work by an animator is required. To make this process easier, several methods have been introduced. Li and Deng [2008] built an independent blendshape model for each facial region and propagated the influence from the user editing to the entire face using a hierarchical model. Lewis and Anjyo [2010] provided direct manipulation of vertices based on a blendshape model and Zhang et al. [2004] performed a local and adaptive blend of basis shapes. High-level editing such as emotion control was shown in Cao et al. [2003] and Chuang and Bregler [2005]. In contrast to the preceding work that considers editing in the spatial domain, there is not much work that performs editing in the temporal domain. Choe et al. [2001] fitted the discrepancy caused by edits into a B-spline curve and composited it with the original parameter curve. Li and Deng [2008] fitted and smoothly manipulated a Catmull-Rom spline based on the weighted sequence. Both methods are capable of generating smooth results. However, original source characteristics are rarely preserved after the edits. Ma et al. [2009] proposed a style learning method that applies learned style to the similar frames in the sequence. While it is an effective strategy, smooth propagation of edits is not considered. In contrast, by design, our proposed movement matching seamlessly applies the user modifications across the surrounding frames while preserving the characteristics of the original source animation.

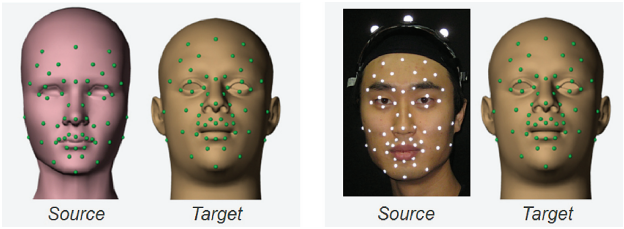


Fig. 5. 54 corresponding markers are used for (a) mesh to mesh retargeting and (b) motion capture to mesh retargeting.

3. PREPROCESSING

3.1 Data Acquisition

A source animation can be obtained from existing facial animation or facial motion capture. We represent facial motion on a motion capture sequence or facial animation using a pattern of $n = 54$ markers as shown in Figure 5. This choice of marker placement is our standard workflow but other sensible possibilities would be equally valid. During the facial motion capture, we recorded the actor's frontal face with a video camera for the final comparison. The actor was asked to make a neutral expression at the beginning and at the end of the capture. The neutral expressions become boundary conditions for movement matching.

Normalization is performed on the acquired source animation data. The head motion is estimated and stored for later resynthesis. As in Deng et al. [2006], translation and rotation are adjusted using Procrustes analysis. For each motion capture frame, $n \times 3$ matrix $\mathbf{X}_l$ that contains xyz positions of markers is constructed. The matrix $\mathbf{Y}$ for the neutral pose is also constructed. A Singular Value Decomposition $\mathbf{Y}^T\mathbf{X}_l = \mathbf{U}_l\mathbf{D}_l\mathbf{V}_l^T$ is used to determine the rotation matrix $\mathbf{R}_l = \mathbf{V}_l\mathbf{U}_l^T$.

A target model consists of a neutral face and a set of blendshape targets. We have chosen arbitrary face models with differing numbers of blendshape basis vectors to demonstrate the generality of our approach. The positions of n markers corresponding to the source markers are manually identified on each target face.

3.2 Scale-Proportion Matching

Retargeting methods generally require that the source and target faces have approximately similar scale and proportion. We deform the marker positions on each target blendshape to the shape of the source face. Similar to Bickel et al. [2007], Orvalho et al. [2008], Radial Basis Functions (RBF) determine the warping

$$\mathbf{d}(\mathbf{x}) = \sum_{i=1}^{n} \mathbf{w}_i \phi(\mathbf{x}, \mathbf{c}_i) + \mathbf{q}(\mathbf{x}),$$

where $\mathbf{x} \in \mathbb{R}^3$ is the marker position on the neutral target face. $\mathbf{w}_i, \mathbf{c}_i \in \mathbb{R}^3$ are the weights and centers of the RBF (the centers coincide with the correspondence points on the neutral target face). For training, the points on the neutral source face are used for $\mathbf{d}(\mathbf{x})$. A kernel function $\phi(\mathbf{x}, \mathbf{c}_i) = \|\mathbf{x} - \mathbf{c}_i\|$ minimizes the bending energy of the deformation and $\mathbf{q} : \mathbb{R}^3 \rightarrow \mathbb{R}$ is a linear polynomial. By using the correspondences on the neutral faces as constraints and solving the resulting linear system, the weights $\mathbf{w}_i$ and the coefficients of $\mathbf{q}$ can be found. Because of the linear term $\mathbf{q}$, the function $\mathbf{d}$ can exactly reproduce affine motions.

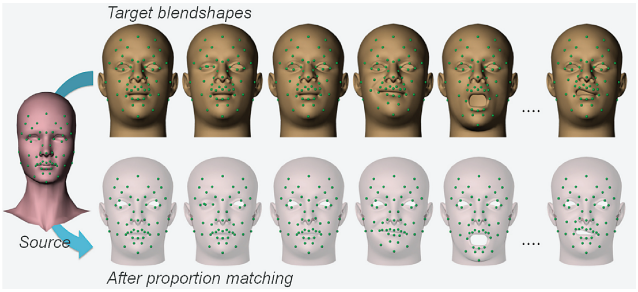


Fig. 6. The corresponding markers on the blendshapes are deformed to match the scale and proportion of the source face (blendshape models are overlaid for the visualization purpose).

Now we deform the marker positions of each target blendshape using the computed $\mathbf{d}$. A set of n marker positions on m blendshapes $\mathbf{b}_{i,j} \in \mathbb{R}^3$, $\{i = 1, \ldots, n\}$, $\{j = 1, \ldots, m\}$ are deformed to $\mathbf{b}^*_{i,j}$

$$\mathbf{b}^*_{i,j} = \mathbf{d}(\mathbf{b}_{i,j}),$$

$\mathbf{b}^*_{i,j}$ reflect the deformed marker positions of the target blendshapes that approximate the shape of the source (Figure 6). The computed $\mathbf{b}^*_{i,j}$ replace $\mathbf{b}_{i,j}$ in the subsequent sections.

4. MOVEMENT MATCHING

This section describes how we use the movement matching principle to compute blendshape weights for the target face from the animation of source markers. A source and a target face can be represented as

$$\begin{aligned} \mathbf{f}_{src} &= \mathbf{n}_{src} + \mathbf{s}, \\ \mathbf{f}_{tgt} &= \mathbf{n}_{tgt} + \mathbf{t} = \mathbf{n}_{tgt} + \mathbf{Bw}, \end{aligned} \tag{1}$$

where $\mathbf{f}_{src}$ and $\mathbf{f}_{tgt}$ are the vectorized marker positions of size $3n$ for the source and the target, respectively, arranged in xyz order. The vectors $\mathbf{n}_{src}$ and $\mathbf{n}_{tgt}$ are neutral face shape vectors and the vectors $\mathbf{s}$ and $\mathbf{t}$ represent deltas from neutrals. $\mathbf{B}$ is the $3n \times m$ blendshape matrix whose columns represent deltas of each blendshape from the neutral position. $\mathbf{w}$ is the blendshape weight vector of size m. In the following, $\mathbf{n}_{src}$ and $\mathbf{n}_{tgt}$ are ignored and assumed to be added back in the end.

