Real-Time High-Fidelity Facial Performance Capture

Chen Cao[1,2] Derek Bradley[2] Kun Zhou[1] Thabo Beeler[2]
1) State Key Lab of CAD&CG, Zhejiang University 2) Disney Research Zurich

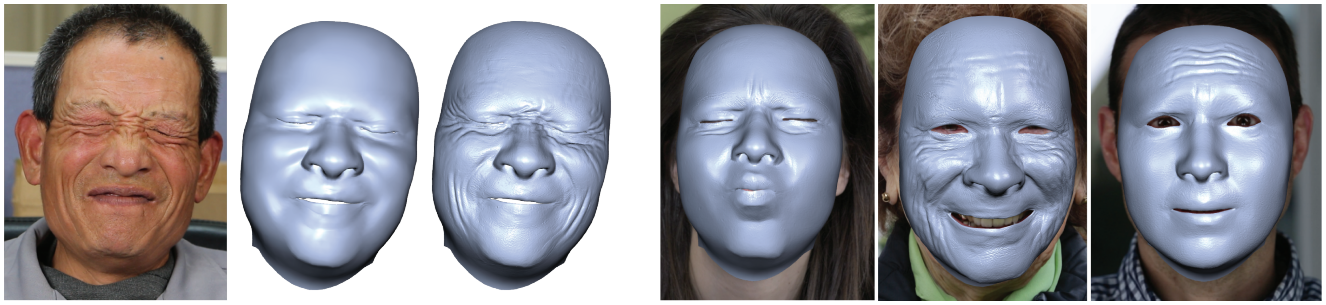

Figure 1: *We propose the first real-time facial tracking system that captures performances in online at high fidelity, including medium scale details such as wrinkles. From a monocular input (left) our system captures both the global shape as well as local details (center). The method is generic and requires no offline training or manual preprocessing steps for novel users (right).*

Abstract

We present the first real-time high-fidelity facial capture method. The core idea is to enhance a global real-time face tracker, which provides a low-resolution face mesh, with local regressors that add in medium-scale details, such as expression wrinkles. Our main observation is that although wrinkles appear in different scales and at different locations on the face, they are locally very self-similar and their visual appearance is a direct consequence of their local shape. We therefore train local regressors from high-resolution capture data in order to predict the local geometry from local appearance at runtime. We propose an automatic way to detect and align the local patches required to train the regressors and run them efficiently in real-time. Our formulation is particularly designed to enhance the low-resolution global tracker with exactly the missing expression frequencies, avoiding superimposing spatial frequencies in the result. Our system is generic and can be applied to any real-time tracker that uses a global prior, e.g. blend-shapes. Once trained, our online capture approach can be applied to any new user without additional training, resulting in high-fidelity facial performance reconstruction with person-specific wrinkle details from a monocular video camera in real-time.

CR Categories: I.3.3 [Computer Graphics]: Picture/Image Generation—Digitizing and scanning;

Keywords: Real-time face reconstruction, local wrinkle model, high-fidelity performance capture.

1 Introduction

Mastering facial animation is a long-standing challenge in computer graphics. The face can describe the emotions of a character, convey their state of mind, and hint at their future actions. Audiences are particularly trained to look at faces and identify these subtle characteristics. Through decades of research, we have learned that capturing the shape and motion of real human faces can lead to realistic virtual characters.

Facial motion capture has come a long way from the original marker-based tracking approaches. Modern performance capture techniques can deliver extremely high-resolution facial geometry with very high fidelity motion information. In recent years the growing trend has been to capture faces in real-time, opening up new applications in immersive computer games, social media and real-time preview for visual effects. These methods approximate the 3D shape and motion of a face during the performance using either depth or web cameras. To make this tractable, real-time approaches use generic, low-resolution face models as a basis for the reconstruction. While these models simplify the capture problem and facilitate performance retargeting to other characters, they fail to capture the unique medium and fine scale facial details of the individual, such as wrinkles on the forehead or the so-called crows feet around the eyes. As a result, real-time performance comes at the cost of facial fidelity, and there exists a large gap in reconstruction quality between current offline and online capture methods.

In this work we aim to reduce the disparity between offline and real-time facial performance capture by presenting the first high-fidelity real-time facial capture method. Our core idea is to enhance an existing low-resolution real-time tracker by adding a local regression method that targets medium scale expression wrinkles. Our main observation is that although wrinkles appear at different locations on the face and with different orientations and scales, they exhibit very similar local appearance because of the way they are formed. The local appearance of wrinkles is a direct consequence of the shading caused by the local shape. By learning the relationship between local image appearance and local wrinkle formation, we can reconstruct plausible face wrinkles in real-time from a single RGB camera. Specifically, we train a regressor with data acquired by a high-resolution performance capture system, which can predict the shape of these features given the captured appearance in an image. The local details are stored in a displacement map that is applied to

the existing global model.

Our approach is general, in that we can extend any existing low-resolution real-time face model (e.g. those based on blend-shapes). In this paper we employ the global face tracking method of Cao et al. [2014]. Our method also provides both robustness and flexibility, as the training can be performed in a one-time pre-processing step given high-resolution capture data (e.g. from Beeler et al. [2011]) and then applied online to any new user.

Our high-fidelity capture approach is made possible through several new technical contributions, including

1. a local description of face wrinkles that enables us to develop a generic model capable of creating user-specific wrinkling in realtime,
2. a method to extract and apply targeted facial details from high-resolution training data, avoiding superimposing spatial frequencies at runtime, and
3. a local regression algorithm specifically designed for medium-scale expression details, which extends traditional boosted regression by making use of the structure present in the data to achieve real-time performance.

To our knowledge, our method is the first for real-time facial performance capture that achieves high-resolution facial fidelity, and we achieve this with a monocular tracking approach. As we will demonstrate, our results are on par with offline reconstruction methods. By offering more realistic facial capture in real-time, our technique allows advances in various application domains including online preview for high-quality offline capture methods, more immersive gaming and social experiences, and the potential for more realistic online facial retargeting to virtual characters.

2 Related Work

Facial performance capture has received a lot of attention over the past decade. During this time, researchers have attacked the problem from two different sides. One line of research has investigated ways to capture the spatio-temporal shape of the face at very high resolution and accuracy. Based on methods designed to acquire single static expressions [Ma et al. 2007; Beeler et al. 2010; Ghosh et al. 2011], researchers have proposed various methods to establish temporal correspondence [Zhang et al. 2004; Furukawa and Ponce 2009; Bradley et al. 2010; Beeler et al. 2011; Huang et al. 2011; Klaudiny and Hilton 2012]. These methods can capture facial performances at very high resolution but require complex setups with multiple cameras and controlled illumination. Therefore, more recent approaches have investigated how to reduce the hardware complexity to binocular [Valgaerts et al. 2012] and lately even monocular setups [Garrido et al. 2013; Suwajanakorn et al. 2014; Shi et al. 2014]. Still, all these approaches can be considered offline methods as they require substantial amounts of time to process the data and typically can integrate both forward and backward information to improve the quality.

