Realtime Performance-Based Facial Animation

Thibaut Weise Sofien Bouaziz Hao Li Mark Pauly
EPFL

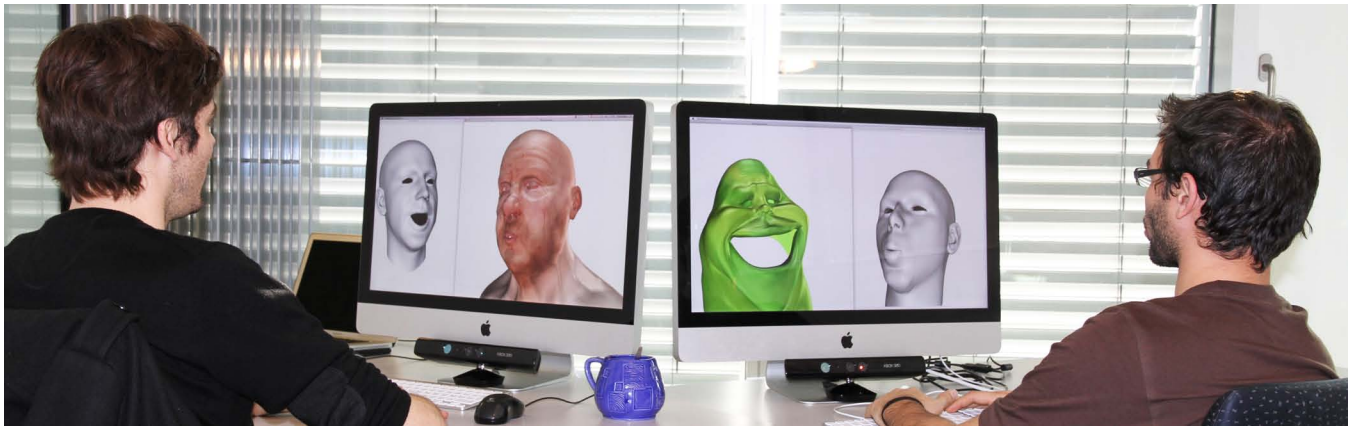


Figure 1: *Our system captures and tracks the facial expression dynamics of the users (grey renderings) in realtime and maps them to a digital character (colored renderings) on the opposite screen to enable engaging virtual encounters in cyberspace.*

Abstract

This paper presents a system for performance-based character animation that enables any user to control the facial expressions of a digital avatar in realtime. The user is recorded in a natural environment using a non-intrusive, commercially available 3D sensor. The simplicity of this acquisition device comes at the cost of high noise levels in the acquired data. To effectively map low-quality 2D images and 3D depth maps to realistic facial expressions, we introduce a novel face tracking algorithm that combines geometry and texture registration with pre-recorded animation priors in a single optimization. Formulated as a maximum a posteriori estimation in a reduced parameter space, our method implicitly exploits temporal coherence to stabilize the tracking. We demonstrate that compelling 3D facial dynamics can be reconstructed in realtime without the use of face markers, intrusive lighting, or complex scanning hardware. This makes our system easy to deploy and facilitates a range of new applications, e.g. in digital gameplay or social interactions.

CR Categories: I.3.6 [Computer Graphics]: Methodology and Techniques—Interaction techniques; I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation;

Keywords: markerless performance capture, face animation, realtime tracking, blendshape animation

Links: DL PDF

1 Introduction

Capturing and processing human geometry, appearance, and motion is at the core of modern computer animation. Digital actors are often created through a combination of 3D scanning, appearance acquisition, and motion capture, leading to stunning results in recent feature films. However, these methods typically require complex acquisition systems and substantial manual post-processing. As a result, creating high-quality character animation entails long turn-around times and substantial production costs. Recent developments in gaming technology, such as the Nintendo Wii and the Kinect system of Microsoft, focus on robust motion tracking for compelling realtime interaction, while geometric accuracy and appearance are of secondary importance. Our goal is to leverage these technological advances and create a low-cost facial animation system that allows arbitrary users to enact a digital character with a high level of realism.

We emphasize *usability*, *performance*, and *robustness*. Usability in our context means ease of deployment and non-intrusive acquisition. These requirements put severe restrictions on the acquisition system which in turn leads to tradeoffs in the data quality and thus higher demands on the robustness of the computations. We show that even a minimal acquisition system such as the Kinect can enable compelling realtime facial animations. Any user can operate our system after recording a few standard expressions that are used to adapt a facial expression model.

Contributions. Our main contribution is a novel face tracking algorithm that combines 3D geometry and 2D texture registration in a systematic way with dynamic blendshape priors generated from existing face animation sequences. Formulated as a probabilistic optimization problem, our method successfully tracks complex facial expressions even for very noisy inputs. This is achieved by mapping the acquired depth maps and images of the performing user into the space of realistic facial expressions defined by the animation prior. Realtime processing is facilitated by a reduced facial expression model that can be easily adapted to the specific expres-

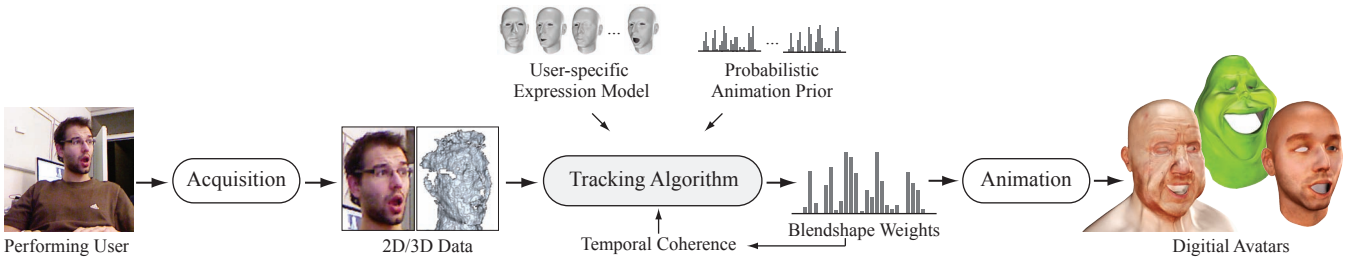


Figure 2: *Overview of the online processing pipeline. The blendshape weights that drive the digital avatar are estimated by matching a user-specific expression model to the acquired 2D image and 3D depth map. A probabilistic animation prior learned from existing blendshape sequences regularizes the tracking. Temporal coherence is exploited by considering a window of consecutive frames.*

sion space and facial geometry of different users. We integrate these components into a complete framework for realtime, non-intrusive, markerless facial performance capture and animation (Figure 1).

