Semantically-aware Blendshape Rigs from Facial Performance Measurements

Wan-Chun Ma * Mathieu Lamarre Etienne Danvoye Chongyang Ma Manny Ko
Javier von der Pahlen Cyrus A. Wilson

Activision Publishing, Inc.

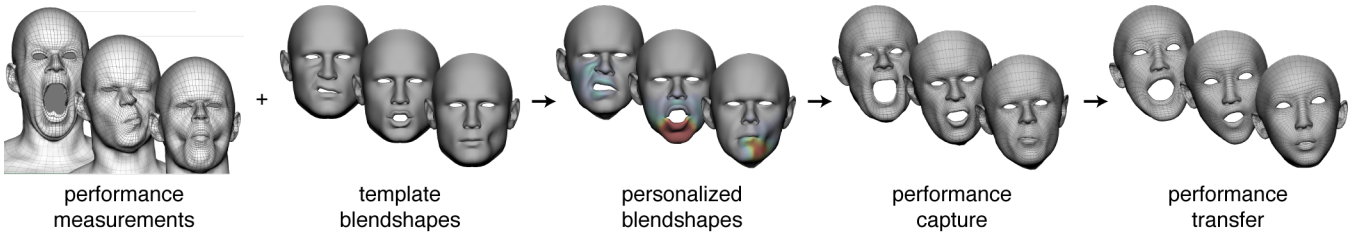


Figure 1: *An artist-created blendshape model is adopted to* 63 *corresponded facial performance measurements by non-rigidly deforming* 35 *basis shapes. The overall reconstruction error drops around* 50% *after refinement. The personalized blendshape model preserves the semantics of the artist-created model such that rigs personalized to different actors behave consistently.*

Abstract

We present a framework for automatically generating personalized blendshapes from actor performance measurements, while preserving the semantics of a template facial animation rig. Firstly, we capture various poses from the subject with our photogrammetry apparatus. The 3D reconstruction from each pose is then corresponded by an image-based tracking algorithm. The core of our framework is an optimization algorithm which iteratively refines the initial estimation of the blendshapes such that they can fit the performance measurements better. This framework facilitates creation of an ensemble of realistic digital-double face rigs for each individual with consistent behavior across the character set.

Keywords: performance capture, facial animation, retargeting, blendshapes, blind signal separation

Concepts: •**Computing methodologies** → **Motion capture; Classification and regression trees;**

1 Introduction

There are several challenges associated with trying to obtain the basis expressions from captured poses of an actor's face. First, rigid head movement must be factored out to yield face motion in a common coordinate system for basis selection or animation curve solving. Second, there is the question of what constitutes an appropriate "neutral" pose. On the acquisition side, this should be a relaxed facial expression which the actor can hold, as well as reproduce consistently across scanning sessions and/or modes of capture. However on the animation side, this is not necessarily the best pose to represent the rest state of the animation rig, or the origin of the shape space. Crucially, several of the desired poses may be difficult for some or all individuals to perform. Furthermore, even easy-to-perform expressions are impractical to achieve in isolation: a captured "jawOpen" expression might contain 5% of "upperBrowRaiser". In most current industry pipelines, the captured shapes have to be carefully processed by manually painting out any undesirable motion, in order to produce "clean" basis shapes. Significant production time is spent decomposing retopologized face shapes into localized, meaningful poses. The quality of the results will depend highly on the skill of the artist. We propose a numerical optimization process which addresses the above challenges in producing digital double facial animation rigs. The goal of our technique is to produce a set of blendshapes capable of reproducing input performances captured for a given actor, while conforming to general semantics such that the rigs produced for each actor behave consistently in the hands of an animator. The key idea of our approach is therefore to use a set of blendshapes, expertly created by professional artists, as a "gold standard" to constrain the solution.