Another line of research has focused on online, real-time facial performance capture. The first approaches in this domain leveraged depth information provided by active techniques such as custom built structured light setups [Weise et al. 2009] and depth sensors [Weise et al. 2011; Bouaziz et al. 2013; Li et al. 2013; Chen et al. 2013](Chen et al. added). More recently, methods have started to emerge that operate only on a single monocular camera [Chai et al. 2003; Rhee et al. 2011; Cao et al. 2013; Cao et al. 2014](Chai et al. added). To provide real-time performance and remain robust, these online methods typically employ a strong prior of the face, such as a blend-shape rig. While providing robustness, this global prior also limits the accuracy that may be achieved. Thus the resulting performances lack person-specific details, such as accurate wrinkling, which are very important for the visual appearance.

Researchers have long recognized the importance of these medium-scale details. Several approaches have been proposed to handle medium-scale details explicitly, either by spectrally decomposing the captured face geometry into multiple scales [Bickel et al. 2007; Ma et al. 2008; Bermano et al. 2014] or by acquiring only the details [Dutreve et al. 2011; Li et al. 2015]. These can then be added to the low-frequency shape provided, for example, from mocap tracking or transferred to other characters. All of these methods, however, are non-generic and require either an offline preprocessing stage to build up an actor-specific dataset or an artist to specify where wrinkles should appear on the character.

Most related to the proposed method is the recent work of Li et al. [2015], in which they propose to synthesize a displacement map from a pool of training examples using texture synthesis. As their method is offline, they do not apply the wrinkle enhancement to every frame, but instead require an offline preprocessing step, where they acquire 20 expressions of a user which are augmented by their wrinkle synthesis method. These 20 shapes form the blend-shape prior for their global tracker, which is consequently limited to remain within the confines of this linear prior. Our method does not require any offline pre-processing for a novel user but only a few seconds of unsupervised online training at the beginning and can capture the complex, non-linear dynamic wrinkling behaviour of skin.

3 Method Overview

The core idea proposed in this paper is to augment a low resolution face mesh L^t, tracked in real-time to frame t by a global tracker [Cao et al. 2014], with appropriate details D^t computed with a local regressor to produce the final high fidelity mesh H^t. This combines the robustness of a global tracker with the flexibility of local regressors, while still maintaining real-time performance. The proposed pipeline consists of two main stages, *Training* and *Online Performance Capture*, as outlined in Figure 2.

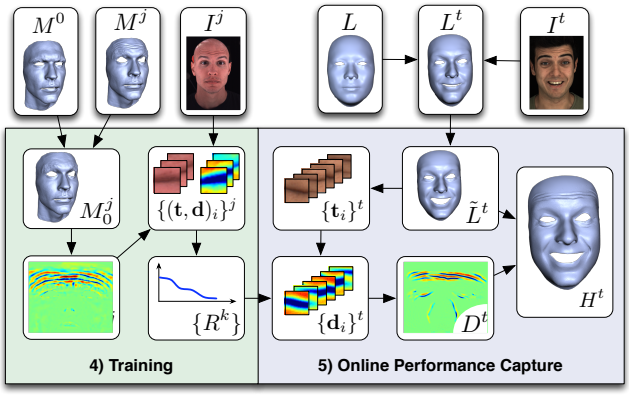

Figure 2: *This figure summarizes the various data types used throughout the paper as well as their interdependencies. The left side shows how the high-resolution sample data provided by an offline reconstruction system is used to train the proposed detail regressor (Section 4). The right side focuses on runtime dataflow and shows how the coarse mesh provided by a global tracker is refined and augmented with medium-scale frequencies (Section 5).*

In the training stage (Section 4) we train a boosted regressor with data acquired by a high-resolution performance capture sys-

Figure 3: *A subset of the high-fidelity training data used to learn the wrinkle regression.*

tem [Beeler et al. 2012]. Training takes place on small patches which are automatically detected in the UV domain. From these patches the regression aims to learn a function Φ that infers local shape from local shading, or more formally

$$\Phi(\mathbf{t}_i) = \mathbf{d}_i, \tag{1}$$

where $\mathbf{t}_i$ is the local image texture and $\mathbf{d}_i$ the local displacement map of the i-th patch. An important component is the proper definition and computation of the detail displacement $\mathbf{d}$ as explained in Section 4.1 to avoid superimposing spatial frequencies, which would lead to undesired artifacts. Training takes place only once, after which the system can be applied to any number of novel users.

At runtime (Section 5), the system first fits the low resolution face mesh to the input image I^t using the global tracker to get an initial estimate. This estimate is then refined using optical flow to produce accurate tracking and thus a temporally stable texture T^t. From this texture, the trained local regressors compute the displacement map D^t, which is used to create the high-resolution face mesh H^t on the GPU. In the following we discuss the individual steps in detail.

4 Training

We train a local detail regressor on a sparse set of high-quality 3D face scans, consisting of various extreme expressions that demonstrate wrinkles, from several different actors, including the captured images. We assume that within the same actor, the subset of meshes have the same topology, and that one of the meshes contains a neutral expression. Such a dataset can be acquired with existing facial capture systems, such as the method of Beeler et al. [2012], which we use in this work.

We begin by extracting the details from the high-resolution meshes as displacement maps, paying close attention to encode only the details that would not already be present in the low-resolution online tracker. We then automatically detect wrinkle patches in the displacement maps, and train a regressor to learn the mapping from local image appearance to wrinkle displacements.