1.1 Related Work

Facial performance capture and performance-driven animation have been active research areas in recent years, with a plethora of different acquisition systems and processing pipelines that share many fundamental principles as well as specific implementation details. Performance-based facial animation typically consists of a *non-rigid tracking* stage (often with a parametric template model) followed by an *expression retargeting* procedure. A full review of these systems is beyond the scope of this paper and we refer to [Pighin and Lewis 2006] for a more detailed discussion.

One fundamental tradeoff in all of these systems is the relation between the quality of the acquired data and the complexity of the acquisition setup. On one end of the spectrum are systems designed for greatest possible accuracy that lead to stunning virtual avatars suitable for movie production. Because of their robustness, marker-based techniques [Williams 1990; Guenter et al. 1993; Lin and Ouhyoung 2005] are widely used for realtime facial animation and generally deliver sufficient motion parameters for convincing retargeting of non-human creatures or simple game characters.

For the realistic digitization of human faces, markerless approaches such as realtime 3D scanners are usually more advantageous due to their ability to capture fine-scale dynamics (e.g. wrinkles and folds). All these methods involve highly specialized sensors and/or controlled studio environments [Zhang and Huang 2004; Borshukov et al. 2005; Ma et al. 2007; Beeler et al. 2010; Bradley et al. 2010]. High-resolution facial motion is generally recovered through variants of non-rigid registration and tracking algorithms across sequences of input geometry, texture, or both [Zhang et al. 2004; Furukawa and Ponce 2009; Alexander et al. 2009; Li et al. 2009; Weise et al. 2009; Bradley et al. 2010; Wilson et al. 2010]. With a focus on precision, these systems are not designed to achieve interactive performance in general environments, a crucial requirement for the type of consumer-level applications targeted by our work. The method of Weise et al. [2009] achieves realtime performance using a customized PCA tracking model, which requires an additional animation retargeting step based on deformation transfer to animate different characters. As their structured-light scanner generates high quality depth maps, online tracking can be limited to geometry registration only.

On the other end of the tradeoff between data quality and hardware complexity are passive, single camera systems that have been a focus of research in computer vision. Most commonly, 2D parametric shape models have been used for non-rigid tracking [Li et al. 1993; Black and Yacoob 1995; Essa et al. 1996; DeCarlo and Metaxas 1996; Pighin et al. 1999]. However, due to the additional challenges posed by uncontrolled lighting environments and unreliable textures, tracking is usually limited to facial features such as eyes, eyebrows, pupils, or inner and outer contours of the lips. Established methods such as active appearance models [Cootes et al. 2001] or Eigen-Points [Covell 1996] employ a probabilistic prior model built from large sets of training data to achieve realtime performance while preventing drifts. As demonstrated in Chuang and Bregler [2002], these parametric models can be used to reliably synthesize simple facial expressions for virtual avatars but inherently lack in facial details. Chai and colleagues [Chai et al. 2003] first extract 2D animation controls using feature tracking and then map these controls to 3D facial expressions using a preprocessed motion capture database to reduce tracking artifacts.

Our goal is to maintain the flexibility, ease of deployment, and non-intrusive acquisition of passive, single camera acquisition, but push the quality of the reconstruction towards that achieved with complex, special-purpose hardware setups. For this purpose we follow the established strategy of using existing animation data for regularization. However, instead of performing a separate post-filtering step as in most previous work, e.g. [Lou and Chai 2010], we integrate an animation prior directly into the tracking optimization using a maximum a posteriori estimation. Our animation prior is based on Mixtures of Probabilistic Principal Component Analyzers (MPPCA) [Tipping and Bishop 1999b], similar in spirit to [Lau et al. 2007] who use a static pose prior for interactive design of facial geometry. In comparison to Gaussian Processes that have been successfully employed as pose prior, e.g. [Grochow et al. 2004] and [Ikemoto et al. 2009], MPPCA scales well with the size of the data set, making it particularly suitable for real-time applications.

1.2 Overview

Performance-driven facial animation requires solving two main technical challenges: We need to accurately track the rigid and non-rigid motion of the user's face, and map the extracted tracking parameters to suitable animation controls that drive the virtual character. Our approach combines these two problems into a single optimization that solves for the most likely parameters of a user-specific expression model given the observed 2D and 3D data. We derive a suitable probabilistic prior for this optimization from pre-recorded animation sequences that define the space of realistic facial expressions. Figure 2 gives an overview of our pipeline.

Blendshape Representation. To integrate tracking and animation into one optimization, we represent facial expressions as a weighted sum of blendshape meshes. This design choice offers a number of advantages: A blendshape model provides a compact representation of the facial expression space, thus significantly reducing the dimensionality of the optimization problem. In addition,

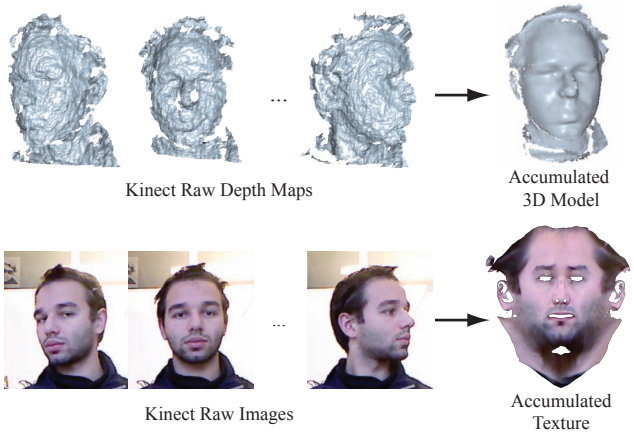


Figure 3: *Acquisition of user expressions for offline model building. Aggregating multiple scans under slight head rotation reduces noise and fills in missing data.*

we can use existing blendshape animations, that are ubiquitous in movie and game production, to define the dynamic expression priors. The underlying hypothesis here is that the blendshape weights of a human facial animation sequence provide a sufficient level of abstraction to enable expression transfer between different characters. Finally, the output generated by our algorithm, a temporal sequence of blendshape weights, can be directly imported into commercial animation tools, thus facilitating integration into existing production workflows.

Acquisition Hardware. All input data is acquired using the Kinect system, i.e. no other hardware such as laser scanners is required for user-specific model building. The Kinect supports simultaneous capture of a 2D color image and a 3D depth map at 30 frames per second, based on invisible infrared projection (Figure 4). Essential benefits of this low-cost acquisition device include ease of deployment and sustained operability in a natural environment. The user is neither required to wear any physical markers or specialized makeup, nor is the performance adversely affected by intrusive light projections or clumsy hardware contraptions. However, these key advantages come at the price of a substantial degradation in data quality compared to state-of-the-art performance capture systems based on markers and/or active lighting. Ensuring robust processing given the low resolution and high noise levels of the input data is the primary challenge that we address in this paper.