The source and the target animation sequences can be represented as matrices $\mathbf{F}_{src}$ and $\mathbf{F}_{tgt}$ of size $3n \times f$ whose columns represent the positions of correspondences for each frame. $\mathbf{W}$ is an $m \times f$ matrix containing the weights for each frame in its columns. Here, f denotes the number of animation frames. Then, the source and the target animations can be represented as

$$\begin{aligned} \mathbf{F}_{src} &= \mathbf{S} \\ \mathbf{F}_{tgt} &= \mathbf{T} = \mathbf{BW}. \end{aligned}$$

Unlike most previous work that assumes the per-frame positions of the source and the target are similar, our method assumes that their movements are similar. This can be expressed as $\mathbf{DS}^T \approx \mathbf{DT}^T$, or $\mathbf{DS}^T \approx \mathbf{D}(\mathbf{BW})^T$, where $\mathbf{D}$ is a derivative operator that works on all the vertices at once. With known boundary conditions from Section 3, our minimization becomes a Poisson problem

$$\min_{\mathbf{W}} \|\mathbf{D}(\mathbf{BW})^T - \mathbf{DS}^T\|_F^2, \tag{2}$$

where F denotes the Frobenius norm. See the appendix for the detailed derivation. The resulting target animation using $\mathbf{W}$ has similar

movement to the source animation and is temporally coherent by construction.

When the employed boundary values are identical to the values from the per-frame transfer without any constraint, the results for both per-frame and velocity domain are the same. However, the advantages of our movement matching become apparent when the retargeting is constrained or user editing is required, which is generally the case in practice. These advantages are described in Sections 5 and 6.

5. MODEL-SPECIFIC PRIOR INFORMATION

As addressed in Bregler et al. [2002] and Chuang and Bregler [2002], naively applying a blendshape model to a source animation can produce very large positive and negative weights. Unwanted extrapolations caused by large weight values lead to the following problems.

(1) Artists typically create a blendshape model with the intended weight range $0 \leq w_i \leq 1$ in mind. Weights that are much outside this range correspond to expressions that lie outside the valid range of poses for the model.

(2) Subsequent user modification becomes unintuitive and nontrivial. Adjusting a blendshape model that is far out of its intended range is a daunting task (note that typical professional models have 50–100 or more basis shapes to explore).

Bregler et al. [2002] and Chuang and Bregler [2002] used Quadratic Programming (QP) and the nonnegative least-squares method to limit the range of weight values. Similarly, Pighin et al. [2002] addressed this by using soft limits on the blendshape weights. However, simply restricting weight values to a certain range is somewhat arbitrary and limits the expressiveness of the combined blendshapes, sometimes failing to achieve consistent movement with the source (Figure 3). We incorporate a model-specific prior to fix the problem. In the following, the formulation of a prior specifically for the velocity domain will be detailed and its effect on retargeting quality will be demonstrated.

Similar to Blanz and Vetter [1999] and Lau et al. [2009], we formulate the probability of a new face being in the valid space of blendshapes. Using this measure as a prior probability, we incorporate the least-squares optimization for the transfer task into a MAP formulation giving the log posterior as sum of data likelihood and prior terms. The prior model can be easily constructed using data from either of the two sources. Arbitrary animations of the target model can be a great source if they happen to be available. If not, an "animation" whose jth frame is the jth blendshape basis shape is used, since a quality blendshape model is by construction a good parameterization of the space of face shapes.

Applying PCA to this data gives a new eigenvector basis $\mathbf{e}_k$ and corresponding eigenvalues λ_k. The model is then represented as

$$\mathbf{f}_{tgt} = \mathbf{E}\mathbf{v} + \bar{\mathbf{t}}, \tag{3}$$

where $\mathbf{E}$ is a $3n \times h$ matrix with h eigenvectors with nonzero eigenvalues in its columns. $\mathbf{v}$ is the PCA coefficient vector of the model, and $\bar{\mathbf{t}}$ is the mean shape vector of the data.

We fit a multivariate normal distribution to our PCA data to construct the prior model. The probability for a coefficient vector $\mathbf{v}$ being a plausible target face is given by

$$p(\mathbf{v}) \sim \exp\left[-\frac{1}{2}\sum_{k=1}^{h}\frac{(v_k)^2}{\lambda_k}\right].$$

The cost function for the prior at a single frame can be expressed in vector form as

$$E_{prior} = \mathbf{v}^T\mathbf{\Lambda}^{-1}\mathbf{v},$$

where $\mathbf{\Lambda}^{-1}$ is the inverse of diagonal matrix of eigenvalues.

The next step is to express the prior probability in terms of the blendshape weight vector $\mathbf{w}$ instead of $\mathbf{v}$, in order to solve it together with movement matching. Note that any facial expression of the target face that is expressed with a blendshape combination can also be expressed via PCA eigenvectors. By equating Eq. (1) and (3), we can solve for $\mathbf{v}$ in terms of $\mathbf{w}$

$$\mathbf{v} = \mathbf{E}^T(\mathbf{B}\mathbf{w} + \mathbf{n}_{tgt} - \bar{\mathbf{t}}),$$

where $\mathbf{E}^T\mathbf{E} = \mathbf{I}$. Eq. (2) defines the matching cost between the source and target. As a least-squares problem such as Eq. (2) can be considered as resulting from a maximum likelihood formulation [Hertzmann 2004], the addition of the prior model allows standard Bayesian approaches to be applied. The MAP formulation minimizes the cost function

$$E = E_{match} + \alpha\, E_{prior}, \tag{4}$$

where α is a weighting factor that regulates the contribution from the prior probability to the matching quality. The particular form of our prior also gives a natural interpretation to Eq. (4) as regularizing the least-squares solution with a function norm defined in the reproducing kernel Hilbert space (RKHS) K derived from the data covariance [Wahba 1990], that is, $\|\mathbf{f}\|_K^2 = \mathbf{v}^T\mathbf{\Lambda}^{-1}\mathbf{v}$. The value of α is empirically decided depending on the size of the faces. In our tests, small α such as 0.01 (Figure 7(f)) worked well to account for the prior model while a large α tends to overly constrain target animation as in Figure 7(e). The eyes do not close all the way because of the strong prior.

