Reconstructing Detailed Dynamic Face Geometry from Monocular Video

Pablo Garrido[1] * Levi Valgaerts[1] † Chenglei Wu[1,2] ‡ Christian Theobalt[1] §
[1]Max Planck Institute for Informatics [2]Intel Visual Computing Institute

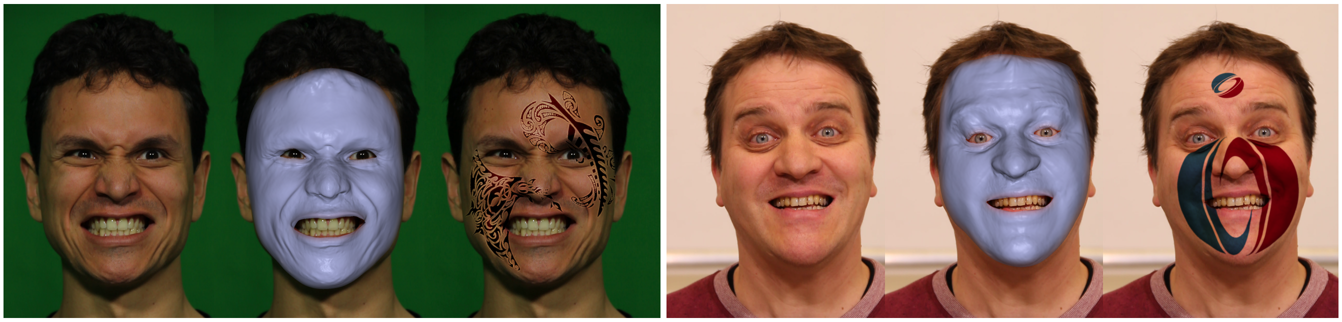

Figure 1: *Two results obtained with our method.* Left: *The input video.* Middle: *The tracked mesh shown as an overlay.* Right: *Applying texture to the mesh and overlaying it with the input video using the estimated lighting to give the impression of virtual face make-up.*

Abstract

Detailed facial performance geometry can be reconstructed using dense camera and light setups in controlled studios. However, a wide range of important applications cannot employ these approaches, including all movie productions shot from a single principal camera. For post-production, these require dynamic monocular face capture for appearance modification. We present a new method for capturing face geometry from monocular video. Our approach captures detailed, dynamic, spatio-temporally coherent 3D face geometry without the need for markers. It works under uncontrolled lighting, and it successfully reconstructs expressive motion including high-frequency face detail such as folds and laugh lines. After simple manual initialization, the capturing process is fully automatic, which makes it versatile, lightweight and easy-to-deploy. Our approach tracks accurate sparse 2D features between automatically selected key frames to animate a parametric blend shape model, which is further refined in pose, expression and shape by temporally coherent optical flow and photometric stereo. We demonstrate performance capture results for long and complex face sequences captured indoors and outdoors, and we exemplify the relevance of our approach as an enabling technology for model-based face editing in movies and video, such as adding new facial textures, as well as a step towards enabling everyone to do facial performance capture with a single affordable camera.

CR Categories: I.3.7 [COMPUTER GRAPHICS]: Three-Dimensional Graphics and Realism; I.4.1 [IMAGE PROCESSING]: Digitization and Image Capture—Scanning; I.4.8 [IMAGE PROCESSING]: Scene Analysis;

Keywords: Facial Performance Capture, Monocular Tracking, Temporally Coherent Optical Flow, Shading-based Refinement

Links: DL PDF WEB VIDEO

1 Introduction

Optical performance capture methods can reconstruct faces of virtual actors in videos to deliver detailed dynamic face geometry. However, existing approaches are expensive and cumbersome as they can require dense multi-view camera systems, controlled light setups, active markers in the scene, and recording in a controlled studio (Sec. 2.2). At the other end of the spectrum are computer vision methods that capture face models from monocular video (Sec. 2.1). These captured models are extremely coarse, and usually only contain sparse collections of 2D or 3D facial landmarks rather than a detailed 3D shape. Recently, Valgaerts et al. [2012] presented an approach for detailed performance capture from binocular stereo. However, 3D face models of a quality level needed for movies and games cannot yet be captured from monocular video.