4.1 Detail Extraction

Our training dataset consists of a total of 18 meshes captured from four different actors. A subset of the training data is shown in Figure 3. In order to describe the detail extraction, let us consider the subset $\{M^j\}$, which consists of the meshes of a single actor, and assume M^0 is the neutral expression. All other actors are processed

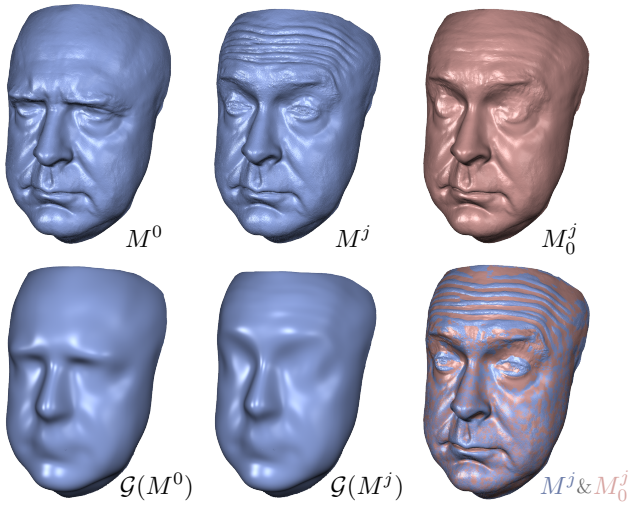

Figure 4: *We extract medium expression details from the high-resolution training expression M^j. We remove the medium frequencies from both the expression M^j and the neutral mesh M^0 and transfer the deformation $\mathcal{G}(M^0) \rightarrow \mathcal{G}(M^j)$ to M^0. The resulting mesh M_0^j does not contain the medium expression details but preserves all other frequencies (bottom right).*

analogously. One expression and a neutral mesh are shown in Figure 4, top left and center. The proposed detail regression is only concerned with the medium frequency details and we thus need to remove both the low and high frequency bands from the training data. Removing the high frequencies (e.g. pores) is straightforward and is achieved by smoothing the meshes with a narrow Gaussian filter $\{M^j\} \leftarrow \{\mathcal{G}^{HF}(M^j)\}$. Removing the low frequencies however is more involved, as these depend on the coarse tracking mesh L. We define a lowpass filter $\mathcal{G}$ which will serve as bandpass to separate low and medium frequencies. With this bandpass filter we wish to extract the medium expression frequencies from the training mesh. Before we can do so, however, we have to ensure that the coarse mesh is compatible.

The coarse tracking mesh may also contain medium frequency expression details that need to be removed to avoid interference during detail regression. These are expression details that most humans have in common, such as the nasolabial fold. In other areas, such as the forehead, the medium expression frequencies tend to be missing from the coarse model, since the wrinkles form very differently for every person. The naïve way to remove these frequencies would be to filter L with the lowpass filter $\mathcal{G}$ to produce $\mathcal{G}(L)$. This, however, would not only remove the frequencies caused by the expression but also attenuate spatial frequencies important for the identity, such as parts of the nose. To separate the identity and expression frequencies from each other, we thus propose to compute the deformation from $\mathcal{G}(L^0)$ to $\mathcal{G}(L^b)$ and to transfer it to L^0 using deformation transfer [Sumner and Popović 2004] in order to create L_0^b, which preserves the identity frequency bands while removing the medium expression frequencies. We process all basis shapes B of our global tracker in this way and thus make it compatible for detail regression. From now on L refers to the compatible coarse mesh.

The goal during runtime will be to add expression wrinkles to the low-resolution result of a global face tracker. We use the recent real-time tracker of Cao et al. [2014], which uses an underlying face model with parameters for identity, expression and rigid transformation. After ensuring the coarse mesh L is compatible as described above, we solve for the identity parameter given all expres-

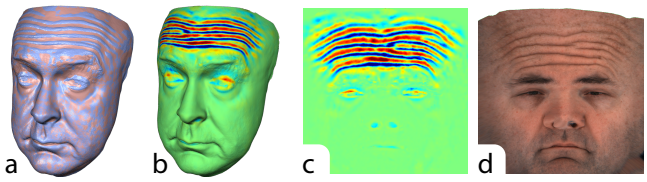

Figure 5: *Given the expression M^j and the aligned mesh M_0^j (a) we compute the positive (red) and negative (blue) displacements along the normal (b). These displacements are stored as a displacement map D^j (c) in the texture uv-layout (d).*

sions $\{M^j\}$. We assume the face model has a low-distortion UV mapping, which we transfer from the aligned neutral mesh L^0 to the high-resolution counterpart M^0.

To extract the expression details, we proceed analogously to preparing the coarse mesh by transferring the deformation from the filtered meshes $\mathcal{G}(M^0) \rightarrow \mathcal{G}(M^j)$ to the unfiltered neutral shape M^0 (Figure 4). The result is a mesh M_0^j, which exhibits the same overall expression as M^j but lacks the medium frequency expression details. These are exactly the relevant details that we aim to retrieve and encode as a displacement map. The displacement map D^j is computed from M_0^j by tracing rays along the normal (in both directions) for all vertices and computing the signed distance to the intersection with M^j. This process is visualized in Figure 5.

4.2 Local Wrinkle Patches

At this point we have image textures and displacement maps for each of the training expressions. We now define our local wrinkle model, which consists of small rectangular patches of both texture and displacement, oriented along wrinkle lines in UV-space.

The training expressions contain extreme facial wrinkles, which appear with strong shadows in the texture images. For a given actor, even though wrinkles may be spread over different parts of the face, the spatial location of wrinkle lines remains fixed over time and thus wrinkles will always appear in the same locations in UV-space. For this reason we can combine wrinkle information from all poses into a single *wrinkle map* for each actor, which encodes the per-pixel likelihood of a wrinkle forming at this location.

As before, let us consider the subset of textures $\{T^j\}$ and displacement maps $\{D^j\}$ that correspond to a single actor, and as before each actor is processed analogously. The actor-specific wrinkle map is created by identifying features in the individual texture images that potentially correspond to wrinkles. To this end, we apply a Difference of Gaussians (DoG) filter to each texture image, and then set the wrinkle map to the average filter response across textures for each filtered pixel value above a user-defined threshold α. A subset of texture images and the combined wrinkle map is shown in Figure 6. We similarly define a 2-channel gradient image by averaging the texture gradients of those pixels that contributed to the wrinkle map.