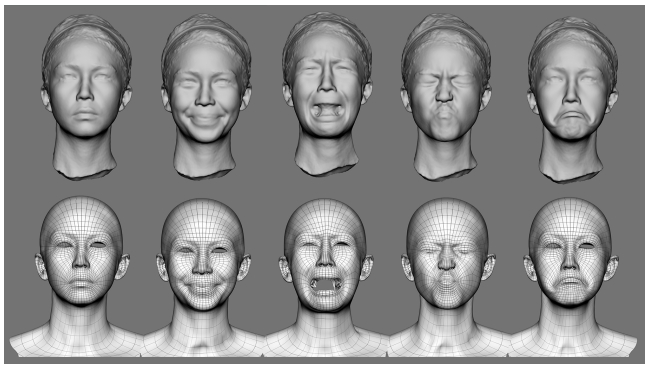


Figure 2: *A subset of the* 80 *shapes we obtained for this subject with the method described in Section 2.* ***Left column:*** *neutral pose.* ***Top row:*** *stereo 3D reconstruction.* ***Bottom row:*** *corresponded meshes.*

2 Facial Capture and Correspondence

Our blendshape personalization (described in Section 3) requires performance measurements of the individual as input towards which the blendshapes will be optimized. These measurements can be discrete poses, or frames of a performance. They do not need to represent individual blendshapes or action units. However, the

*e-mail:wanchun.ma@gmail.com

SA '16 Technical Briefs ,, December 05-08, 2016, , Macao
ISBN: 978-1-4503-4541-5/16/12
DOI: http://dx.doi.org/10.1145/3005358.3005378

measurements do need to be corresponded to each other. Our capture system consists of an array of 16 monochrome global-shutter machine vision cameras (Ximea xiQ), 4M pixels each, synchronized at 75 Hz and mounted in a lighting apparatus similar to [Wenger et al. 2005]. To sample the full space of movement of the actor's face, the actor performs a series of poses, returning to a neutral rest pose in between each. The duration of each single pose capture is around 2 seconds. A typical range-of-motion (ROM) capture session contains about 60 video sequences. We compute the camera parameters using commercial photogrammetry software.

2.1 Feature Matching

To establish the correspondence, we first obtain a neutral mesh, with a topology appropriate for visual production use. This "tracking" mesh is first manually registered to the neutral scan. However, methods such as morphable models or non-rigid registration could make this process automatic. Any further registration tasks will be performed automatically. We then track the transition from neutral to each pose by deforming this tracking mesh. We developed an optimization to track the motion by directly solving for the mesh at each frame using Laplacian deformation with barycentric coordinates, guided by three different levels of image-based correspondences in a coarse-to-fine fashion:

- **Face Features.** An open-source face detector and facial features predictor [Kazemi and Sullivan 2014] provide a robust way to initialize the correspondence search at the coarsest level. One of the most popular face feature model [Sagonas et al. 2015] has 68 landmarks of which 51 can be used as surface matches.
- **Sparse Robust Image Features.** Features like SIFT, SURF, or FAST are good choices for intermediate matches. They require tuning multiple parameters: search region, descriptor distance threshold, peak threshold and edge threshold. The peak and edge threshold are selected to obtain more than 10k features per view. In this stage, we iterate image feature matching and mesh registration multiple times to gradually increase the number of matches by simultaneously reducing the search region and increasing the descriptor distance threshold and the regularization coefficient.
- **Dense Optical Flow Matches.** At the finest level, we use optical flow to establish dense correspondence. Our optical flow algorithm, which is based on Beeler et al. [2011], computes inter-frame motion from the images of each camera. It proceeds from coarse to fine using a Gaussian image pyramid. The local 2D search is done by using normalized cross correlation (NCC) with bicubic filtering for subpixel matching. The filtering leverages the reconstructed 3D surface to avoid smoothing across discontinuous boundaries (e.g. occlusion).

The 2D feature matching is initialized by the estimated motion from a coarser level of detail or a previous iteration. We use the registered mesh at the current iteration to render a dense motion field using hardware accelerated rasterization. This motion field will be employed as a guide to reduce search regions for the next level.