In movement matching, the blendshape weight values are stored in the matrix $\mathbf{W}$ of size $m \times f$. In order to make a linear system that reflects both the likelihood of matching and the model-specific prior, we convert the matrix $\mathbf{W}$ into a vector of size mf, which is necessary for $\mathbf{W}$ and the prior to be compatible and compactly represented in a single equation. Vectorizing Eq. (2) can be achieved by utilizing the Kronecker product (e.g., $\text{vec}(\mathbf{AXB}) = (\mathbf{B}^T \otimes \mathbf{A})\text{vec}(\mathbf{X})$, where $\text{vec}(\mathbf{X})$ denotes the vectorization of the matrix $\mathbf{X}$ formed by stacking the columns of $\mathbf{X}$ into a single column vector).

$$\begin{aligned} E_{match} &= \|\mathbf{D}(\mathbf{BW})^T - \mathbf{D}\mathbf{S}^T\|_F^2 = \|\mathbf{BWD}^T - \mathbf{S}\mathbf{D}^T\|_F^2 \\ &= \|(\mathbf{D}\otimes\mathbf{B})\text{vec}(\mathbf{W}) - (\mathbf{D}\otimes\mathbf{I})\text{vec}(\mathbf{S})\|^2 \end{aligned} \tag{5}$$

Similarly, an expression for the prior over the entire range of f frames can be denoted as $E_{prior} = \bar{\mathbf{v}}^T(\mathbf{I}_f \otimes \mathbf{\Lambda}^{-1})\bar{\mathbf{v}}$, where $\mathbf{I}_f$ is an identity matrix of size f and $\bar{\mathbf{v}}$ is a vector of size mf containing the coefficients for all frames. With these transformations, Eq. (4) becomes a sparse linear system that can be solved for $\mathbf{w}$.

Note that the prior is a positional constraint imposed at every frame. Consequently, the prior implies a conflicting goal in a per-frame position domain formulation suggesting a different movement from the source as shown in Figure 8(a), which potentially causes unnatural artifacts as shown in Figure 7(d). In contrast, the prior and the velocity domain match work in harmony as the prior provides boundary values for the velocity domain formulation to respect. As a result, the movement of the source is reproduced faithfully in a plausible expression space of the target face as shown in Figures 7(f) and 8(b).

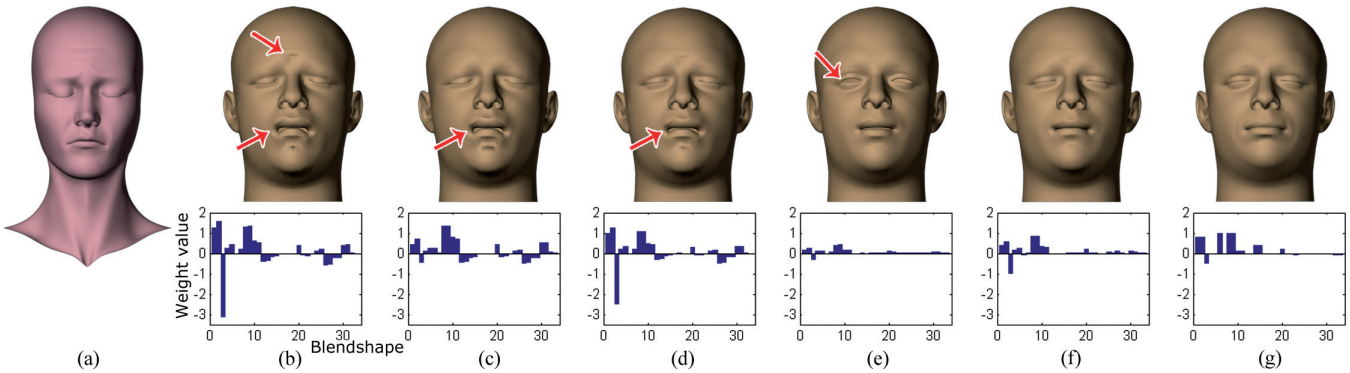


Fig. 7. Retargeted facial expressions are shown with their blendshape weight distributions displayed on the graphs below. (a) A source expression (b) retargeting with either per-frame or movement matching without weight constraints (c) per-frame retargeting with hard constraints $-0.5 \leq w_i \leq 1.5$ on the weights. (d) per-frame matching with prior $\alpha = 0.5$ (e) prior in the velocity domain with $\alpha = 0.5$ and (f) with $\alpha = 0.01$. Retargeting without constraints produces artifacts on the mouth and the forehead as the target face is contorted to follow the position of points that it cannot naturally reproduce. The restricted weight range or prior with per-frame matching does not prevent artifacts properly. A small α with movement matching produces a nice weight distribution and reflects the source well while prior with a large α overly restricts animation (the eyes are not closed). (g) An artist manually reproduced the source expression on the target face. Both the expression and the weight distribution are similar to those of (f).

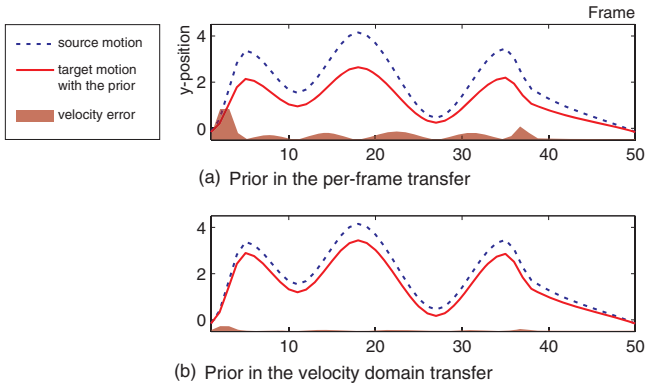


Fig. 8. Comparison of the effects when the prior is applied differently. While (a) the prior in the per-frame matching generates a different target movement from the source movement (see the curve variation), (b) the prior in the velocity domain reproduces the source movement faithfully with little velocity error. The same α values from Figure 7(d) and (f) are used here.

6. SPACETIME EDITING FOR RETARGETING

As mentioned in Havaldar [2006], the results of facial retargeting do not necessarily meet the requirements of the final animation even if the retargeting algorithms are correct and the process is applied as intended. They presented reasons for why this happens.

(a) The combination of artistically designed blendshapes cannot perfectly match the actual actor's motion.

(b) The proportions of CG face model and an actor's face can be significantly different.

(c) Motion capture marker placements differ from day to day.

(d) The desired performance is not what the actor performed. Either the required expression is not present in the motion capture data or it needs to be exaggerated.

In practice, an animator modifies the result by adjusting blendshape weights. Although the modifications can be interpolated, realistic animation requires placing keyframes as frequently as every three to five frames [Luamanuvae 2010].

Our framework excels in this respect by providing the powerful spacetime editing. The movement matching allows the influence of any user-specified pose modification to be smoothly propagated across neighboring frames while intelligently incorporating the source movement data. After the initial movement matching (Eq. (4)) is performed, the animator adjusts the target blendshape weights at any selected frame as desired. The animator's modifications work as additional boundary conditions for the Poisson problem. The animator can also specify a range of propagation, in which case the values at the start and end frames from the original solution (Eq. (4)) are also used as boundary conditions. Solving the system with the new boundary conditions applies the correction smoothly. By considering only the matrix elements that are within the user-specified range, the results of modification can be easily updated at real-time rates.

7. RESULTS

We performed a set of different blendshape-based retargeting tasks to verify the validity of our method. Both existing animation (Figure 15(a)) and motion capture (Figure 15(b)) were used as facial animation sources. Three models were employed as target faces with 33, 50, and 18 blendshape targets, respectively. The position of 54 correspondences were identified between the source and the target.