In this work, we aim to push the boundary and application range further and move towards monocular video. We propose a new method to automatically capture *detailed* dynamic face geometry from *monocular* video filmed under general lighting. It fills an important algorithmic gap in the spectrum of facial performance capture techniques between expensive controlled setups and low-quality monocular approaches. It opens up new application possibilities for professional movie and game productions by enabling facial performance capture on set, directly from the primary camera. Finally, it is a step towards democratizing face capture technology for everyday users with a single inexpensive video camera.

A 3D face model for a monocular video is also a precondition for many relevant video editing tasks (Sec. 2.3). Examples in-

*e-mail: pgarrido@mpi-inf.mpg.de
†e-mail: valgaerts@mpi-inf.mpg.de
‡e-mail: chenglei@mpi-inf.mpg.de
§e-mail: theobalt@mpi-inf.mpg.de

clude video face transfer and face replacement [Vlasic et al. 2005; Alexander et al. 2009], facial animation retiming for dubbing [Dale et al. 2011], or face puppeteering [Kemelmacher-Shlizerman et al. 2010; Li et al. 2012]. For these results, a tracked geometry model of moderate shape detail was sufficient, but even then, substantial manual work is unavoidable to obtain a 3D face model that overlays sufficiently with the video footage. To achieve a higher quality of edits on more general scenes, and to show advanced edits such as relighting or virtual make-up, we require much higher shape detail to be captured from a single video.

Our approach relies on several algorithmic contributions that are joined with state-of-the-art 2D and 3D vision and graphics techniques adapted to monocular video. In a one-time preparatory step, we create a personalized blend shape model for the captured actor by transferring the blend shapes of a generic model to a single static 3D face scan of the subject (Sec. 4). This task is the only manual interaction in our technique. Then, in the first step of our automatic algorithm, we track a sparse set of 2D facial features throughout the video by adapting a probabilistic face tracking method that is regularized by a parametric 3D face model, learned once from a training set (Sec. 5). To increase the accuracy of the 2D landmark localization, we introduce a new feature correction scheme that uses optical flow for tracking correction relative to key poses of the face. These stabilizing key poses are automatically selected by splitting the sequence between frames with similar facial appearance. After 2D landmark tracking, we obtain the blend shape and pose parameters of the personalized 3D face model by solving a constrained quadratic programming problem at every frame (Sec. 6). To further refine the alignment of the face model to the video, a non-rigid, temporally coherent geometry correction is performed using a novel multi-frame variational optical flow approach (Sec. 7). Finally, a shape-from-shading-based shape refinement approach reconstructs fine scale geometric face detail after estimating the unknown incident lighting and face albedo (Sec. 8).

We emphasize the simplicity and robustness of our lightweight and versatile performance capture method. We do not claim to achieve a higher reconstruction quality than multi-view methods, but we think that our approach is one of the first to capture long sequences of expressive face motion for scenarios where none of these other methods are applicable. Our method requires only a little user intervention during blend shape model creation and initial alignment, and tracking itself is fully automatic. As an additional benefit, our tracker estimates blend shape parameters that can be used by animators in standard software tools, which is an important feature also advocated in previous 3D facial performance capture work [Weise et al. 2011]. We show qualitative and quantitative results on several expressive face sequences captured under uncontrolled lighting, both indoors and outdoors. Our approach compares favorably to the recent binocular performance capture method of Valgaerts et al.[2012], and even performs better for certain aspects. Finally, we show a proof-of-concept application of advanced video editing by applying virtual face textures to video.