2.2 Registration

Given a tracking mesh that is registered to the source frame, feature matches from each view, calibrated cameras and 3D surface reconstruction surfaces, we begin to estimate the registration for the target frame. The method we now describe is used both for registering a single global neutral mesh to the first frame of every sequences and for propagating the mesh from one frame to the next. We start by inverting the perspective projection to lift all 2D feature matches to 3D world space using per view depth maps, which are rendered by the GPU. Since our tracking mesh has far fewer vertices compared to the dense matches, we use barycentric coordinates with respect to the tracking mesh to represent the locations of the matches. We represent world space matches as normal displacements following Kobbelt et al. [1999]. For each matched point pair $(\mathbf{u}_i, \mathbf{u}_j)$, we find a base surface point $\mathbf{b}_i \in \mathcal{S}$ contained in triangle $\Delta(\mathbf{a}, \mathbf{b}, \mathbf{c})$ such that $\mathbf{b}_i$ is the root of $(\mathbf{u}_i - \mathbf{b}_i) \times \mathbf{n}_i = 0$, where

$$\mathbf{b}_i = \alpha\mathbf{a} + \beta\mathbf{b} + \gamma\mathbf{c}, \mathbf{n}_i = \frac{\alpha\mathbf{n_a} + \beta\mathbf{n_b} + \gamma\mathbf{n_c}}{\|\alpha\mathbf{n_a} + \beta\mathbf{n_b} + \gamma\mathbf{n_c}\|},$$

$$\text{subject to } \alpha + \beta + \gamma = 1,$$

$$\mathbf{u}_i = \mathbf{b}_i + h_i \cdot \mathbf{n}_i, h_i \in \mathbf{R}.$$

$\mathbf{n}_i$ is the interpolated normal vector at $\mathbf{b}_i$. The barycentric constraints $\mathbf{c}_j$ on the target mesh are computed by making the interpolated normal and normal displacements equal in the source and target. By minimizing this specific algebraic distance instead of the geometric distance we obtain a least squares solution.

$$\mathbf{c}_j = \mathbf{u}_j - h_i \cdot \mathbf{n}_i$$

To solve the vertex positions of the tracking mesh $\mathbf{p}$, we use Laplacian deformation [Botsch and Sorkine 2008], minimizing the following energy:

$$\mathbf{E}(\mathbf{p}) = \|\mathbf{Lp} - \delta\|^2 + \lambda \|\mathbf{MBp} - \mathbf{M}\boldsymbol{c}\|^2 . \quad (1)$$

$\mathbf{L}$ is the Laplacian matrix. The barycentric equation matrix $\mathbf{B}$ is built using one row per world space match with only three non-zero entries for (α, β, γ). The weight matrix $\mathbf{M}$ contains the per-match quality score.

3 Blendshape Personalization

At the core of our framework is the blendshape optimization process, where we would like to personalize a set of template blendshapes such that they can faithfully reproduce performance measurements of an actor. We formulate this task as the following optimization problem:

$$\min_{\mathbf{w}_i, \mathbf{R}_i, \mathbf{t}_i, \mathbf{D}, \mathbf{b}_0} \sum_{i=1}^{n_f} \mathbf{E}_g^i, \quad (2)$$

where

$$\mathbf{E}_g^i = \|\mathbf{M}_i'(\mathbf{x}_i - \mathbf{p}_i)\|^2,$$

$$\mathbf{x}_i = (\mathbf{I}_{n_v} \otimes \mathbf{R}_i)(\mathbf{D}\mathbf{w}_i + \mathbf{b}_0) + (\mathbf{1}_{n_v} \otimes \mathbf{t}_i).$$