We now define local wrinkle patches of user-defined size $w \times w$ aligned with the wrinkle map. Due to the anatomy of the face, expression wrinkles are most likely to appear on the forehead, around the eyes, and along the crease of the cheek. We therefore constrain our computation to these areas. Even still, it is not necessary to create densely overlapping wrinkle patches, so we prioritize patch placement by creating a list ℓ of all pixels within the valid regions, sorted in decreasing order by probability. Wrinkle patches are then created iteratively by choosing the pixel p with highest probability from ℓ, placing a rectangle centered at p, and orienting the rectangle with the gradient at p (see Figure 6). A local wrinkle patch is then

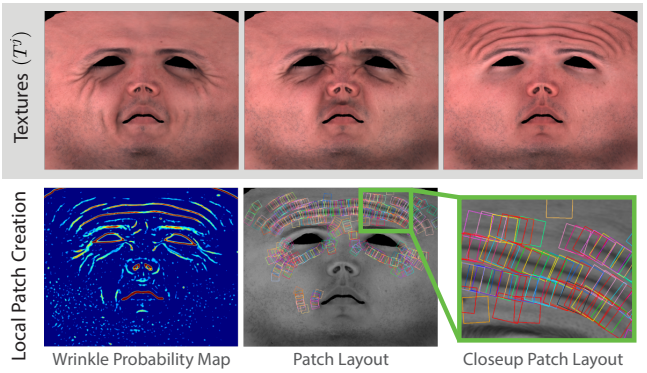

Figure 6: *From the sample textures T^j (top row) the method automatically determines the locations where wrinkling may occur by computing a wrinkle probability map (bottom, left). Guided by this map, patches are placed and oriented at these locations to extract texture patches $\{\mathbf{t}_i\}$ for training (bottom, center+right).*

defined by the pair of vectors $(\mathbf{t}, \mathbf{d})$, representing the image texture and displacement pixels within the rectangle, respectively. For each pixel p we generate a new patch for every expression j where the filtered texture image at p is larger than α. Once the patches for a pixel location are created, we remove all the pixels from ℓ that are covered by the sub-rectangle of size $\rho w \times \rho w$ centered at p, and continue until ℓ is empty. The parameter ρ allows us to control the patch density.

In all our examples, the resolution of the image texture and displacement maps is 1024×1024, and the Difference of Gaussians filter has a small kernel of size 6 and large kernel of size 8. The threshold α is fixed to 0.005, the patch size w is 32, and the density parameter ρ is set to 0.5. The training data comes from 18 meshes captured from four different actors undergoing different expressions. In our experience, too little training data can result in more pronounced wrinkle reconstructions but the result will be unstable, where too much training data can lead to over-smoothed wrinkle results. The dataset we employ produces a suitable trade-off. The type of wrinkles used for training can also have an impact on the regressed results, as some wrinkles are thin and deep while others are wide and contain only low-frequency shape. We selectively target the main expression wrinkles, which occur around the forehead, eyes and cheeks, however additional training sets could be used to target different types of wrinkles.

4.3 Detail Regression Training

To reduce the complexity and to better constrain the regression we aim to explore the structure present within the data. With this in mind we chose the patches in Section 4.2 such that they are centered at wrinkles and oriented consistently. Since these patches are therefore all aligned to each other we can reduce the dimensionality of our regression space substantially using principal component analysis (PCA) on the full set of N displacement patches $\{\mathbf{d}_i\}$ from all training data, and extract the 16 dimensions with the largest variance. These capture approximately 85% of the energy present in the training data. Figure 7 shows that these dimensions intuitively capture the predominant variations in wrinkle shape. In our experience, using higher dimensions does not significantly improve the wrinkle quality but can introduce noise and increase computation time, and fewer dimensions can lead to a loss of some high frequency details. Projecting the original displacement patches $\{\mathbf{d}_i\}$ into this subspace gives us the ground truth displacement coefficients $\{\hat{\mathbf{c}}_i\}$, which we use as the regression target. The input to the regression are the N corresponding patches $\{\mathbf{t}_i\}$ extracted from the

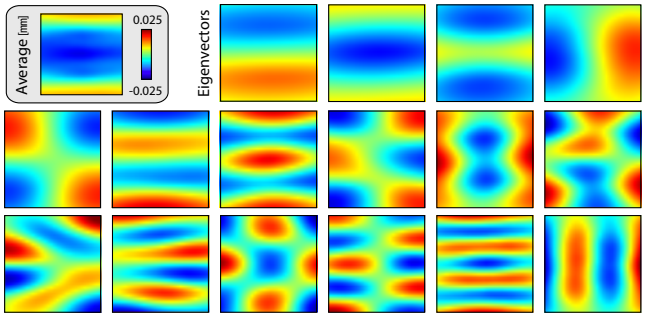

Figure 7: *The average and 16 major eigenvectors of the displacement patches from the training set capture about 85% of the energy and intuitively encode the predominant variations in wrinkle shape.*

image textures $\{T^j\}$.

From these patches we learn a regression function $\Phi^\star$ that maps $\mathbf{t}_i$ to $\hat{\mathbf{c}}_i$ by minimizing the energy

$$E^{reg} = \sum_{i=1}^{N} \|\Phi^\star(\mathbf{t}_i) - \hat{\mathbf{c}}_i\|^2 . \tag{2}$$

Reconstructing the output of $\Phi^\star$ with the PCA subspace basis links $\Phi^\star$ back to Φ defined in Equation 1.

Our regression function $\Phi^\star$ is similar in spirit to the boosted regressor proposed by Cao et al. [2012], which has shown to provide good results for similar tasks [Cao et al. 2013; Cao et al. 2014]. The boosted regressor has a hierarchical character by cascading K weak regressors $\{R^k\}$. This provides both robustness and accuracy since early regressors handle large scale variation which is progressively refined at later stages by regressing on the remaining residual from the previous stage $\boldsymbol{\varepsilon}_i^{k-1} = \hat{\mathbf{c}}_i - \mathbf{c}_i^{k-1}$. Cao et al. [2012] propose a two-level boosted regressor consisting of an external level, which is concerned with global alignment, and an internal level, which regresses the "normalized target" as the authors call it. Normalization is provided by the external level, which transforms the target into a canonical space. In our case however, we do not require the external level as our data is explicitly structured such that it is aligned. It is thus inherently normalized, which allows us to simplify the algorithm to a one-level regressor. Unlike the two-level weak regressor $R^k(\mathbf{t}_i, \mathbf{c}_i^{k-1})$ used in Cao et al. [2012], our one-level weak regressor $R^k(\mathbf{t}_i)$ does not depend on the previous result $\mathbf{c}_i^{k-1}$, which they only required for regressing the external level. This removes the sequential interdependency of the K weak regressors during runtime and allows to evaluate them in parallel.

Note that during training we still have the sequential interdependency as we train the weak regressor R^k on the residuals $\{\boldsymbol{\varepsilon}_i^{k-1}\}$ from the previous stage, which is desired as it preserves the hierarchical character. As weak regressors we use *random ferns* which we train on the input samples by minimizing

$$R^k = \underset{R}{\operatorname{argmin}} \sum_{i=1}^{N} \|\hat{\mathbf{c}}_i - (\mathbf{c}_i^{k-1} + R(\mathbf{t}_i))\|. \tag{3}$$

As an initial estimate of $\mathbf{c}_i^0$ we choose the average over the training set $\bar{\mathbf{c}} = \operatorname{avg}(\{\hat{\mathbf{c}}_i\})$.