2 Facial Expression Model

A central component of our tracking algorithm is a facial expression model that provides a low-dimensional representation of the user's expression space. We build this model in an offline prepro-

Figure 4: *The Kinect simultaneously captures a 640 × 400 color image and corresponding depth map at 30 Hertz, computed via triangulation of an infrared projector and camera.*

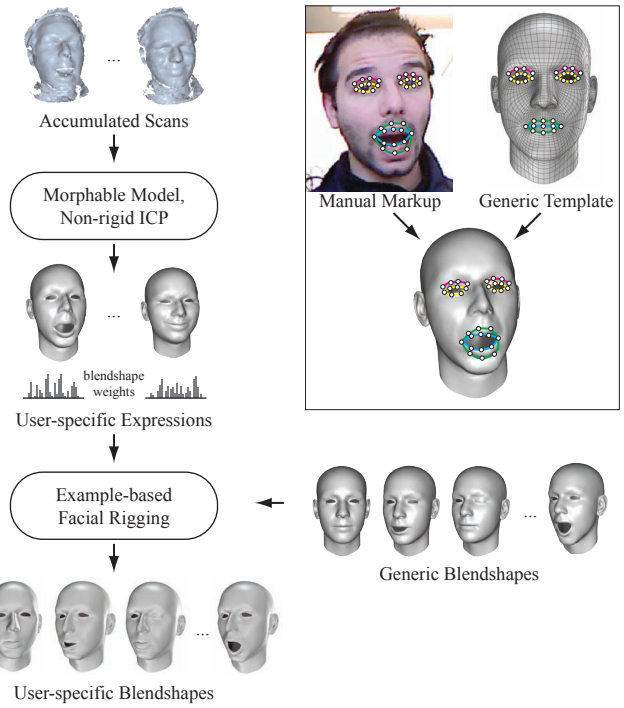


Figure 5: *Offline pre-processing for building the user-specific expression model. Pre-defined example poses of the user with known blendshape weights are scanned and registered to a template mesh to yield a set of user-specific expressions. An optimization solves for the user-specific blendshapes that maintain the semantics of a generic blendshape model. The inset shows how manually selected feature correspondences guide the reconstruction of user-specific expressions.*

cessing step by adapting a generic blendshape model with a small set of expressions performed by the user. These expressions are captured with the Kinect prior to online tracking and reconstructed using a morphable model combined with non-rigid alignment methods. Figure 5 summarizes the different steps of our algorithm for building the facial expression model. We omit a detailed description of previous methods that are integrated into our algorithm. Please refer to the cited papers for parameter settings and implementation details.

Data Capture. To customize the generic blendshape rig, we record a pre-defined sequence of example expressions performed by the user. Since single depth maps acquired with the Kinect exhibit high noise levels, we aggregate multiple scans over time using the method described in [Weise et al. 2008] (see Figure 3). The user is asked to perform a slight head rotation while keeping the expression fixed (see accompanying video). Besides exposing the entire face to the scanner, this rotational motion has the additional benefit of alleviating reconstruction bias introduced by the spatially fixed infrared dot pattern projected by the Kinect. We use the method of [Viola and Jones 2001] to detect the face in the first frame of the acquisition and accumulate the acquired color images to obtain the skin texture using Poisson reconstruction [Pérez et al. 2003].

Expression Reconstruction. We use the morphable model of Blanz and Vetter [1999] to represent the variations of different human faces in neutral expression. This linear PCA model is first registered towards the recorded neutral pose to obtain a high-quality

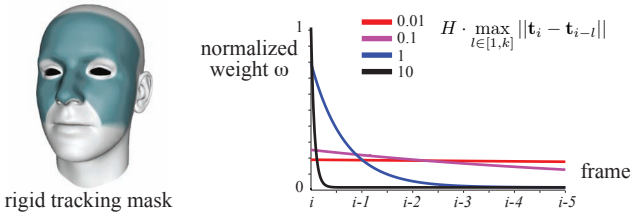


Figure 6: *The colored region on the left indicates the portion of the face used for rigid tracking. The graph on the right illustrates how temporal filtering adapts to the speed of motion.*

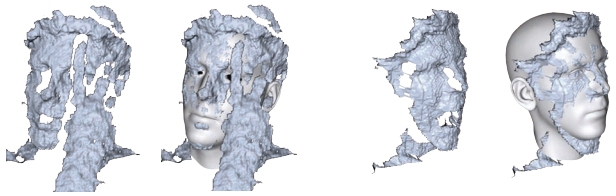


Figure 7: *Robustly tracking the rigid motion of the face is crucial for expression reconstruction. Even with large occlusions and fast motion, we can reliably track the user's global pose.*

template mesh that roughly matches the geometry of the user's face. We then warp this template to each of the recorded expressions using the non-rigid registration approach of [Li et al. 2009]. To improve registration accuracy, we incorporate additional texture constraints in the mouth and eye regions. For this purpose, we manually mark features as illustrated in Figure 5. The integration of these constraints is straightforward and easily extends the framework of [Li et al. 2009] with positional constraints.

Blendshape Reconstruction. We represent the dynamics of facial expressions using a generic blendshape rig based on Ekman's Facial Action Coding System (FACS) [1978]. To generate the full set of blendshapes of the user we employ example-based facial rigging as proposed by Li et al. [2010]. This method takes as input a generic blendshape model, the reconstructed example expressions, and approximate blendshape weights that specify the appropriate linear combination of blendshapes for each expression. Since the user is asked to perform a fixed set of expressions, these weights are manually determined once and kept constant for all users. Given this data, example-based facial rigging performs a gradient-space optimization to reconstruct the set of user-specific blendshapes that best reproduce the example expressions (Figure 5). We use the same generic blendshape model with $m = 39$ blendshapes in all our examples.

3 Realtime Tracking

The user-specific blendshape model defines a compact parameter space suitable for realtime tracking. We decouple the rigid from the non-rigid motion and directly estimate the rigid transform of the user's face before performing the optimization of blendshape weights. We found that this decoupling not only simplifies the formulation of the optimization, but also leads to improved robustness of the tracking.

Rigid Tracking. We align the reconstructed mesh of the previous frame with the acquired depth map of the current frame using ICP with point-plane constraints. To stabilize the alignment we use a pre-segmented template (Figure 6, left) that excludes the chin region from the registration as this part of the face typically exhibits the strongest deformations. As illustrated in Figure 7 this results in robust tracking even for large occlusions and extreme facial expressions. We also incorporate a temporal filter to account for the high-frequency flickering of the Kinect depth maps. The filter is based on a sliding window that dynamically adapts the smoothing coefficients in the spirit of the exponentially weighted moving average method [Roberts 1959] to reduce high frequency noise while avoiding disturbing temporal lags. We independently filter the translation vector and quaternion representation of the rotation. For a translation or quaternion vector $\mathbf{t}_i$ at the current time frame i, we compute the smoothed vector as weighted average in a window of size k as

$$\mathbf{t}_i^\star = \frac{\sum_{j=0}^{k} w_j \mathbf{t}_{i-j}}{\sum_{j=0}^{k} w_j} \tag{1}$$

where $\mathbf{t}_{i-j}$ denotes the vector at frame $i-j$. The weights w_j are defined as

$$w_j = e^{-j \cdot H \cdot \max_{l \in [1,k]} ||\mathbf{t}_i - \mathbf{t}_{i-l}||}, \tag{2}$$

with a constant H that we empirically determine independently for rotation and translation based on the noise level of a static pose. We use a window size of $k = 5$ for all our experiments.