$\mathbf{x}_i$ is the reconstructed face pose based on the blendshape model (pose offsets $\mathbf{D}$, the neutral pose $\mathbf{b}_0$, and blendshape weights $\mathbf{w}_i$) and estimated rigid motion (rotation $\mathbf{R}_i$ and translation $\mathbf{t}_i$) at the ith frame; and $\mathbf{p}_i$ is the input tracked facial performance. $\mathbf{M}_i' = \mathbf{M}_i \otimes \mathbf{I}_3$, where $\mathbf{M}_i$ is the weight matrix described in Equation 1 where each diagonal element stores the matching quality score of each vertex. In the following text, we will always assume that the confidence matrix is pre-multiplied to the performance and the blendshape model. $\mathbf{I}_{n_v}$ is an identity matrix with size of number of vertices n_v; and $\mathbf{1}_{n_v}$ is a column vector of ones with the length of n_v. Based on Equation 2, we would like to find the optimal solution for all the variables: head rigid motion and blending weights at each frame, and the optimal blendshapes and neutral pose that can best explain the input performance. Nevertheless, the problem is extremely ill-posed, which is impossible to solve directly. To produce our preferred solution, we need to introduce additional regularizations. This optimization process is similar to that of Li et al. [2010], except that their method does not solve for rigid motion, as their technique primarily fits to manually modeled poses.

First, an approximate neutral pose of the subject is manually selected from the tracked performance. Note that we do not assume a "perfect" neutral pose, as our algorithm will optimize for it as well. We then create an initial blendshape model using deformation transfer [Sumner and Popović 2004]. Each synthesized expression $\mathbf{b}_i$ for the subject is generated by applying the deformation gradients from a manually created source template character pose to $\mathbf{b}_0$. The result becomes a generic linear deformable model $\mathbf{D}^*$ for the subject, where each column of $\mathbf{D}^*$ equals to $\mathbf{b}_i$ - $\mathbf{b}_0$. This model serves as an initial guess to bootstrap the following optimization process.

3.1 Weights and Rigid Motion Update

For each frame, we compute the weights $\mathbf{w}_i$ which "pose" the current blendshape model $\mathbf{D}$ (initially $\mathbf{D} = \mathbf{D}^*$) to best reproduce the input corresponded performance measurement $\mathbf{P}$. We cast this problem as a simplified version of Equation 2 where $\mathbf{D}$ and $\mathbf{b}_0$ are fixed:

$$\min_{\mathbf{w}_i, \mathbf{R}_i, \mathbf{t}_i} \mathbf{E}_g^i. \tag{3}$$

The weights $\mathbf{w}_i$ are constrained between 0.0 and 1.0. Minimizing Equation 3 therefore calls for a constrained nonlinear optimization algorithm, as the problem is nonlinear due to the rotation variables. An alternative way to solve Equation 3 efficiently is to use a local/global method [Sorkine and Alexa 2007; Weise et al. 2011] that solves rigid transformation and other parameters separately. The optimal rigid transformation can be solved with singular value decomposition when $\mathbf{w}_i$ is fixed. On the other hand, the weights can be solved with constrained quadratic programming when $\mathbf{R}$ and $\mathbf{t}$ are fixed. We found the algorithm usually converges quickly and produces correct head motions. To provide meaningful blendshape weights and to avoid overfitting, we introduce two regularization terms into Equation 3:

$$\min_{\mathbf{w}_i, \mathbf{R}_i, \mathbf{t}_i} \mathbf{E}_g^i + \lambda_s \mathbf{E}_s^i + \lambda_t \mathbf{E}_t. \tag{4}$$

Sparseness. Much as an animator will focus on the major rig controls necessary to produce a desired performance, we wish our posing computation to favor using as few blendshapes as possible to fit the input measurement. We introduce the following energy to promote sparsity in the computed blendshape weights. The sparseness term is the square of the L1 norm: $\mathbf{E}_s^i = \|\mathbf{w}_i\|_1^2$. This is different than the traditional Lasso regularization, where $\|\mathbf{w}\|_1$ is used. Taking advantage that the weight $\mathbf{w}_i$ is non-negative, $\|\mathbf{w}_i\|_1$ becomes $\mathbf{1}_{n_b}^\top \mathbf{w}_i$, where n_b denotes the number of poses. (Note that n_b does not count for the neutral pose as it does not belong to $\mathbf{D}$ in our formulation.) Minimizing $\|\mathbf{w}_i\|_1$ is equivalent to minimizing $\|\mathbf{w}_i\|_1^2 = \|\mathbf{1}_{n_b}^\top \mathbf{w}_i\|^2$. Therefore, the L1 regularization problem is turned into a constrained least squares problem, which can be easily solved with any quadratic programming solver. We set λ_s to be 0.1 in all the examples.