2 Related Work

2.1 Monocular Face Tracking

Many monocular methods for tracking the motion of sparse 2D or 3D facial feature sets have been developed. These often represent the face as a parametric 2D or 3D shape model, which is matched against features in the video, e.g. [Li et al. 1993; Black and Yacoob 1995; Essa et al. 1996; DeCarlo and Metaxas 1996]. In this class of algorithms, methods using variants of active appearance models are very popular [Cootes et al. 2001; Xiao et al. 2004]. Such models are linear approximations to the non-rigid deformation of sparse 2D and 3D feature sets, which are learned from labeled training data. Recent work trained regression forests to find very sparse face features [Dantone et al. 2012]. Model-based optical flow has also been applied for monocular non-rigid tracking and built up from a coarse face template [Brand and Bhotika 2001].

Chuang et al. [2002] track a coarse blend shape model, albeit with actively placed markers on the face, and coarsely map facial motion to a 3D avatar. Chai et al. [2003] also extract animation parameters from coarse tracked landmarks and map them to an avatar. Kemelmacher-Shlizerman et al. [2010] use sparse feature tracking to puppet a target character from an input face video. The result is a coarse sequence of similar faces retrieved from a video via feature matching. Li et al. [2012] propose a variant of this retrieval idea that produces temporally smoother results. The state-of-the-art sparse, monocular face tracker of Saragih et al. [2011] combines statistical shape and appearance models, but falls short of the accuracy we aim for. We build additional innovations on top of this tracker to achieve a sufficient level of accuracy and stability. In concurrent work [Cao et al. 2013; Li et al. 2013; Bouaziz et al. 2013], real-time monocular face trackers have been proposed that are either based on a trained shape regression model for video or on a run time shape correction strategy for combined depth and video data. All these works use a personalized blend shape model of some sort, but their application is limited to face retargeting. Instead, we move outside the blend shape domain by recovering a more detailed and expressive face geometry and we show accurate video overlays.

2.2 Dense 3D Facial Performance Capture

Most algorithms for dense detailed 3D facial performance capture use complex and dense camera, motion capture, or scanner systems, or rely on sophisticated lighting and a special studio [Pighin and Lewis 2006]. Some methods use dense camera sets to track markers or invisible make-up [Williams 1990; Guenter et al. 1998; Furukawa and Ponce 2009; Bickel et al. 2007]. Combining marker-based motion capture with blending between static face scans enables synthesis of detailed moving faces [Huang et al. 2011]. Other 3D methods track shape templates from dynamic active 3D scanner data [Zhang et al. 2004; Wang et al. 2004; Weise et al. 2007].

Image-based approaches help to overcome the limitations in shape detail and tracking accuracy that purely geometric and scanner-based methods still have. Template-based methods fit a deformable shape model to images of a face [DeCarlo and Metaxas 1996; Pighin et al. 1999; Blanz et al. 2003]. These methods yield spatio-temporally coherent reconstructions, but the captured geometry is coarse. High-quality facial performances can be obtained by combining passive stereo and mesh tracking [Borshukov et al. 2003; Bradley et al. 2010; Beeler et al. 2011; Valgaerts et al. 2012]. Some commercial systems also fall into this category, e.g. the MOVA [1] system or the approach by DepthAnalysis [2]. Pore-level skin detail can be reconstructed by recording under controlled illumination and employing photometric cues [Alexander et al. 2009; Wilson et al. 2010]. The approaches mentioned here produce high-quality results, but most require complex, expensive setups and would be inapplicable for the use cases motivating our work.