Figure 9 shows retargeting results created by per-frame retargeting methods and our method when constraints are imposed. "No Const", (b), represents either per-frame or movement matching with no constraints (i.e., $\min_{\mathbf{W}} \|\mathbf{BW} - \mathbf{S}\|_F^2$). "Range PF", (c), represents per-frame retargeting with the restricted weight range as in Bregler et al. [2002] (hard constraint). The weight range $-0.5 \leq w_i \leq 1.5$ is used for Range PF. "Prior PF", (d), incorporates the PCA prior from Section 5 with per-frame retargeting as in Blanz and Vetter [1999] (soft constraint). To make the PCA prior, we employed all the target blendshapes as training examples. The α for Prior PF and our method was determined to produce the same weight range used for Range PF ($\alpha = 0.5, \alpha = 0.01$, respectively). The results from our method (e) closely recreated the source motion and look plausible

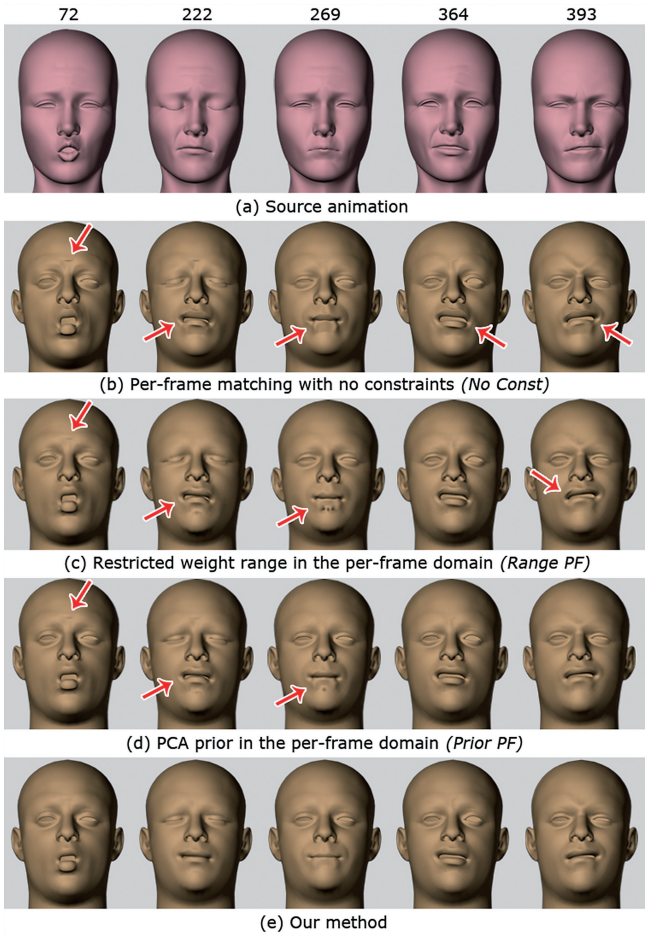


Fig. 9. Blendshape-based facial animation retargeting results by per-frame retargeting methods and our method when constraints are imposed. While constrained per-frame matching methods, (c) and (d), produce less distortion compared to (b), artifacts such as an improperly open mouth are not prevented. The red arrows indicate noticeable artifacts. Our method (e) produces natural expressions in accord with the prior. See the accompanying video for the entire animation sequence.

on the target face with less distortion than the results of the constrained per-frame matching methods. For instance, the per-frame matching results at frame 222 reveal an incompletely closed mouth and unpleasant distortions while our method effectively reproduces the source shape for the same frame. Figure 10 shows additional retargeting examples of source motion with extreme expressions. The retargeting results fail to recreate exactly the same expressions because of the absence of appropriate blendshapes. Nevertheless, our method effectively avoids the type of overfitting artifacts that appear in the results of per-frame transfers.

Figure 11 shows the amount of velocity error between the source and target correspondences (i.e., $\|\mathbf{D}(\mathbf{BW})^T - \mathbf{DS}^T\|_F^2$) compared to that of retargeting without constraints (No Const). While the per-frame matching methods (Range PF and Prior PF) produce large additional errors when constraints are imposed, our method incurs little additional errors, which leads to better reconstruction of the source movements. This test verifies the illustrations in Figures 3 and 8.

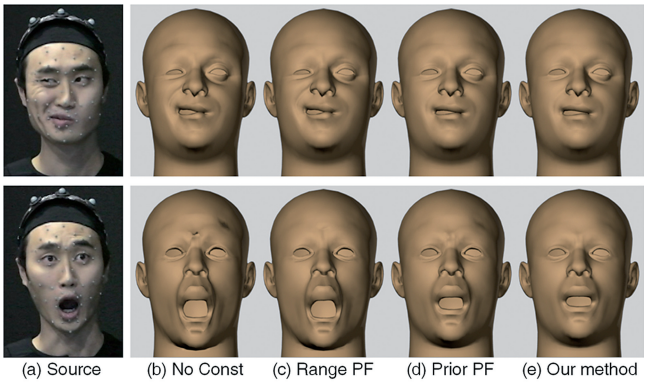


Fig. 10. Retargeting results of source motion with extreme expressions by different methods.

Most importantly, our method provides intelligent spacetime editing; the influences from user modifications are smoothly propagated. Figure 12 shows an example of a spacetime editing. Suppose a user wants to alter the shape of the lips and the left eye at frame 68 and limit the influence to the range from 63 to 74. Then, the weights at the three frames (corrected frame 68, frame 63, and frame 74) become new boundary conditions. By solving the linear system within this range with the new constraints, the desired animation can be efficiently obtained. See the accompanying video for the full performance and an interactive editing scenario.

We performed a test to investigate the effect of the model-specific prior when a different number of input expressions are used. In Figure 13, (a) shows retargeting results when all blendshape targets are used (same as other figures in the article). When only 20 mouth-related blendshapes are used in (b), the artifact around the forehead is not fixed as only the mouth region of the face is constrained. Similarly, when only 11 eye-related blendshapes are used in (c), only eye region is constrained in the results. The partial selection of blendshapes for the prior as shown in (b) and (c) can achieve a specific purpose in retargeting. (d) shows retargeting results when 10 additional highly expressive facial expressions are provided together with all the blendshape targets for the prior. The expression space of the prior model is expanded, but quality improvement was not noticeable in most fames. This makes sense because the expression space of the target face remains the same, regardless of the expanded expression space of the prior. This experiment supports our choice of providing blendshapes as input shapes for the prior.

The results from our method and by a nonblendshape-based approach (i.e., deformation transfer [Sumner and Popovic 2004]) are presented together for comparison in Figure 14. Our results are comparable to those from deformation transfer, although their underlying concepts are fundamentally different (and in fact are complementary to each other). Our approach both retains the editing advantages of the blendshape representation and supports the spacetime editing described earlier.

Along with facial expressions, the head motion of the source that was calculated in Section 3.1 is directly applied to the target face. Figure 15 shows retargeting results from various target models with their head motion applied. Note that the eye movements are manually keyframed since the eyes are generally not part of a blendshape model. The target face faithfully recreates the source animation.