Temporal Smoothness. If the input performance is captured contiguously, the blendshape weights from the previous frame can serve as a strong prior as we solve for the next frame. The temporal smoothness term penalizes the difference between the current weight and that at the previous frame: $\mathbf{E}_t^i = \|\mathbf{w}_i - \mathbf{w}_{i-1}\|^2$.

3.2 Blendshape Update

Similar to Li et al. [2010], the blendshapes and weights are alternatively optimized in an expectation–maximization (EM) fashion. In this step, we fix the weights and the rigid transformation. To solve for a new set of blendshapes $\mathbf{D}$, the optimization then becomes:

$$\min_{\mathbf{D}} \mathbf{E}_{\tilde{g}}, \tag{5}$$

$$\mathbf{E}_{\tilde{g}} = \|(\mathbf{W}^\top \otimes \mathbf{I}_{n_b})\mathrm{vec}(\mathbf{D}) - \mathrm{vec}(\tilde{\mathbf{P}})\|^2. \tag{6}$$

The blendshape weights can be viewed as the relative contribution of each shape to the reconstructed pose, therefore they also specify how the residuals should be redistributed back to the blendshapes. This strategy follows the approach of Hyneman et al. [2005].

Deformation Regularization. To ensure that the personalized blendshapes retain the semantics of the template shapes, we introduce a regularization term $\mathbf{E}_r$ based on deformation gradients:

$$\min_{\mathbf{D}} \mathbf{E}_{\hat{g}} + \lambda_r \mathbf{E}_r + \lambda_d \mathbf{E}_d, \tag{7}$$

$$\mathbf{E}_r = \|\mathbf{G}'\mathrm{vec}(\mathbf{D}) + \mathbf{G}'(\mathbf{1}_{n_b}^\top \otimes \mathbf{b}_0) - \mathbf{g}^*\|^2,$$
$$\mathbf{E}_d = \|\mathrm{vec}(\mathbf{D}) - \mathrm{vec}(\mathbf{D}^*)\|^2.$$

$\mathbf{D}^*$ is the initial blendshape model, $\mathbf{G}' = \mathbf{I}_{n_b} \otimes (\mathbf{G} \otimes \mathbf{I}_3)$, where $\mathbf{G}$ is the deformation gradient operator matrix [Sumner and Popović 2004]. $\mathbf{g}^*$ is the stacked vector of all deformation gradients from the initial blendshapes. This term constrains the blendshapes to have similar deformation with respect to the initial blendshapes. However, we found that relying solely on deformation gradient regularization does not provide desirable results, as a regularization in the differential space does not consider the direction of motion in each blendshape, which is one of the important ingredients of the shape semantic. Therefore, we add the regularization term $\mathbf{E}_d$ which indicates that the offset $\mathbf{D}$ should be similar to $\mathbf{D}^*$; in other words the offsets should have similar directions. This additional term greatly helps the stability of the optimization. We also prefer to keep the localized property of $\mathbf{D}^*$: if a vertex is static in a particular template shape, the same vertex in the personalized shape remains static. Similar to Li et al. [2010], we enforce $\mathbf{D}_{ki} = 0$ as a boundary condition if $\mathbf{D}_{ki}^* = 0$ (the subscript indicates row and column indices). By optimizing in this way, we limit the region that each new shape is allowed to vary, and we maintain the deformation semantics of each expression. This step is key because it eliminates the need for subsequent spatial segmentation of shapes. The scalars λ_r and λ_d control how much the resulting shapes are allowed to deform relative to the template shapes. We use $\lambda_r = 0.1$ and $\lambda_d = 0.05$ in all of our results.