Scaling the time scale with the maximum variation in the temporal window ensures that less averaging occurs for fast motion, while high-frequency jitter is effectively removed from the estimated rigid pose (Figure 6, right). As shown in the video, this leads to a stable reconstruction when the user is perfectly still, while fast and jerky motion can still be recovered accurately.

Non-rigid Tracking. Given the rigid pose, we now need to estimate the blendshape weights that capture the dynamics of the facial expression of the recorded user. Our goal is to reproduce the user's performance as closely as possible, while ensuring that the reconstructed animation lies in the space of realistic human facial expressions. Since blendshape parameters are agnostic to realism and can easily produce nonsensical shapes, parameter fitting using geometry and texture constraints alone will typically not produce satisfactory results, in particular if the input data is corrupted by noise (see Figure 8). Since human visual interpretation of facial imagery is highly sophisticated, even small tracking errors can quickly lead to visually disturbing artifacts.

3.1 Statistical Model

We prevent unrealistic face poses by regularizing the blendshape weights with a dynamic expression prior computed from a set of existing blendshape animations $\mathcal{A} = \{A_1, \ldots, A_l\}$. Each animation A_j is a sequence of blendshape weight vectors $\mathbf{a}_j^i \in \mathbb{R}^m$ that sample a continuous path in the m-dimensional blendshape space. We exploit temporal coherence of these paths by considering a window of n consecutive frames, yielding an effective prior for both the geometry and the motion of the tracked user.

MAP Estimation. Let $D_i = (G_i, I_i)$ be the input data at the current frame i consisting of a depth map G_i and a color image I_i. We want to infer from D_i the most probable blendshape weights $\mathbf{x}_i \in \mathbb{R}^m$ for the current frame given the sequence $X_n^i = \mathbf{x}_{i-1}, \ldots, \mathbf{x}_{i-n}$ of the n previously reconstructed blendshape vectors. Dropping the index i for notational brevity we formulate this inference problem as a maximum a posteriori (MAP) estimation

$$\mathbf{x}^\star = \arg\max_{\mathbf{x}} p(\mathbf{x}|D, X_n), \tag{3}$$

where $p(\cdot|\cdot)$ denotes the conditional probability. Using Bayes' rule we obtain

$$\mathbf{x}^\star = \arg\max_{\mathbf{x}} p(D|\mathbf{x}, X_n)p(\mathbf{x}, X_n). \quad (4)$$

Assuming that D is conditionally independent of X_n given $\mathbf{x}$, we can write

$$\mathbf{x}^\star \approx \arg\max_{\mathbf{x}} \underbrace{p(D|\mathbf{x})}_{\text{likelihood}} \underbrace{p(\mathbf{x}, X_n)}_{\text{prior}}. \quad (5)$$

Prior Distribution. To adequately capture the nonlinear structure of the dynamic expression space while still enabling realtime performance, we represent the prior term $p(\mathbf{x}, X_n)$ as a Mixtures of Probabilistic Principal Component Analyzers (MPPCA) [Tipping and Bishop 1999b]. Probabilistic principal component analysis (PPCA) (see [Tipping and Bishop 1999a]) defines the probability density function of some observed data $\mathbf{x} \in \mathbb{R}^s$ by assuming that $\mathbf{x}$ is a linear function of a latent variable $\mathbf{z} \in \mathbb{R}^t$ with $s > t$, i.e.,

$$\mathbf{x} = C\mathbf{z} + \mu + \epsilon, \quad (6)$$

where $\mathbf{z} \sim \mathcal{N}(0, I)$ is distributed according to a unit Gaussian, $C \in \mathbb{R}^{s \times t}$ is the matrix of principal components, μ is the mean vector, and $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$ is a Gaussian-distributed noise variable. The probability density of $\mathbf{x}$ can then be written as

$$p(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\mu, CC^T + \sigma^2 I). \quad (7)$$

Using this formulation, we define the prior in Equation 5 as a weighted combination of K Gaussians

$$p(\mathbf{x}, X_n) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}, X_n|\mu_k, C_k C_k^T + \sigma_k^2 I). \quad (8)$$

with weights π_k. This representation can be interpreted as a reduced-dimension Gaussian mixture model that attempts to model the high-dimensional animation data with locally linear manifolds modeled with PPCA.

Learning the Prior. The unknown parameters in Equation 8 are the means μ_k, the covariance matrixes $C_k C_k^T$, the noise parameters σ_k, and the relative weights π_k of each PPCA in the mixture model. We learn these parameters using the Expectation Maximization (EM) algorithm based on the given blendshape animation sequences $\mathcal{A}$. To increase the robustness of these computations, we estimate the MPPCA in a latent space of the animation sequences $\mathcal{A}$ using principal component analysis. By keeping 99% of the total variance we can reduce the dimensionality of the training data by two-thirds allowing a more stable learning phase with the EM algorithm. Equation 8 can thus be rewritten as

$$p(\mathbf{x}, X_n) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}, X_n|P\mu_k + \mu, PMP^T), \quad (9)$$

where $M = (C_k C_k^T + \sigma_k^2 I)$ is the covariance matrix in the latent space, P is the principal component matrix, and μ the mean vector. Since the EM algorithm converges to local minima, we run the algorithm 50 times with random initialization to improve the learning accuracy. We use 20 Gaussians to model the prior distribution and we use one-third of the latent space dimension for the PPCA dimension. More details on the implementation of the EM algorithm can be found in [McLachlan and Krishnan 1996].