The weak regressor R consists of F features and randomly selected thresholds, which divide the training samples into 2^F bins. As features we use intensity differences of two pixels in the image patch $\mathbf{t}_i$. Due to the inherent normalization of our regression data we do not need to employ shape indexed features as suggested by Cao et al. [2012], but instead can just randomly sample P pixels in the patches. These P pixels provide P^2 pixel pairs, from which we pick F good candidates according to the correlation-based feature selection proposed in Cao et al. [2012]. The regression output $\delta\mathbf{c}$ for each bin b is calculated as

$$\delta\mathbf{c}_b = \frac{\sum_{i\in\Omega_b} (\hat{\mathbf{c}}_i - \mathbf{c}_i)}{|\Omega_b| + \beta}, \tag{4}$$

where Ω_b is the subset of training samples falling into bin b and β is a free shrinkage parameter that prevents overfitting when there is insufficient training data in the bin. Subsequently, we update the coefficients $\mathbf{c}_i$ for samples in Ω_b as $\mathbf{c}_i \leftarrow \mathbf{c}_i + \delta\mathbf{c}_b$ to prepare for the next regression level. For the results generated in this paper, we choose $K = 1280, F = 5, P = 50$ and $\beta = 1000$.

5 Online Performance Capture

After training our algorithm on a sparse set of example expressions, our online capture method can reconstruct high-fidelity facial performances of novel users from a single camera. We start by tracking the low-resolution face mesh using the global face tracker. This face mesh is refined using optical flow (Section 5.1) to generate temporally consistent and drift bounded textures for each frame. Based on patches extracted from the textures, the regression function Φ (Section 4.3) predicts the local wrinkle shape (Section 5.2). After combining the local estimates into a global displacement map, the final high-resolution mesh is reconstructed and rendered on the GPU (Section 5.3).

5.1 Coarse Mesh Refinement

The coarse mesh produced by the global face tracker [Cao et al. 2014] is robust but typically not very accurate. This poses problems for our system since we rely on temporally stable textures for the detail regression. We therefore propose to refine the coarse mesh using optical flow [Lucas and Kanade 1981] as follows. We project the vertices $\{\mathbf{v}_i^t\}$ of the coarse mesh L^t at time t into the image I^t using the projection matrix P of the camera to compute the corresponding 2D pixel coordinates $\{\mathbf{u}_i^t\} = P(\{\mathbf{v}_i^t\})$. Given the 2D pixel coordinates $\{\tilde{\mathbf{u}}_i^{t-1}\}$ of the refined mesh $\tilde{L}^{t-1}$ from the previous frame, we aim to compute refined current positions $\{\tilde{\mathbf{u}}_i^t\}$ by solving the linear system

$$\mathbf{Z}(\tilde{\mathbf{u}}^t - \tilde{\mathbf{u}}^{t-1}) = \mathbf{e}, \tag{5}$$

where the matrix $\mathbf{Z}$ and the vector $\mathbf{e}$ contain the spatial and temporal intensity gradient information in the surrounding region, respectively. We solve for $\tilde{\mathbf{u}}^t$ using the GPU implementation of Lukas-Kanade in OpenCV[1].

Lucas-Kanade flow requires only local information (we employ a 21×21 neighborhood), which makes it extremely fast to compute and thus well suited for our real-time application, but at the same time it is also prone to errors and outliers. We thus cannot trust the flow unconditionally and we filter outliers in two cases, (1) if the intensity values $I^{t-1}(\tilde{\mathbf{u}}_i^{t-1})$ and $I^t(\tilde{\mathbf{u}}_i^t)$ differ by more than a threshold ξ_α, and (2) if the Euclidean 2D distance between the flow estimated position $\tilde{\mathbf{u}}_i^t$ and the projected position $\mathbf{u}_i^t = P(\mathbf{v}_i^t)$ of the mesh vertex $\mathbf{v}_i^t$ is larger than a threshold ξ_β. The thresholds were set to $\xi_\alpha = 0.02$ and $\xi_\beta = 5$ pixels.

The positions of the outliers are recovered from trustworthy neighbors using moving least squares (MLS) [Schaefer et al. 2006] in

[1] http://opencv.org

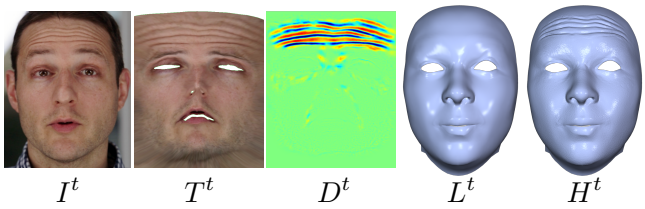

Figure 8: *From input image I^t the method first computes a texture map T^t using the coarse mesh L^t. The local regressor then predicts a displacement map D^t from this texture which is applied to the coarse mesh to produce the final high-fidelity mesh H^t.*

the 2D image domain. As distance metric we employ the geodesic distance, which is precomputed in the first frame. To verify if the recovered point is satisfactory, we check if the second condition mentioned above holds. If not we reset it to $\mathbf{u}_i^t$.

Finally we reconstruct the refined coarse mesh $\tilde{L}^t$ by adopting the post-processing step in Cao et al. [2014] to include the 2D mesh positions as constraints in the 3D fit. This will deform the vertices of the coarse mesh to be as close as possible to our refined locations within the limits imposed by the underlying prior. To allow for shapes outside of the prior, we employ fast Laplacian deformation following Li et al. [2013] using both the 2D landmarks provided by the global tracker and the refined 2D mesh positions as constraints.

This way of combining the local flow with the global face tracker bounds error accumulation or drift, which is a common problem for optical flow methods. The image textures $\{T^t\}$ computed from the refined meshes are therefore not only accurate but also robust, which is an essential property for the detail regression discussed in the next section.

5.2 Detail Regression

From the temporally stable texture T^t computed in the previous step, the detail regression will predict a displacement map D^t containing the medium-scale details missing in the low resolution face mesh $\tilde{L}^t$. As a first step we construct the wrinkle probability map from the first N frames as described in Section 4.3, where N is a user controlled parameter. Since the wrinkle map will determine the possible locations wrinkles may appear later on, it is important that these N frames contain as many facial wrinkles of the user as possible. This is achieved by performing a small set of extremal expressions and takes only a few seconds.