The timing information for the test cases is as follows. All tests were carried out on a standard PC with an Intel Xeon 2.8 GHz and 8GByte of memory. RBF training and interpolation (Section 3.2) take milliseconds due to sparse correspondences. Movement

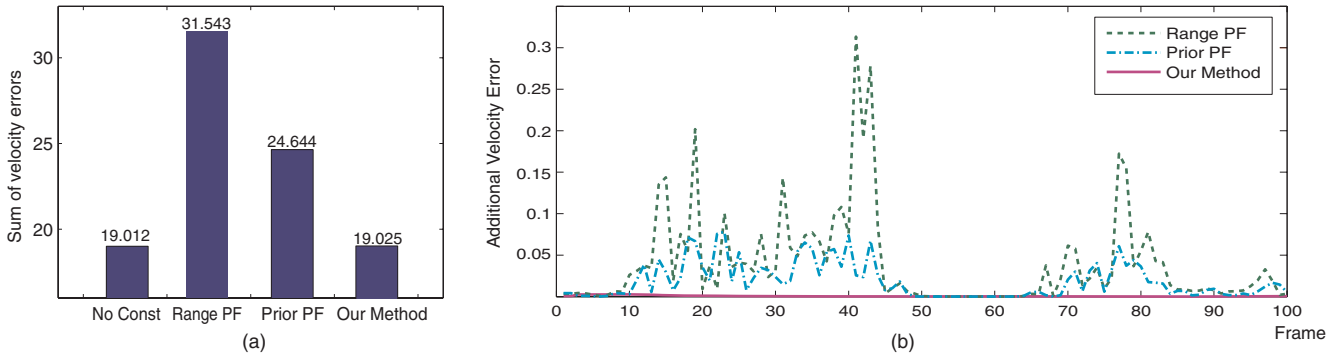


Fig. 11. (a) Sum of velocity errors (i.e., $\|\mathbf{D}(\mathbf{BW})^T - \mathbf{DS}^T\|_F^2$) of various methods. (b) Deltas of velocity errors from those of "No Const" along the time line. While retargeting schemes based on per-frame matching (the dotted lines) generate bigger velocity errors when constraints are imposed, our method produces little additional errors, which leads to better reconstruction of the source movements.

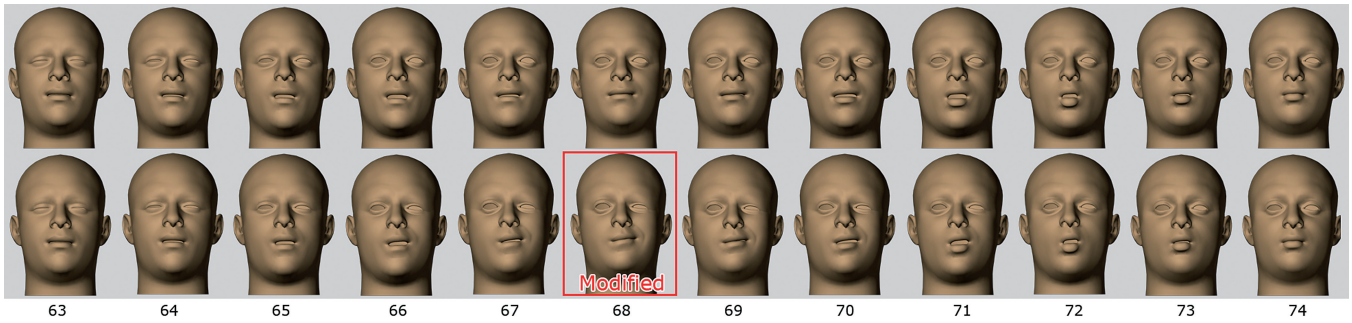


Fig. 12. Spacetime editing: the first row shows the initial retargeted animation and the second row shows the result after the modification. At frame 68, the expression is changed and its influence is propagated across surrounding frames. Recalculating the weight values took less than a second.

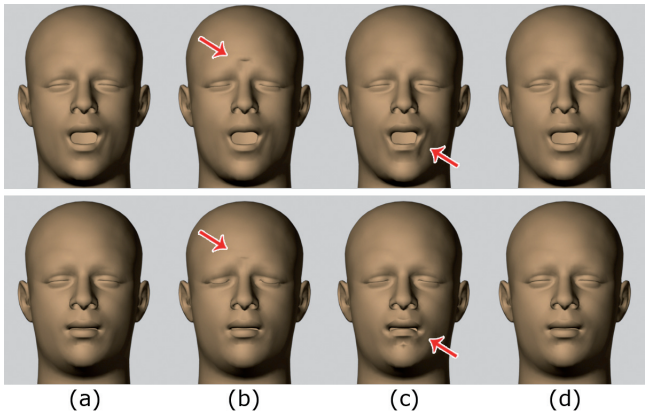


Fig. 13. Retargeting results with a varying number of input shapes for the model-specific prior. (a) all blendshapes, (b) 20 mouth-related blendshapes, (c) 11 eye-related blendshapes, and (d) additional 10 highly expressive expressions together with all blendshapes are used as input shapes of the prior.