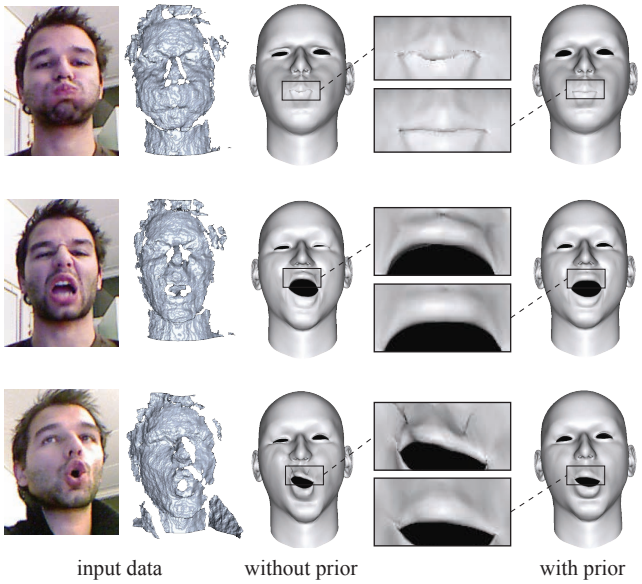


Figure 8: *Without the animation prior, tracking inaccuracies lead to visually disturbing self-intersections. Our solution significantly reduces these artifacts. Even when tracking is not fully accurate as in the bottom row, a plausible pose is reconstructed.*

Likelihood Distribution. By assuming conditional independence, we can model the likelihood distribution in Equation 5 as the product $p(D|\mathbf{x}) = p(G|\mathbf{x})p(I|\mathbf{x})$. The two factors capture the alignment of the blendshape model with the acquired depth map and texture image, respectively. We represent the distribution of each likelihood term as a product of Gaussians, treating each vertex of the blendshape model independently.

Let V be the number of vertices in the template mesh and $B \in \mathbb{R}^{V \times m}$ the blendshape matrix. Each column of B defines a blendshape base mesh such that $B\mathbf{x}$ generates the blendshape representation of the current pose. We denote with $\mathbf{v}_i = (B\mathbf{x})_i$ the i-th vertex of the reconstructed mesh. The likelihood term $p(G|\mathbf{x})$ models a geometric registration in the spirit of non-rigid ICP by assuming a Gaussian distribution of the per-vertex point-plane distances

$$p(G|\mathbf{x}) = \prod_{i=1}^{V} \frac{1}{(2\pi\sigma_{\text{geo}}^2)^{\frac{3}{2}}} \exp\left(-\frac{||\mathbf{n}_i^T(\mathbf{v}_i - \mathbf{v}_i^*)||^2}{2\sigma_{\text{geo}}^2}\right), \quad (10)$$

where $\mathbf{n}_i$ is the surface normal at $\mathbf{v}_i$, and $\mathbf{v}_i^*$ is the corresponding closest point in the depth map G.

The likelihood term $p(I|\mathbf{x})$ models texture registration. Since we acquire the user's face texture when building the facial expression model (Figure 3), we can integrate model-based optical flow constraints [Decarlo and Metaxas 2000], by formulating the likelihood function using per-vertex Gaussian distributions as

$$p(I|\mathbf{x}) = \prod_{i=1}^{V} \frac{1}{2\pi\sigma_{\text{im}}^2} \exp\left(-\frac{||\nabla I_i^T(\mathbf{p}_i - \mathbf{p}_i^*)||^2}{2\sigma_{\text{im}}^2}\right), \quad (11)$$

where $\mathbf{p}_i$ is the projection of $\mathbf{v}_i$ into the image I, ∇I_i is the gradient of I at $\mathbf{p}_i$, and $\mathbf{p}_i^*$ is the corresponding point in the rendered texture image.

3.2 Optimization

In order to solve the MAP problem as defined by Equation 5 we minimize the negative logarithm, i.e.,

$$\mathbf{x}^\star = \arg\min_{\mathbf{x}} -\ln p(G|\mathbf{x}) - \ln p(I|\mathbf{x}) - \ln p(\mathbf{x}, X_n). \quad (12)$$

Discarding constants, we write

$$\mathbf{x}^\star = \arg\min_{\mathbf{x}} E_{\text{geo}} + E_{\text{im}} + E_{\text{prior}}, \quad (13)$$

where

$$E_{\text{prior}} = -\ln p(\mathbf{x}, X_n), \quad (14)$$

$$E_{\text{geo}} = \frac{1}{\sigma_{\text{geo}}^2} \sum_{i=1}^{V} ||\mathbf{n}_j^T(\mathbf{v}_i - \mathbf{v}_i^*)||^2, \text{and} \quad (15)$$

$$E_{\text{im}} = \frac{1}{\sigma_{\text{im}}^2} \sum_{i=1}^{V} ||\nabla I_i^T(\mathbf{p}_i - \mathbf{p}_i^*)||^2. \quad (16)$$

The parameters σ_{geo} and σ_{im} model the noise level of the data that controls the emphasis of the geometry and image likelihood terms relative to the prior term. Since our system provides realtime feedback, we can experimentally determine suitable values that achieve stable tracking performance. For all our results we use the same settings $\sigma_{\text{geo}} = 1$ and $\sigma_{\text{im}} = 0.45$.

The optimization of Equation 13 can be performed efficiently using an iterative gradient solver, since the gradients can be computed analytically (see the derivations in the Appendix). In addition, we precompute the inverse covariance matrices and the determinants of the MPPCA during the offline learning phase. We use a gradient projection algorithm based on the limited memory BFGS solver [Lu et al. 1994] in order to enforce that the blendshape weights are between 0 and 1. The algorithm converges in less that 6 iterations as we can use an efficient warm starting with the previous solution. We then update the closest point correspondences in E_{geo} and E_{im}, and re-compute the MAP estimation. We found that 3 iterations of this outer loop are sufficient for convergence.

4 Results

We present results of our realtime performance capture and animation system and illustrate potential applications. The output of the tracking optimization is a continuous stream of blendshape weight vectors $\{\mathbf{x}_i\}$ that drive the digital character. Please refer to the accompanying video to better appreciate the facial dynamics of the animated characters and the robustness of the tracking. Figures 1 and 9 illustrates how our system can be applied in interactive applications, where the user controls a digital avatar in realtime. Blendshape weights can be transmitted in realtime to enable virtual encounters in cyberspace. Since the blendshape representation facilitates animation transfer, the avatar can either be a digital representation of the user himself or a different humanoid character, assuming compatible expression spaces.

While we build the user-specific blendshape model primarily for realtime tracking, our technique offers a simple way to create personalized blendshape rigs that can be used in traditional animation tools. Since the Kinect is the only acquisition device required, generating facial rigs becomes accessible for non-professional users.