From the wrinkle probability map we determine where to retrieve patches $\{\mathbf{t}_i\}$ to estimate the local shape. The displacement patches $\{\mathbf{d}_i\}$ are estimated using the trained regressor as $\mathbf{d}_i = \Phi(\mathbf{t}_i)$. In practice we do not regress on the displacement directly but rather in PCA space on the coefficients $\mathbf{c}_i$ as

$$\mathbf{c}_i = \bar{\mathbf{c}} + \sum_{k=1}^{K} R^k(\mathbf{t}_i), \tag{6}$$

where $\bar{\mathbf{c}}$ is the average coefficients of all training samples.

This equation can be evaluated very efficiently and in parallel since the proposed one-level regressors do not exihibit any interdependies at runtime anymore. From the regressed coefficients the local displacements can be reconstructed via the PCA basis.

These reconstructed displacement patches $\{\mathbf{d}_i\}$ are then merged to generate the displacement texture D^t. Pixels in D^t which belong to multiple patches average the predicted displacements, and pixels which are not covered by any patch are smoothly filled in by solving a Poisson system on the texture domain. Since the patch coverage is fixed once the wrinkle map has been computed, we can pre-factorize the Poisson matrix to achieve real-time performance.

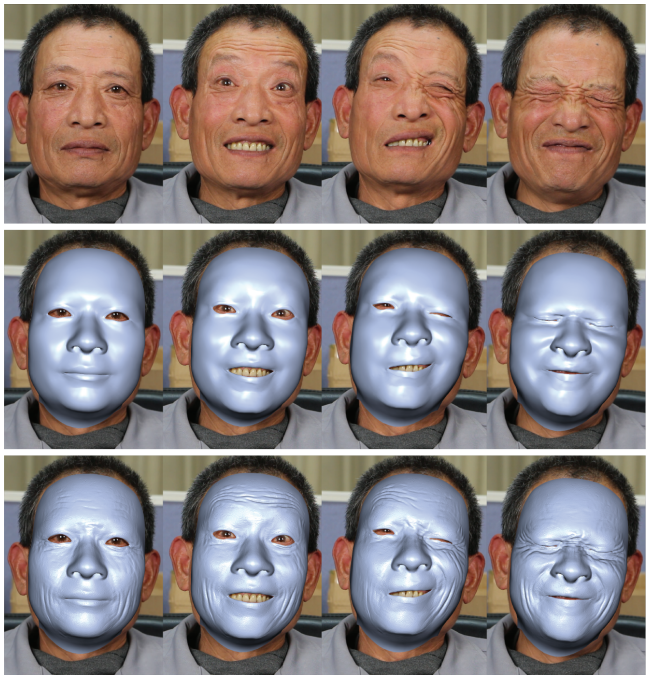

Figure 9: *The proposed system is both robust and accurate since it combines a global model to track the overall pose and expression with local models to reproduce medium scale details such as wrinkles, which are essential to convey expression and identity.*

5.3 High-Fidelity Model Reconstruction

The previous steps provide both low (e.g. overall expression) and medium (e.g. wrinkles) scale frequencies. To also add in the high frequency details (e.g. pores) we resort to a simplified version of mesoscopic augmentation [Beeler et al. 2010]. From the first frame we compute a mesoscopic map by filtering the image with a narrow band DoG filter (3x3 for the high and 9x9 for the low kernel), which we convert into the displacement map D^{HF} with a user defined scale parameter γ. We fixed $\gamma = 5$ in the results of this paper.

The final step is to augment the refined low-resolution mesh $\tilde{L}^t$ with the combined displacement map $(D^t + D^{HF})$ to produce the high-fidelity mesh H^t. This step is straightforward, since modern graphics hardware natively supports displacement mapping.

6 Results

The proposed method can track facial performances including medium scale details such as wrinkles in real-time. Being able to faithfully capture and reproduce these details adds greatly to the visual quality and perceived intensity of facial expressions as shown in Figure 9.

Our system is generic and runs on novel users without requiring any offline training or manual preprocessing steps. We demonstrate the robustness and flexibility of the system in Figure 10, where we show results for different people exhibiting a large variety in shape, location and intensity of skin wrinkles. From this huge variety it is obvious that a purely global model would probably not be able to faithfully represent all of these details and motivates the use of our coupled global/local model.

For the local model to work, temporally stable textures are required.

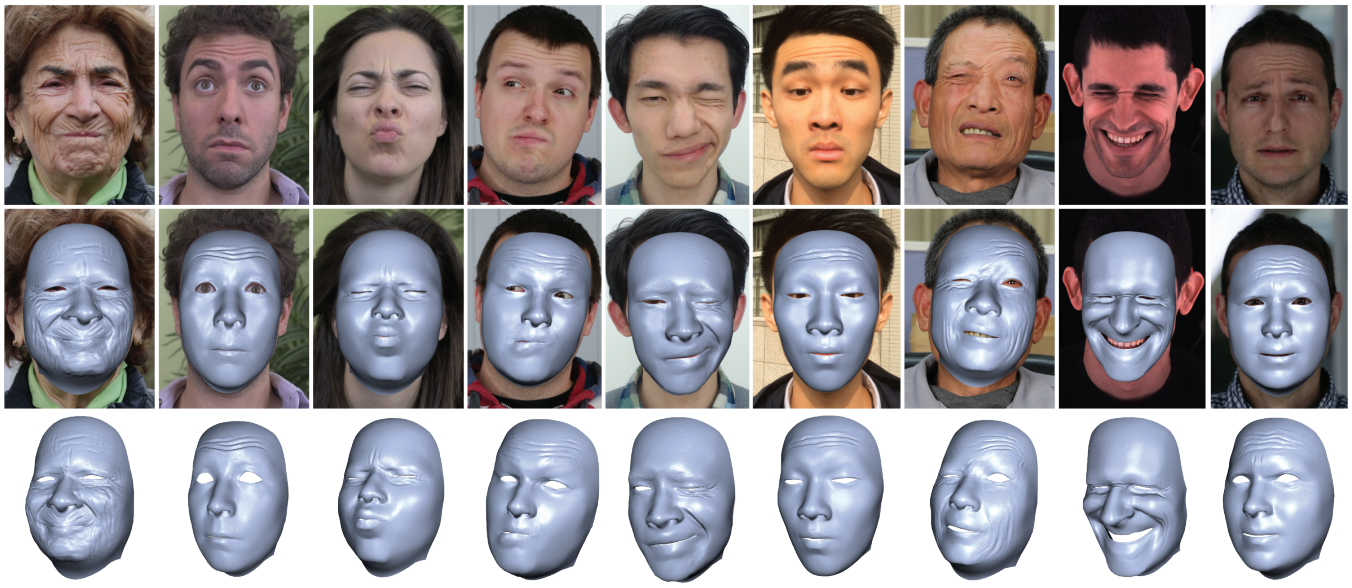

Figure 10: *Our system is generic and can accurately track novel users without any offline training or manual preprocessing steps. This figure shows users of varying age and different amounts of skin wrinkling.*

Our global tracker is based on the DDE tracker [Cao et al. 2014], which exhibits substantial tracking error due to the strong underlying face prior (Figure 11 (center)). Such a strong prior is required for robustness, but prevents accurate tracking. The proposed refinement step (Section 5.1) improves tracking substantially using optical flow as shown in Figure 11 (right). The price to pay is an increase in computation time. Our system is implemented mostly on the GPU using CUDA and runs at 18 fps on a standard desktop computer (Intel i7 3.6 GHz with NVidia Gtx980). Global tracking takes up ∼58% of the runtime, roughly half of this falls to the refinement using optical flow. The remaining ∼42% are spent by the local detail regressor.