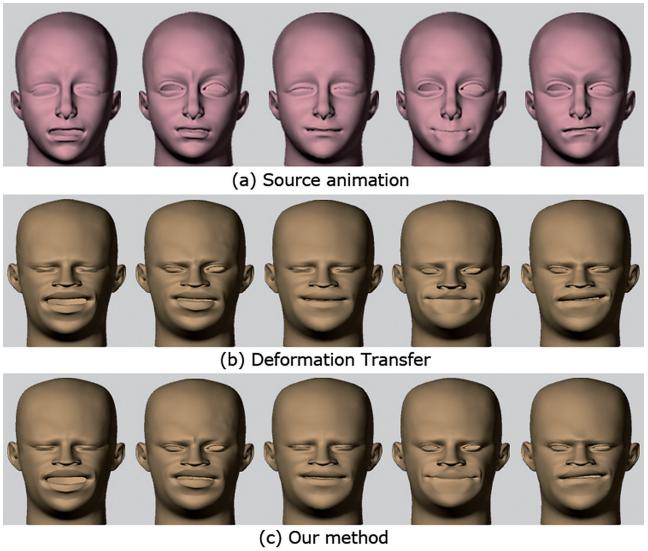
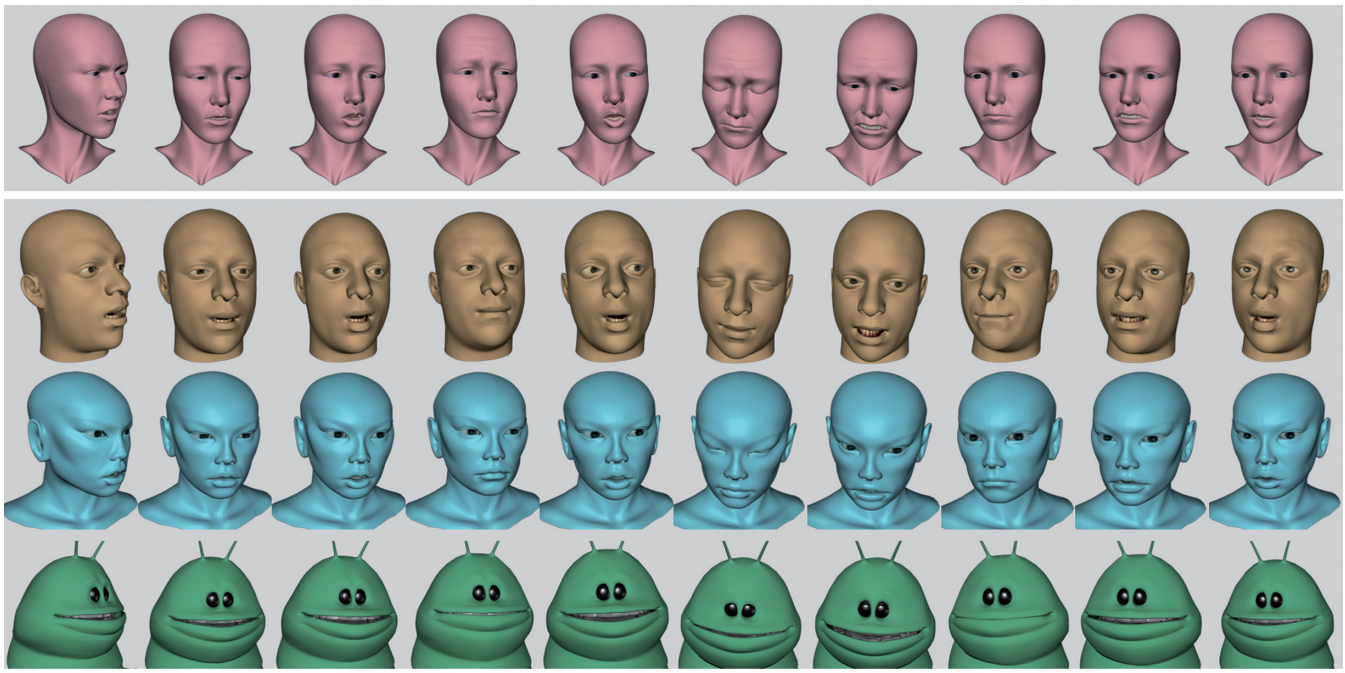
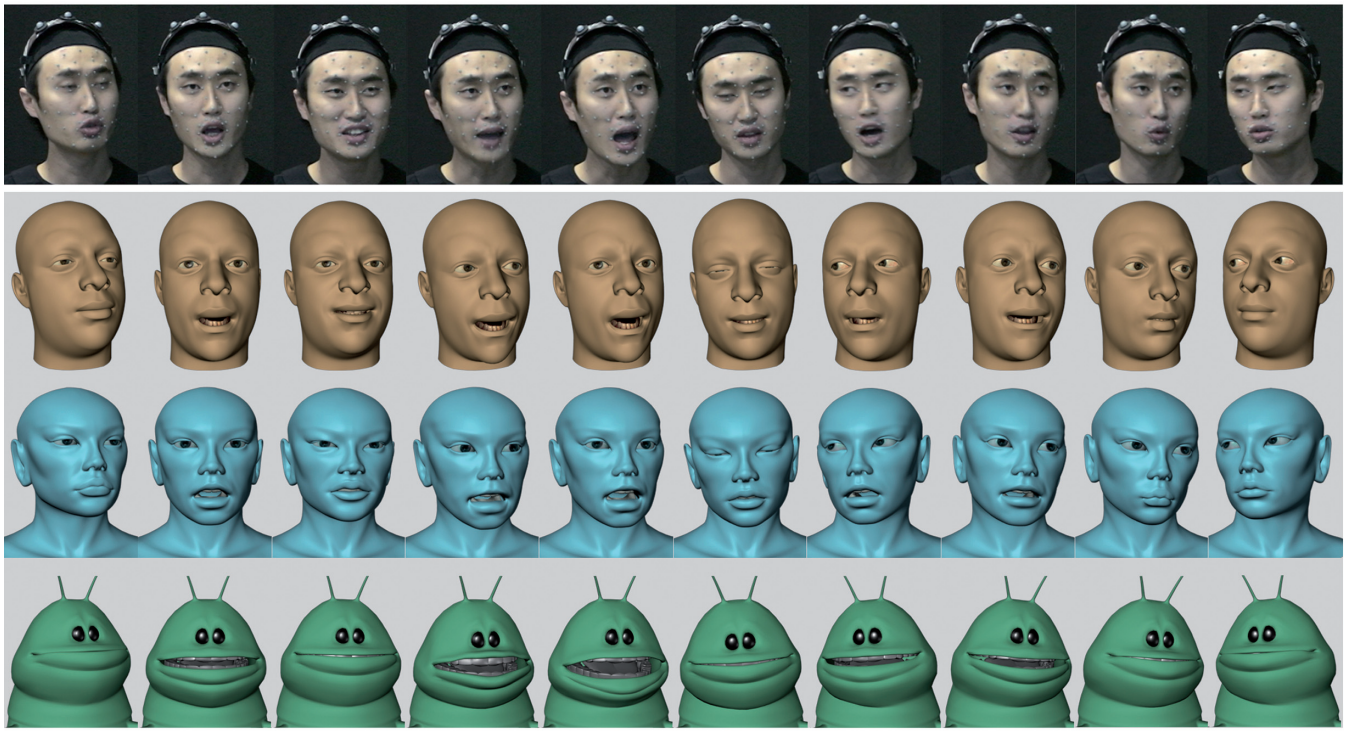


Fig. 14. Retargeting animation (a) by (b) deformation transfer and (c) our method. Our results are comparable to those of deformation transfer while retaining the editing advantages of blendshapes and spacetime editing.

matching with the prior creates a sparse linear system that is solved using Matlab. Table I presents the timing information corresponding to different lengths of animations. While the computation does take some time, retargeting is not required to be an interactive process, and the times involved are small compared to other tasks such as rendering. As described in Section 6, edits to small ranges of animation can be done at real-time rates (and a short range of animation is all that an animator can manually adjust in any case). Also



(a)



(b)

Fig. 15. Retargeted animations with head motion. (a) mesh to mesh retargeting (b) motion capture to mesh retargeting. The first row shows the source animation retargeted to different target models on the second, the third, and the fourth rows. Despite the large difference in proportion and the small number of blendshapes ($m = 18$) used for the bug model, the retargeted results show reasonably good reconstruction of the source motion. The prior weight $\alpha = 0.01$ is used for these examples.

Table I. Timing for Different Sequences

# frames	10	50	100	500	1000	1500
Timing(sec.)	0.22	1.32	2.76	15.21	31.65	47.38

54 markers and 50 blendshapes were used.

note that sequences of longer than several hundred frames rarely arise in practice, since most movie shots are only 3–5 seconds in length.

7.1 Discussion

Although retargeting was successful with sparse marker correspondences in our test cases, very complex face models may require dense correspondences. Dense correspondences can be obtained by applying cylindrical projection [Noh and Neumann 2001] or the enhanced iterative closest point algorithms in Sumner and Popovic [2004] and Li et al. [2008]. Applying parameterization schemes such as Levy et al. [2002] or Kraevoy et al. [2003] is also a viable possibility. A large-size linear system will result from dense correspondences. We note that multigrid schemes have successfully accelerated other Poisson problems and it may be possible to devise a multigrid scheme for this problem. Also, however, it is not clear if dense correspondences are truly necessary to produce realistic facial animation; in fact current industry practice indicates otherwise, since most major performance animated films have used sparse motion capture markers (*The Matrix* sequels are an exception [Borshukov et al. 2003]).