Neutral Vertex Positions. Thus far we have discussed optimization of blendshape vertex offsets, while keeping the neutral mesh constant. However, neutral mesh vertex positions can be optimized as well, by replacing $\mathbf{W}$, $\mathbf{D}^*$, and $\mathbf{g}^*$ with:

$$\hat{\mathbf{W}} = \begin{pmatrix} \mathbf{1}_{n_f} \\ \mathbf{W} \end{pmatrix}, \hat{\mathbf{D}}^* = \begin{pmatrix} \mathbf{0}_{(3n_v)} \mid \mathbf{D}^* \end{pmatrix}, \hat{\mathbf{g}^*} = \begin{pmatrix} \mathbf{1}_{n_t} \otimes \mathbf{I}_3 \\ \mathbf{g}^* \end{pmatrix}.$$

$\mathbf{1}_{n_t} \otimes \mathbf{I}_3$ is introduced as the deformation gradients of the neutral undeformed mesh, where n_t is the number of triangles. The first column of the solved $\mathbf{D}$ is the update for the neutral shape. We found that neutral shape estimation is very useful in dealing with consistent bias in the fitting residuals. Any constant offset shared across all optimized blendshapes should be absorbed by the neutral shape rather than pollute the blendshape offsets.

4 Results

We use the QP solver in CVXOPT and the sparse linear solver based on LU decomposition in SciPy to solve Equations 4 and 7, respectively. For Equation 7, there are about 86k variables (where $\mathbf{D}_{ki}^* \neq 0$), and it takes averagely 30 seconds to solve on a personal workstation with an Intel Xeon 3.0GHz CPU. The refined blendshapes are shown in Figure 3. The semantics are well preserved

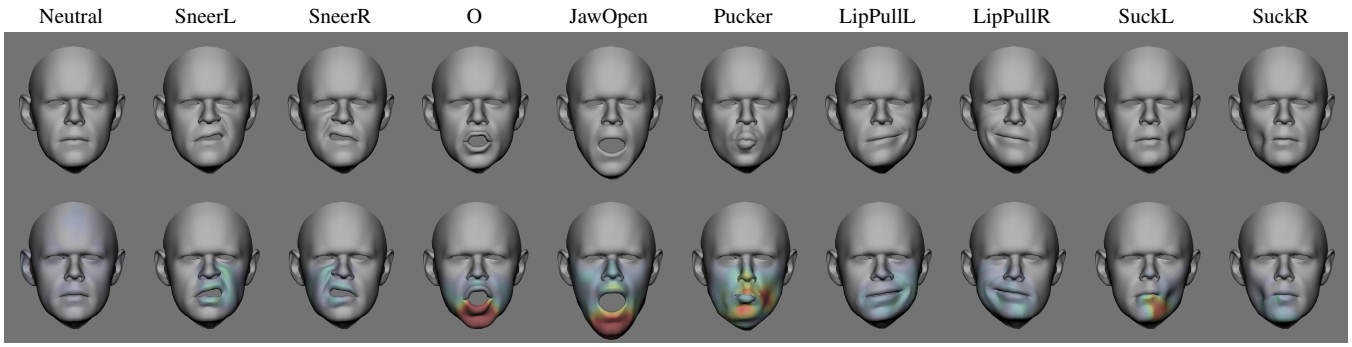
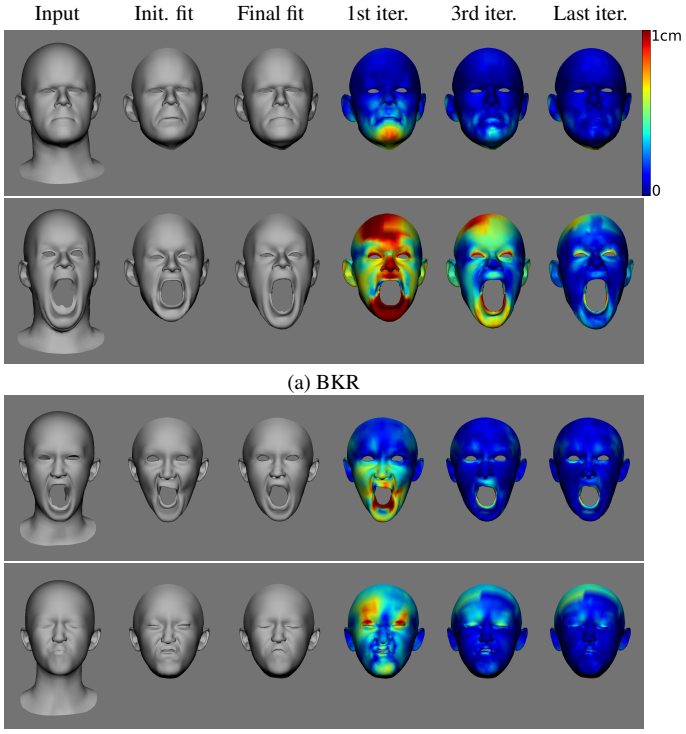