When comparing our method with previous work on offline high-fidelity reconstruction, we found that our method produces results on par with other monocular reconstruction techniques [Garrido et al. 2013; Shi et al. 2014] but in real-time. Figure 12 (left) shows a comparison to the monocular system of Garrido et al. [2013] and on the right to the multi-view stereo system of Beeler et al. [2011]. While the global shape is not as accurate since we do not have depth constraints, the wrinkles produced by our detail regression are more pronounced.

Figure 13 shows a comparison to the method of Li et al. [2015] which employs texture synthesis in an offline preprocessing step to produce user-specific blendshapes with baked in wrinkles. Unlike our method, which regresses displacements only at the wrinkle location, they synthesize displacements for every pixel. This does not only require more time but will also introduce noise for small radii or break the structure of the wrinkles for larger neighborhoods. Increasing the regularization factor λ helps to reduce noise but will also attenuate the wrinkle intensity.

In summary, we have shown a wide variety of results demonstrating the versatility, robustness and accuracy of our generic facial performance capture system. Compared to previous offline techniques on high-fidelity performance capture, we produce results of similar visual quality but in real-time from a single input video.

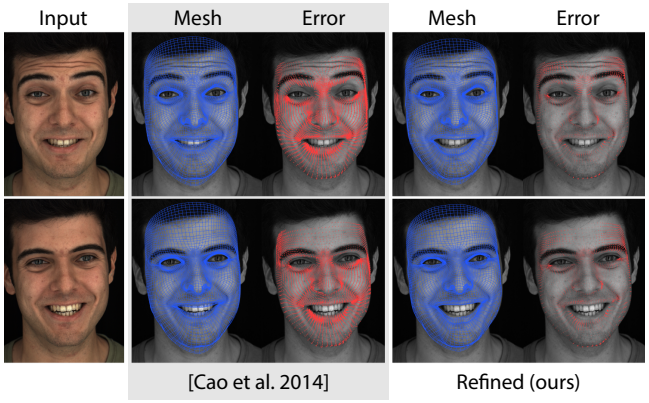

Figure 11: *When comparing the tracking error of Cao et al. [2014] with ours, we can clearly see the improvement achieved using optical flow. The red vectors show the difference to ground truth, tracked by the offline system of Beeler et al. [2011].*

7 Conclusion

In this work we present the first method capable of capturing facial performances in real-time at high fidelity, including medium scale details such as wrinkles. Combining real-time facial performance tracking with high-fidelity reconstruction opens up a variety of new applications, ranging from casual home users who, for example, wish to control a game character that resembles their own likeness, to commercial applications where, for example, a customer can apply and view virtual makeup, to production applications, such as virtual production where the director can observe and assess an actor's performance transferred to a digital creature as he is filming.

The proposed method is both robust and accurate since it combines a global face prior to track the overall shape with local regressors to produce the local detail. In doing so we present a number of novel technical contributions, including (1) a generic model that describes local detail on human faces and (2) an automatic technique to extract them such that they can be applied at runtime without interference, and (3) a novel local regressor that leverages the structure

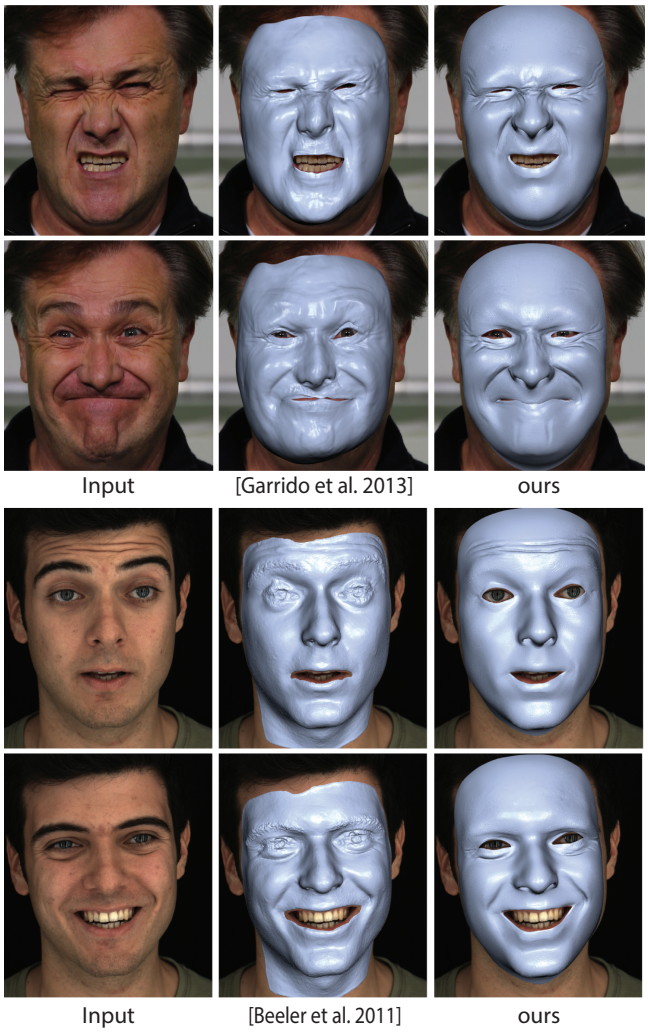

Figure 12: *Compared to previous offline techniques, such as monocular [Garrido et al. 2013] or multi-view systems [Beeler et al. 2011], the proposed method produces results of similar quality but in real-time.*