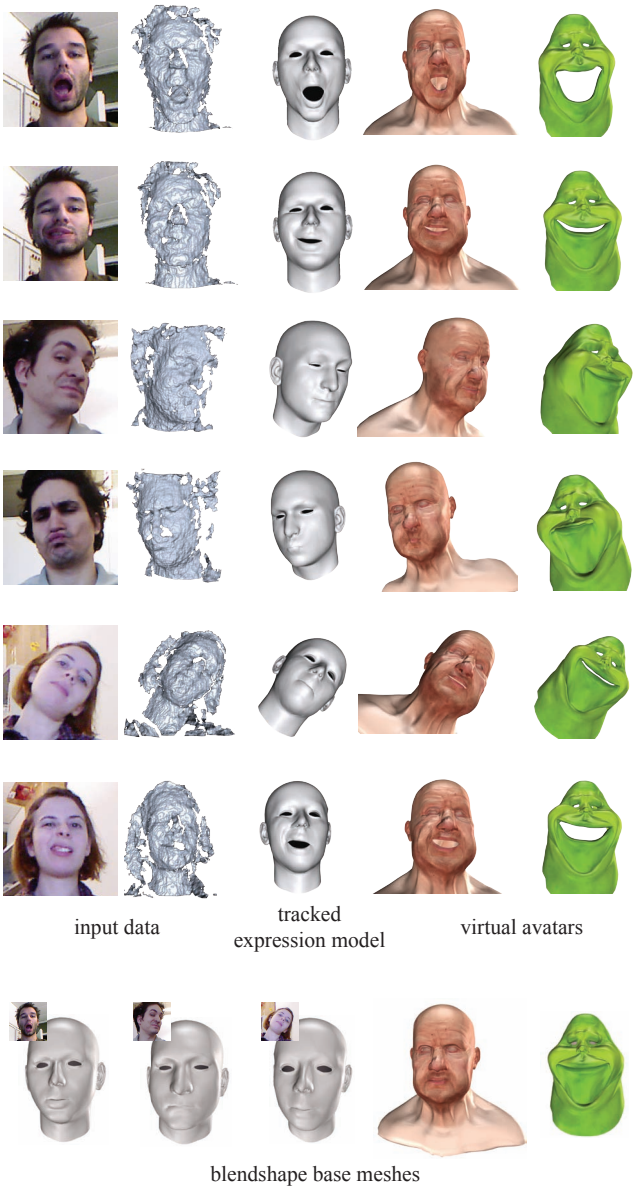


Figure 9: *The user's facial expressions are reconstructed and mapped to different target characters in realtime, enabling interactive animations and virtual conversations controlled by the performance of the tracked user. The smile on the green character's base mesh gives it a happy countenance for the entire animation.*

Statistics. We use 15 user-specific expressions to reconstruct 39 blendshapes for the facial expression model. Manual markup of texture constraints for the initial offline model building requires approximately 2 minutes per expression. Computing the expression model given the user input takes less than 10 minutes. We precompute the Gaussian mixture model that defines the dynamic expression prior from a total of 9,500 animation frames generated on the generic template model by an animation artist. Depending on the size of the temporal window, these computations take between 10 and 20 minutes.

Our online system achieves sustained framerates of 20 Hertz with a latency below 150 ms. Data acquisition, preprocessing, rigid registration, and display take less than 5 ms. Nonrigid registration

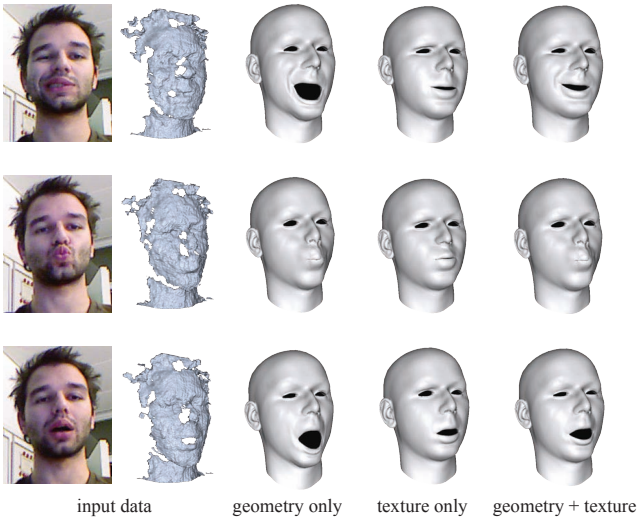


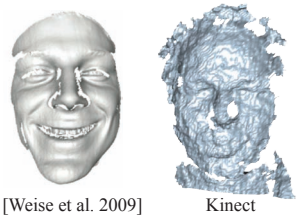
Figure 10: *The combination of geometric and texture-based registration is essential for realtime tracking. To isolate the effects of the individual components, no animation prior is used in this example.*

including constraint setup and gradient optimization require 45 ms per frame. All timing measurements have been done on a Intel I7 2.8Ghz with 8 GBytes of main memory and a ATI Radeon HD 4850 graphics card.

5 Evaluation

We focus our evaluation on the integration of 2D and 3D input data and the effect of animation training data. We also comment on limitations and drawbacks of our approach.

Geometry and Texture. Figure 10 evaluates the interplay between the geometry and texture information acquired with the Kinect.



Tracking purely based on geometry as proposed in [Weise et al. 2009] is not successful due to the high noise level of the Kinect data. Integrating model-based optical flow constraints reduces temporal jitter and stabilizes the reconstruction. In our experiments, only the combination of both modalities yielded satisfactory results. Compared to purely image-based tracking as e.g. in [Chai et al. 2003], direct access to 3D geometry offers two main benefits: We can significantly improve the robustness of the rigid pose estimation in particular for non-frontal views (see also Figure 7). In addition, the expression template mesh generated during preprocessing much more closely matches the geometry of the user, which further improves tracking accuracy. Figure 11 shows difficult tracking configurations and provides an indication of the limits of our algorithm.

Animation Prior. Figure 12 studies the effectiveness of our probabilistic tracking algorithm when varying the amount of training data used for the reconstruction. The figure illustrates that if the training data does not contain any sequences that are sufficiently close to the captured performance, the reconstruction can differ substantially from the acquired data. With more training data, the

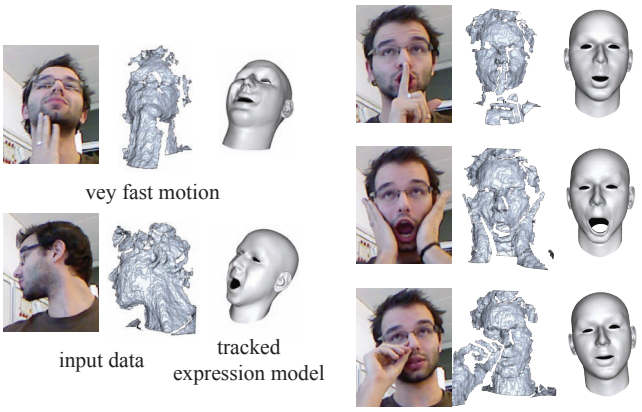


Figure 11: *Difficult tracking configurations. Right: despite the occlusions by the hands, our algorithm successfully tracks the rigid motion and the expression of the user. Left: with more occlusion or very fast motion, tracking can fail.*

tracked model more closely matches the performing user. What the prior achieves in any case is that the reconstructed pose is plausible, even if not necessarily close to the input geometrically (see also Figure 8). We argue that this is typically much more tolerable than generating unnatural or even physically impossible poses that could severely degrade the visual perception of the avatar. In addition, our approach is scalable in the sense that if the reconstructed animation does not well represent certain expressions of the user, we can manually correct the sequence using standard blendshape animation tools and add the corrected sequence to the training data set. This allows to successively improve the animation prior in a bootstrapping manner. For the temporal window X_n used in the animation prior, we found a window size of $3 \leq n \leq 5$ to yield good results in general. Longer temporal spans raise the dimensionality and lead to increased temporal smoothing. If the window is too small, temporal coherence is reduced and discontinuities in the tracking data can lead to artifacts.