Figure 3: *Selected blendshapes personalized to subject BKR. The figure shows the initial estimates (top) and refined blendshapes (bottom). The color indicates the magnitude of the change.*



(a) BKR

(b) SOT

Figure 4: *Fitting improvements from two subjects BKR and SOT. Each row shows the blendshape fitting result to a selected captured pose at initial and final iterations. Shape differences are visualized (color plots) at the first, third, and final iterations.*

from the template blendshapes thanks to the use of deformation regularization as well as hard constraints on the static vertices, an important feature for blendshape rigs. In Figure 4 we show fitting improvements of several selected measurements from two subjects. Note that some regions exhibit residual error even in the final result. For regions which were not corresponded correctly in the input performance measurements, such as around the inner mouth loop, a high error in the final fit is expected and desired; we do not wish to learn from spurious input. Our algorithm is also capable of processing long sequences of dense geometry reconstruction, we introduce a tracking technique based on non-rigid registration, which is similar to Weise et al. [2009]. For motion retargeting, since all result characters have semantically matched blendshapes, blendshape weights are simply copied between source and target characters.

Acknowledgements

We thank Mike Sanders, Alex Smith, Bernardo Antoniazzi, Jennifer Velazquez, Neil Yang, Andy Hendrickson, and the Activision Capture Studio for their support. We also thank the reviewers for their constructive comments.

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*, 4, 75:1–75:10.

BOTSCH, M., AND SORKINE, O. 2008. On linear variational surface deformation methods. *IEEE Trans. on Visualization and Computer Graphics 14*, 1, 213–230.

HYNEMAN, W., ITOKAZU, H., WILLIAMS, L., AND ZHAO, X. 2005. Human face project. In *ACM SIGGRAPH 2005 Courses*.

KAZEMI, V., AND SULLIVAN, J. 2014. One millisecond face alignment with an ensemble of regression trees. In *Proc. CVPR*, 1867–1874.

KOBBELT, L., VORSATZ, J., AND SEIDEL, H.-P. 1999. Multiresolution hierarchies on unstructured triangle meshes. *Computational Geometry 14*, 1, 5–24.

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

SAGONAS, C., ANTONAKOS, E., TZIMIROPOULOS, G., ZAFEIRIOU, S., AND PANTIC, M. 2015. 300 faces in-the-wild challenge: database and results. *Image and Vision Computing*.

SORKINE, O., AND ALEXA, M. 2007. As-rigid-as-possible surface modeling. In *Proc. SCA*, SGP '07, 109–116.

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

WEISE, T., LI, H., VAN GOOL, L., AND PAULY, M. 2009. Face/off: Live facial puppetry. In *Proc. SCA*, 7–16.

WEISE, T., BOUAZIZ, S., LI, H., AND PAULY, M. 2011. Realtime performance-based facial animation. *ACM Trans. Graph. 30*, 4, 77:1–77:10.

WENGER, A., GARDNER, A., TCHOU, C., UNGER, J., HAWKINS, T., AND DEBEVEC, P. 2005. Performance relighting and reflectance transformation with time-multiplexed illumination. *ACM Trans. Graph. 24*, 3, 756–764.