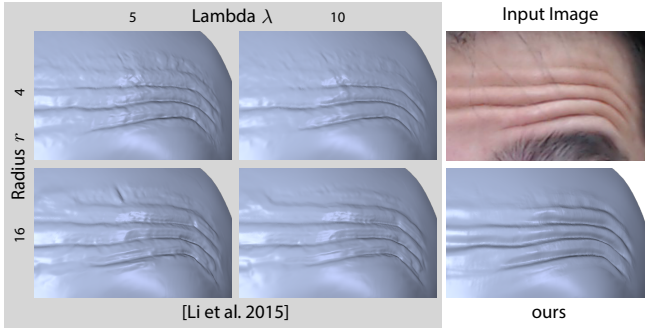

Figure 13: *The method of Li et al. [2015] employs texture synthesis to construct user specific blendshapes in an offline pre-processing step. Applying texture synthesis to every pixel in the displacement map is not only slow but also introduces noise for small radii r. Increasing λ reduces noise but attenuates detail intensity and increasing the radius breaks the structure of the wrinkles.*

Figure 14: *Strong illumination changes can cause inaccurate wrinkles if such illumination is not present in the training data. This split-screen shows the same expression with directional light from the front (left) and from above (right). The resulting wrinkles (also shown in split-screen) essentially slide along the surface to follow the shadow.*

present in the data to achieve real-time performance.

Our method has a few limitations which open up for future research. The optical flow employed to improve tracking accuracy of the global model can fail when there is fast motion or motion blur. While any introduced error will be bounded by the global tracker and thus cannot cause the system to diverge, it might lead to a subtle local shift in tracked texture. This could potentially be addressed by relating the current frame back to the neutral expression, but so far we have not noticed a degradation of the final results due to this. Another source of error for the flow would be changing illumination. Changing illumination and especially hard shadows can also impact the local regressor. If the illumination during runtime differs substantially from the illumination during wrinkle-training, the resulting wrinkles can be inaccurate. We demonstrate this drawback in Figure 14, which shows a performance under a moving directional light. When the light comes from above, forehead wrinkles tend to be higher on the face than when the light comes from the front, as the wrinkles essentially slide on the face to follow the shadow. We expect that this problem would be alleviated by expanding the training data either with more samples or by synthetically relighting them. Another problem are (partial) occlusions, which will confuse the system. However, once the occluder is removed, it will quickly recover thanks to the robust global model (please see the accompanying video for a demonstration). Finally, with increasing quality and resolution of the capture hardware we might soon be able to also capture the temporal behaviour of mesoscopic detail such as pores with a consumer grade camera.

While most existing methods for facial capture so far aim for either high quality or speed, we believe these two lines of research are going to converge eventually. The proposed method presents a substantial step towards this goal, providing real-time performance capture at high visual quality for the first time. The system requires only a single uncalibrated camera as input and is generic in that it does not require any offline training or manual steps for a novel user.

Acknowledgements

We would like to thank all of our actors, as well as Garrido et al. for providing datasets for comparison. Kun Zhou was supported in part by NSFC (No. 61272305) and the National Program for Special Support of Eminent Professionals of China.

References

BEELER, T., BICKEL, B., SUMNER, R., BEARDSLEY, P., AND GROSS, M. 2010. High-quality single-shot capture of facial

geometry. *ACM Trans. Graphics (Proc. SIGGRAPH)*.

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. Graphics (Proc. SIGGRAPH) 30*, 75:1–75:10.

BEELER, T., BRADLEY, D., ZIMMER, H., AND GROSS, M. 2012. Improved reconstruction of deforming surfaces by cancelling ambient occlusion. In *ECCV*. 30–43.

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

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. Graphics (Proc. SIGGRAPH)*, 33.

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

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

CAO, X., WEI, Y., WEN, F., AND SUN, J. 2012. Face alignment by explicit shape regression. In *IEEE CVPR*, 2887–2894.

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

CAO, C., HOU, Q., AND ZHOU, K. 2014. Displaced dynamic expression regression for real-time facial tracking and animation. *ACM Trans. Graphics (Proc. SIGGRAPH) 33*, 4, 43:1–43:10.

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

CHEN, Y.-L., WU, H.-T., SHI, F., TONG, X., AND CHAI, J. 2013. Accurate and robust 3d facial capture using a single rgbd camera. In *ICCV*.

DUTREVE, L., MEYER, A., AND BOUAKAZ, S. 2011. Easy acquisition and real-time animation of facial wrinkles. *Comput. Animat. Virtual Worlds 22*, 2-3, 169–176.

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

GARRIDO, P., VALGAERTS, L., WU, C., AND THEOBALT, C. 2013. Reconstructing detailed dynamic face geometry from monocular video. In *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, vol. 32, 158:1–158:10.

GHOSH, A., FYFFE, G., TUNWATTANAPONG, B., BUSCH, J., YU, X., AND DEBEVEC, P. 2011. Multiview face capture using polarized spherical gradient illumination. *ACM Trans. Graphics (Proc. SIGGRAPH Asia) 30*, 6, 129:1–129:10.

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

KLAUDINY, M., AND HILTON, A. 2012. High-detail 3d capture and non-sequential alignment of facial performance. In *3DIMPVT*.

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

LI, J., XU, W., CHENG, Z., XU, K., AND KLEIN, R. 2015. Lightweight wrinkle synthesis for 3d facial modeling and animation. *Computer-Aided Design 58*, 0, 117–122.

LUCAS, B. D., AND KANADE, T. 1981. An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th IJCAI*, 674–679.

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*, 183–194.

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. Graphics (Proc. SIGGRAPH Asia) 27*, 5, 121.

RHEE, T., HWANG, Y., KIM, J. D., AND KIM, C. 2011. Real-time facial animation from live video tracking. In *Proc. SCA*, 215–224.

SCHAEFER, S., MCPHAIL, T., AND WARREN, J. 2006. Image deformation using moving least squares. *ACM Trans. Graphics 25*, 3, 533–540.

SHI, F., WU, H.-T., TONG, X., AND CHAI, J. 2014. Automatic acquisition of high-fidelity facial performances using monocular videos. *ACM Trans. Graphics (Proc. SIGGRAPH Asia) 33*.

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

SUWAJANAKORN, S., KEMELMACHER-SHLIZERMAN, I., AND SEITZ, S. M. 2014. Total moving face reconstruction. In *ECCV*.

VALGAERTS, L., WU, C., BRUHN, A., SEIDEL, H.-P., AND THEOBALT, C. 2012. Lightweight binocular facial performance capture under uncontrolled lighting. *ACM Trans. Graphics (Proc. SIGGRAPH Asia) 31*, 6.

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. Real-time performance-based facial animation. *ACM Trans. Graphics (Proc. SIGGRAPH) 30*, 4, 77:1–77:10.

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