Limitations. The resolution of the acquisition system limits the amount of geometric and motion detail that can be tracked for each user, hence slight differences in expressions will not be captured adequately. This limitation is aggravated by the wide-angle lens of the Kinect installed to enable full-body capture, which confines the face region to about 160×160 pixels or less than 10% of the total image area. As a result, our system cannot recover small-scale wrinkles or very subtle movements. We also currently do not model eyes, teeth, tongue, or hair.

In our current implementation, we require user support during preprocessing in the form of manual markup of lip and eye features to register the generic template with the recorded training poses (see Figure 5). In future work, we want to explore the potential of generic active appearance models similar to [Cootes et al. 2001] to automate this step of the offline processing pipeline as well.

While offering many advantages as discussed in Section 1.2, the blendshape representation also has an inherent limitation: The number of blendshapes is a tradeoff between expressiveness of the model and suitability for tracking. Too few blendshapes may result in user expressions that cannot be represented adequately by the pose space of the model. Introducing additional blendshapes to the rig can circumvent this problem, but too many blendshapes may result in a different issue: Since blendshapes may become approximately linearly dependent, there might not be a unique set of

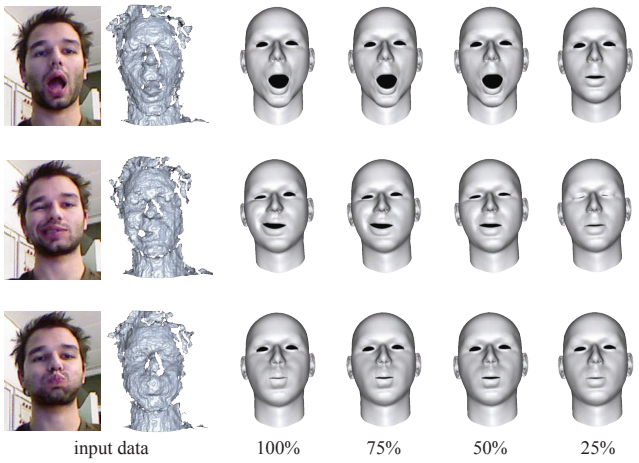


Figure 12: *Effect of different amounts of training data on the performance of the tracking algorithm. We successively delete blendshapes from the input animation sequences, which removes entire portions of the expression space. With only 25% of the blendshapes in the training data the expressions are not reconstructed correctly.*

blendshape weights for a given expression. This can potentially result in unstable tracking due to overfitting of the noisy data. While the prior prevents this instability, a larger number of blendshapes requires a larger training database and negatively affects performance.

6 Conclusion

We have demonstrated that high-quality performance-driven facial animation in realtime is possible even with a low-cost, non-intrusive, markerless acquisition system. We show the potential of our system for applications in human interaction, live virtual TV shows, and computer gaming.

Robust realtime tracking is achieved by building suitable user-specific blendshape models and exploiting the different characteristics of the acquired 2D image and 3D depth map data for registration. We found that learning the dynamic expression space from existing animations is essential. Combining these animation priors with effective geometry and texture registration in a single MAP estimation is our key contribution to achieve robust tracking even for highly noisy input data. While foreseeable technical advances in acquisition hardware will certainly improve data quality in coming years, numerous future applications, e.g. in multi-people tracking, acquisition with mobile devices, or performance capture in difficult lighting conditions, will produce even worse data and will thus put even higher demands on robustness. Our algorithm provides a systematic framework for addressing these challenging problems.

We believe that our system enables a variety of new applications and can be the basis for substantial follow-up research. We currently focus on facial acquisition and ignore other important aspects of human communication, such as hand gestures, which pose interesting technical challenges due to complex occlusion patterns. Enhancing the tracking performance using realtime speech analysis, or integrating secondary effects such as simulation of hair are further areas of future research that could help increase the realism of the generated virtual performances. More fundamentally, being able to deploy our system at a massive scale can enable interesting new research in human communication and paves the way for new interaction metaphors in performance-based game play.

Acknowledgements. We are grateful to Lee Perry-Smith for providing the face model for our generic template, Dan Burke for sculpting the CG characters, and Cesar Bravo, Steven McLellan, David Rodrigues, and Volker Helzle for the animations. We thank Gabriele Fanelli for our valuable discussions, Duygu Ceylan and Mario Deuss for being actors, and Yuliy Schwartzburg for proofreading the paper. This research is supported by Swiss National Science Foundation grant 20PA21L_129607.

Appendix

We derive the gradients for the optimization of Equation 13. The energy terms for geometry registration E_{geo} and optical flow E_{im} can both be written in the form

$$f(\mathbf{x}) = \frac{||A\mathbf{x} - b||^2}{2\sigma^2} \tag{17}$$

hence the gradients can easily be computed analytically as

$$\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} = \frac{A^T(A\mathbf{x} - b)}{\sigma^2}. \tag{18}$$

The prior term is of the form

$$E_{\text{prior}} = -\ln \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}, X_n | \mu_k, \Sigma_k), \tag{19}$$

where Σ_k is the covariance matrix. The Gaussians $\mathcal{N}(\mathbf{x}, X_n|\mu_k, \Sigma_k)$ model the combined distribution of the current blendshape vector $\mathbf{x} \in \mathbb{R}^m$ and the n previous vectors X_n, hence the Σ_k are matrices of dimension $(n+1)m \times (n+1)m$. Since we are only interested in the gradient with respect to $\mathbf{x}$, we can discard all components that do not depend on this variable. We split the mean vectors as $\mu_k = (\mu_k^1, \mu_k^n)$, corresponding to $\mathbf{x}$ and X_n, respectively. We can write the inverse of Σ_k as

$$\Sigma_k^{-1} = \left[\begin{array}{c|c} A_k & B_k \\ \hline C_k & D_k \end{array}\right] = \left[\begin{array}{c|c} (m \times m) & (m \times nm) \\ \hline (nm \times m) & (nm \times nm) \end{array}\right] \tag{20}$$

with $B_k = C_k^T$. We then obtain for the gradient of the prior energy term

$$\frac{\partial E_{\text{prior}}}{\partial \mathbf{x}} = \tag{21}$$
$$\frac{\sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}, X_n|\mu_k, \Sigma_k)[(\mathbf{x} - \mu_k^1)^T A_k + (X_n - \mu_k^n)^T C_k]}{\sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}, X_n|\mu_k, \Sigma_k)}.$$

The complete gradient is the sum of the three energy gradients derived above

$$g(\mathbf{x}) = \frac{\partial E_{\text{geo}}}{\partial \mathbf{x}} + \frac{\partial E_{\text{im}}}{\partial \mathbf{x}} + \frac{\partial E_{\text{prior}}}{\partial \mathbf{x}}. \tag{22}$$

References

ALEXANDER, O., ROGERS, M., LAMBETH, W., CHANG, M., AND DEBEVEC, P. 2009. The digital emily project: photoreal facial modeling and animation. *ACM SIGGRAPH 2009 Courses*.