Although movement matching reproduces the movement of the source faithfully, it is not the only consideration in practical animation retargeting. Ensuring that the mouth or eyelids are fully closed at specific frames is a standard requirement. In this case, minimizing a Sobolev norm involving the position as well as the velocity can be a valid alternative (i.e., $\min_{\mathbf{W}} \|\mathbf{D}(\mathbf{BW})^T - \mathbf{DS}^T\|_F^2 + \beta\|\mathbf{BW} - \mathbf{S}\|_F^2$). A weighted Sobolev blending that only incorporates specific vertices at specific frames can provide more direct control over the results. For example, by applying higher weights β around the mouth region than the other regions on the face, the least-squares method will concentrate on the retargeting around the mouth region. With a simple user interface that allows easy specification of the vertices, more specific and direct control may be possible.

Our formulation utilizes whole face blendshapes. Although our method produces faithful retargeting in general, large differences in proportion between a source and a target model or using a limited number of blendshape may bring difficulty in reproducing the source motion correctly. An obvious alternative would be a weight transfer scheme using localized blending approach such as Joshi et al. [2003] or faceIK [Zhang et al. 2004]. Due to the local blend weight matching, a smaller set of blendshape may produce comparable results. In order to produce a retargeted animation, Eq. (4) has to be solved for every segment or region with correspondences.

8. CONCLUSIONS AND FUTURE WORK

An increasing number of recent movies are founded on transferring facial motion from a human actor to a computer graphic model. This article presents a novel solution to this problem based on the obvious principle of movement matching together with a model-specific prior. The movement matching principle leads directly to a velocity domain formulation (Poisson boundary value problem) with boundary conditions imposed at the start and/or end of the facial performance as well as at any frames where pose modifications are specified by the user. By introducing a simple prior on the plausible space of target face poses, we formulate the motion transfer in a MAP framework and solve for the entire motion simultaneously.

In this approach, the potential artifacts of solving for every frame independently are avoided and temporal coherence is guaranteed. While movement matching would be identical to per-frame matching if the source and target shapes are perfectly calibrated so that additional constraints are not required, this is rarely or never the case. When the target face cannot exactly match the source shape, it is arguably better to match movement in a least-squares sense constrained to the expression space of the target than to strive to match per-frame displacement and fail in a time-varying manner.

Most importantly, spacetime editing is naturally incorporated in our system. The artist uses ordinary blendshape editing on the facial pose, but instead of applying the modification with simple spline interpolation, the modification is intelligently propagated while respecting the original captured movement. In this respect, our approach resembles the powerful gradient domain editing approaches that have been introduced in other areas (e.g., Pérez et al. [2003]), although our mathematical formulation is necessarily different due to the need to drive standard blendshape models.

In this work, we concentrated on large-scale facial deformation only. The importance of dynamic small-scale detail such as wrinkles for realistic facial animation has recently been recognized [Bickel et al. 2007; Ma et al. 2008; Beeler et al. 2011]. Developing retargeting techniques that take full advantage of such small-scale data is our future research direction in order to take the quality of facial animation up a notch. Finally, it would be interesting to apply our framework to various animation retargeting tasks other than facial retargeting.

APPENDIX: Movement Matching Minimization

In Section 4, $\mathbf{D}$ is a derivative matrix which is applied to a vector of values over time. This subtracts the previous value from the current one, that is, it is the backwards first finite difference.

$$\mathbf{D} = \begin{bmatrix} -1 & 1 & & & \\ & -1 & 1 & & \\ & & -1 & 1 & \\ & & & \cdots & \\ & & & -1 & 1 \end{bmatrix}$$

Because several rows of $\mathbf{W}^T$ are known from the boundary constraints, $\mathbf{D}(\mathbf{BW})^T - \mathbf{DS}^T$ in Eq. (2) can be reordered so that the known elements of $\mathbf{W}^T$ appear in the bottom of the matrix. This can be written as a block matrix system

$$\begin{bmatrix} \mathbf{D}_1 & \mathbf{D}_3 \\ \mathbf{D}_2 & \mathbf{D}_4 \end{bmatrix} \begin{bmatrix} \mathbf{W}_1^T \\ \mathbf{W}_2^T \end{bmatrix} \mathbf{B}^T - \mathbf{DS}^T,$$

where $\mathbf{W}_2^T$ contains the known *boundary* rows of $\mathbf{W}^T$. Separating the known parts gives

$$\begin{bmatrix} \mathbf{D}_1\mathbf{W}_1^T + \mathbf{D}_3\mathbf{W}_2^T \\ \mathbf{D}_2\mathbf{W}_1^T + \mathbf{D}_4\mathbf{W}_2^T \end{bmatrix} \mathbf{B}^T - \mathbf{DS}^T$$

$$= \begin{bmatrix} \mathbf{D}_1 \\ \mathbf{D}_2 \end{bmatrix} \mathbf{W}_1^T\mathbf{B}^T - \left(\mathbf{DS}^T - \begin{bmatrix} \mathbf{D}_3 \\ \mathbf{D}_4 \end{bmatrix} \mathbf{W}_2^T\mathbf{B}^T\right).$$

We can write

$$\mathbf{M} = \begin{bmatrix} \mathbf{D}_1 \\ \mathbf{D}_2 \end{bmatrix}, \quad \mathbf{X}^T = \mathbf{W}_1^T$$

$$\mathbf{R} = \left(\mathbf{DS}^T - \begin{bmatrix} \mathbf{D}_3 \\ \mathbf{D}_4 \end{bmatrix} \mathbf{W}_2^T\mathbf{B}^T\right)$$

then Eq. (2) becomes

$$\min_{\mathbf{X}} \|\mathbf{M}(\mathbf{BX})^T - \mathbf{R}\|_F^2$$

which is equivalent to

$$\min_{\mathbf{X}} \mathrm{tr}(\mathbf{M}(\mathbf{BX})^T - \mathbf{R})^T(\mathbf{M}(\mathbf{BX})^T - \mathbf{R}).$$

Note that $\|\mathbf{K}\|_F^2 = \mathrm{tr}(\mathbf{K}^T\mathbf{K})$. Expanding this equation gives

$$\min_{\mathbf{X}} \mathrm{tr}(\mathbf{BXLX}^T\mathbf{B}^T - \mathbf{BXM}^T\mathbf{R} - \mathbf{R}^T\mathbf{MX}^T\mathbf{B}^T + \mathbf{R}^T\mathbf{R}),$$

where $\mathbf{L} = \mathbf{M}^T\mathbf{M}$ is a discrete invertible Laplacian matrix. Taking the derivative with respect to $\mathbf{X}$ gives

$$\mathbf{LX}^T\mathbf{B}^T\mathbf{B} = \mathbf{M}^T\mathbf{RB}$$
$$\mathbf{LX}^T = \mathbf{M}^T\mathbf{RB}(\mathbf{B}^T\mathbf{B})^{-1}$$

which is a linear system that can be solved for $\mathbf{X}$.