BEELER, T., BICKEL, B., BEARDSLEY, P., SUMNER, B., AND GROSS, M. 2010. High-quality single-shot capture of facial geometry. *ACM Trans. Graph. 29*, 40:1–40:9.

BLACK, M. J., AND YACOOB, Y. 1995. Tracking and recognizing rigid and non-rigid facial motions using local parametric models of image motion. In *ICCV*, 374–381.

BLANZ, V., AND VETTER, T. 1999. A morphable model for the synthesis of 3d faces. In *Proc. SIGGRAPH 99*.

BORSHUKOV, G., PIPONI, D., LARSEN, O., LEWIS, J. P., AND TEMPELAAR-LIETZ, C. 2005. Universal capture - image-based facial animation for "the matrix reloaded". In *SIGGRAPH 2005 Courses*.

BRADLEY, D., HEIDRICH, W., POPA, T., AND SHEFFER, A. 2010. High resolution passive facial performance capture. *ACM Trans. Graph. 29*, 41:1–41:10.

CHAI, J. X., XIAO, J., AND HODGINS, J. 2003. Vision-based control of 3d facial animation. In *SCA*.

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

COOTES, T., EDWARDS, G., AND TAYLOR, C. 2001. Active appearance models. *PAMI 23*, 681 –685.

COVELL, M. 1996. Eigen-points: Control-point location using principle component analyses. In *FG '96*.

DECARLO, D., AND METAXAS, D. 1996. The integration of optical flow and deformable models with applications to human face shape and motion estimation. In *CVPR*.

DECARLO, D., AND METAXAS, D. 2000. Optical flow constraints on deformable models with applications to face tracking. *IJCV 38*, 99–127.

EKMAN, P., AND FRIESEN, W. 1978. *Facial Action Coding System: A Technique for the Measurement of Facial Movement.* Consulting Psychologists Press.

ESSA, I., BASU, S., DARRELL, T., AND PENTLAND, A. 1996. Modeling, tracking and interactive animation of faces and heads using input from video. In *Proc. Computer Animation*.

FURUKAWA, Y., AND PONCE, J. 2009. Dense 3d motion capture for human faces. In *CVPR*.

GROCHOW, K., MARTIN, S. L., HERTZMANN, A., AND POPOVIĆ, Z. 2004. Style-based inverse kinematics. *ACM Trans. Graph. 23*, 522–531.

GUENTER, B., GRIMM, C., WOOD, D., MALVAR, H., AND PIGHIN, F. 1993. Making faces. *IEEE Computer Graphics and Applications 13*, 6–8.

IKEMOTO, L., ARIKAN, O., AND FORSYTH, D. 2009. Generalizing motion edits with gaussian processes. *ACM Trans. Graph. 28*, 1:1–1:12.

LAU, M., CHAI, J., XU, Y.-Q., AND SHUM, H.-Y. 2007. Face poser: interactive modeling of 3d facial expressions using model priors. In *SCA*.

LI, H., ROIVAINEN, P., AND FORCHEIMER, R. 1993. 3-d motion estimation in model-based facial image coding. *PAMI 15*, 545–555.

LI, H., ADAMS, B., GUIBAS, L. J., AND PAULY, M. 2009. Robust single-view geometry and motion reconstruction. *ACM Trans. Graph. 28*, 175:1–175:10.

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

LIN, I.-C., AND OUHYOUNG, M. 2005. Mirror mocap: Automatic and efficient capture of dense 3d facial motion parameters from video. *The Visual Computer 21*, 6, 355–372.

LOU, H., AND CHAI, J. 2010. Example-based human motion denoising. *IEEE Trans. on Visualization and Computer Graphics 16*, 870–879.

LU, P., NOCEDAL, J., ZHU, C., BYRD, R. H., AND BYRD, R. H. 1994. A limited-memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing*.

MA, W.-C., HAWKINS, T., PEERS, P., CHABERT, C.-F., WEISS, M., AND DEBEVEC, P. 2007. Rapid acquisition of specular and diffuse normal maps from polarized spherical gradient illumination. In *EUROGRAPHICS Symposium on Rendering*.

MCLACHLAN, G. J., AND KRISHNAN, T. 1996. *The EM Algorithm and Extensions*. Wiley-Interscience.

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

PIGHIN, F., AND LEWIS, J. P. 2006. Performance-driven facial animation. In *ACM SIGGRAPH 2006 Courses*.

PIGHIN, F., SZELISKI, R., AND SALESIN, D. 1999. Resynthesizing facial animation through 3d model-based tracking. *ICCV 1*, 143–150.

ROBERTS, S. 1959. Control chart tests based on geometric moving averages. In *Technometrics*, 239250.

TIPPING, M. E., AND BISHOP, C. M. 1999. Probabilistic principal component analysis. *Journal of the Royal Statistical Society, Series B*.

TIPPING, M. E., AND BISHOP, C. M. 1999. Mixtures of probabilistic principal component analyzers. *Neural Computation 11*.

VIOLA, P., AND JONES, M. 2001. Rapid object detection using a boosted cascade of simple features. In *CVPR*.

WEISE, T., LEIBE, B., AND GOOL, L. V. 2008. Accurate and robust registration for in-hand modeling. In *CVPR*.

WEISE, T., LI, H., GOOL, L. V., AND PAULY, M. 2009. Face/off: Live facial puppetry. In *SCA*.

WILLIAMS, L. 1990. Performance-driven facial animation. In *Comp. Graph. (Proc. SIGGRAPH 90)*.

WILSON, C. A., GHOSH, A., PEERS, P., CHIANG, J.-Y., BUSCH, J., AND DEBEVEC, P. 2010. Temporal upsampling of performance geometry using photometric alignment. *ACM Trans. Graph. 29*, 17:1–17:11.

ZHANG, S., AND HUANG, P. 2004. High-resolution, real-time 3d shape acquisition. In *CVPR Workshop*.

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