ACKNOWLEDGMENTS

We are grateful to Jaewon Song for editing capture data and Joe Letteri, Sebastian Sylwan, and Dejan Momcilovic for discussions and help. We also acknowledge the use of face models created by Jason Osipa, Hiroki Itokazu, and Kyle Labad.

REFERENCES

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*, 75:1–75:10.

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, 33.

BLANZ, V. AND VETTER, T. 1999. A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive techniques*. ACM Press/Addison-Wesley Publishing Co., New York, 187–194.

BORSHUKOV, G., PIPONI, D., LARSEN, O., LEWIS, J., AND TEMPELAAR-LIETZ, C. 2003. Universal capture: Image-based facial animation for "The Matrix Reloaded." In *ACM SIGGRAPH 2003 Sketches & Applications*. ACM, New York, 1–1.

BREGLER, C., LOEB, L., CHUANG, E., AND DESHPANDE, H. 2002. Turning to the masters: Motion capturing cartoons. *ACM Trans. Graph. 21,* 3, 399–407.

CAO, Y., FALOUTSOS, P., AND PIGHIN, F. 2003. Unsupervised learning for speech motion editing. In *Proceedings of the ACM SIGGRAPH/ Eurographics Symposium on Computer Animation*. Eurographics Association, 225–231.

CHOE, B., LEE, H., AND SEOK KO, H. 2001. Performance-driven muscle-based facial animation. *J. Vis. Comput. Anim. 12*, 67–79.

CHUANG, E. AND BREGLER, C. 2002. Performance driven facial animation using blendshape interpolation. Tech. rep., Department of Computer Science, Stanford University.

CHUANG, E. AND BREGLER, C. 2005. Mood swings: Expressive speech animation. *ACM Trans. Graph. 24,* 2, 331–347.

DENG, Z., CHIANG, P.-Y., FOX, P., AND NEUMANN, U. 2006. Animating blendshape faces by cross-mapping motion capture data. In *Proceedings of the Symposium on Interactive 3D Graphics and Games*. ACM, New York, 43–48.

DENG, Z. AND NOH, J. 2007. *Computer Facial Animation: A Survey*. Springer, London.

EKMAN, P. AND FRIESEN, W. 1977. *Manual for the Facial Action Coding System*. Consulting Psychologists Press, Palo Alto, CA.

HAVALDAR, P. 2006. Performance driven facial animation. In *ACM SIGGRAPH '06 Course #30 Notes*.

HERTZMANN, A. 2004. Introduction to Bayesian learning. In *ACM SIGGRAPH'04 Course Notes*. ACM, New York, 22.

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

KOVAR, L. AND GLEICHER, M. 2003. Flexible automatic motion blending with registration curves. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. Eurographics Association, 214–224.

KRAEVOY, V., SHEFFER, A., AND GOTSMAN, C. 2003. Matchmaker: constructing constrained texture maps. *ACM SIGGRAPH'03 Papers*. ACM, 326–333.

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

LEE, J., CHAI, J., REITSMA, P. S. A., HODGINS, J. K., AND POLLARD, N. S. 2002. Interactive control of avatars animated with human motion data. *ACM Trans. Graph. 21,* 3, 491–500.

LEVY, B., PETITJEAN, S., RAY, N., AND MAILLO T, J. 2002. Least squares conformal maps for automatic texture atlas generation. In *ACM SIGGRAPH Conference Proceedings*.

LEWIS, J. AND ANJYO, K. 2010. Direct manipulation blendshapes. *IEEE Comput. Graph. Appl. 30,* 4, 42–50.

LI, H., SUMNER, R. W., AND PAULY, M. 2008. Global correspondence optimization for non-rigid registration of depth scans. *Comput. Graph. Forum 27,* 5.

LI, H., WEISE, T., AND PAULY, M. 2010. Example-based facial rigging. *ACM Trans. Graph. 29,* 4, 1–6.

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

LUAMANUVAE, J. 2010. Personal communication (Weta Digital).

MA, W.-C., JONES, A., CHIANG, J.-Y., HAWKINS, T., FREDERIKSEN, S., PEERS, P., VUKOVIC, M., OUHYOUNG, M., AND DEBEVEC, P. 2008. Facial performance synthesis using deformation-driven polynomial displacement maps. *ACM Trans. Graph. 27,* 5, 1–10.

MA, X., LE, B. H., AND DENG, Z. 2009. Style learning and transferring for facial animation editing. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. ACM, New York, 123–132.

NOH, J. AND NEUMANN, U. 2001. Expression cloning. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*. ACM. 277–288.

ORVALHO, V. C., ZACUR, E., AND SUSIN, A. 2008. Transferring the rig and animations from a character to different face models. *Comput. Graph. Forum 27,* 8, 1997–2012.

PARKE, F. I. 1972. Computer generated animation of faces. In *Proceedings of the ACM Annual Conference*. ACM, New York, 451–457.

PARKE, F. I. AND WATERS, K. 1996. *Computer Facial Animation*. A. K. Peters.

PÉREZ, P., GANGNET, M., AND BLAKE, A. 2003. Poisson image editing. *ACM Trans. Graph. 22,* 3, 313–318.

PIGHIN, F., SZELISKI, R., AND SALESIN, D. H. 2002. Modeling and animating realistic faces from images. *Int. J. Comput. Vis. 50,* 2, 143–169.

PYUN, H., KIM, Y., CHAE, W., KANG, H. W., AND SHIN, S. Y. 2003. An example-based approach for facial expression cloning. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. Eurographics Association, 167–176.

REVERET, L. AND ESSA, I. 2001. Visual coding and tracking of speech related facial motion. Tech. rep., IEEE CVPR International Workshop on Cues in Communication.

SAGAR, M. AND GROSSMAN, R. 2006. Facial performance capture and expressive translation for King Kong. In *SIGGRAPH Sketches*.

SUMNER, R. W. AND POPOVIC, J. 2004. Deformation transfer for triangle meshes. *ACM SIGGRAPH Papers*. ACM. 399–405.

SUMNER, R. W., ZWICKER, M., GOTSMAN, C., AND POPOVIĆ, J. 2005. Mesh-based inverse kinematics. In *ACM SIGGRAPH Papers*. ACM, New York, 488–495.

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

WAHBA, G. 1990. *Spline Models for Observational Data*. SIAM.

WEISE, T., LI, H., VAN GOOL, L., AND PAULY, M. 2009. Face/off: Live facial puppetry. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. ACM, New York, 7–16.

WILLIAMS, L. 1990. Performance-driven facial animation. In *Proceedings of the 17th Annual Conference on Computer Graphics and Interactive Techniques*. ACM, New York, 235–242.

ZHANG, L., SNAVELY, N., CURLESS, B., AND SEITZ, S. M. 2004. Spacetime faces: High resolution capture for modeling and animation. *ACM SIGGRAPH Papers*. ACM, New York, 548–558.

Received September 2010; revised August 2011; accepted October 2011