The first steps toward more lightweight setups have been taken. Weise et al. [2009; 2011] capture blend shape parameters in real-time with a Kinect. Similar to our work, the ability to obtain meaningful blend shape parameters is considered very important. However, their goal is a coarse control of avatars in real-time and not a highly detailed face reconstruction. Valgaerts et al. [2012] capture detailed facial performances under uncontrolled lighting using

[1] www.mova.com

[2] www.depthanalysis.com

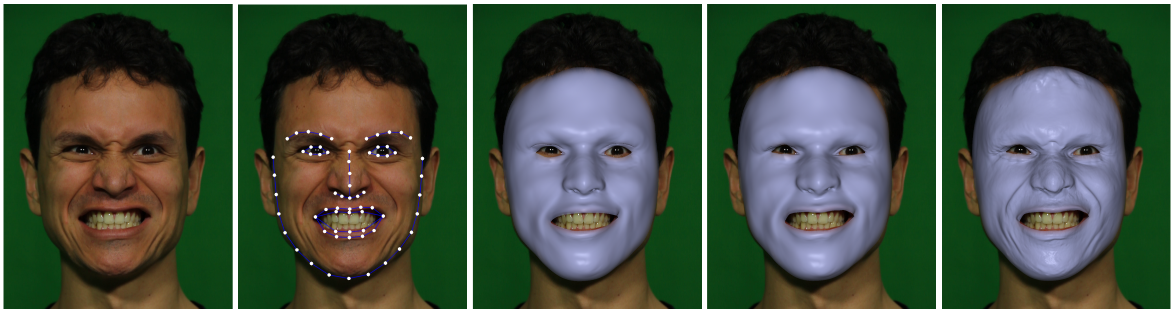

Figure 2: *Algorithm overview:* Left to right: *(a) Input video frame, (b) sparse feature tracking* **(Sec. 5)***, (c) expression and pose estimation using a blend shape model* **(Sec. 6)***, (d) dense expression and pose correction* **(Sec. 7)***, (e) shape refinement* **(Sec. 8)***.*

a single binocular stereo camera. In this paper, we go one step further and capture detailed space-time-coherent face geometry from a single video.

2.3 Dynamic Monocular Face Reconstruction

Many methods for monocular dense 3D reconstruction were developed to enable advanced video editing. Blanz et al. [2003] fit a PCA face model, learned from a large database of 3D scans, to video and perform simple editing tasks. However, fine face detail such as wrinkles and laugh lines cannot be recovered with their approach, and the tracked faces do not always overlap exactly with the video. Vlasic et al. [2005] introduce multilinear face models that learn separate dimensions for facial expressions and identity from large databases of face scans and use them to track coarse-to-medium scale, dynamic face geometry for face transfer in videos exhibiting very little head motion. Dale et al. [2011] use the tracker from [Vlasic et al. 2005], but require a 3D model of much higher shape quality to enable faithful face replacement and video retiming in more unconstrained and general videos. The multilinear tracker does not meet these requirements and considerable manual correction in several frames is needed. In the Digital Emily project [Alexander et al. 2009], a commercial software by Image Metrics [3] was used to capture face animation parameters of a blend shape rig that matches the actor in video. The high reconstruction quality and exact alignment of the face to the video required considerable manual work by an artist. Thus, so far, only facial geometry of moderate quality can be captured in a monocular setting, and this requires substantial manual intervention. In contrast, our new method captures spatio-temporally coherent, dynamic face geometry at high quality and with minimal manual interaction. It succeeds on sequences filmed in general environments and for expressive faces and head motion, and it paves the way for high quality advanced face editing in movies.

3 Method Overview

Our method uses as input a single video of a face captured under unknown lighting. It is composed of four main computational steps:

S0 **Personalized face model creation (Sec. 4).** We construct a customized parametric 3D blend shape model for every actor, which is used to reconstruct all sequences starring that actor.

S1 **Blend shape tracking (Sec. 5 and Sec. 6).** We track 2D image features throughout the monocular video by combining sparse facial feature tracking with automatic key frame selection and reliable optical flow, see Fig. 2 (b). From the established sparse feature set, we estimate a global 3D transformation (head pose) and a set of model parameters (facial expression) for the blend shape model, see Fig. 2 (c).

S2 **Dense tracking correction (Sec. 7).** Next, we improve the facial expression and head pose obtained from sparse blend shape tracking by computing a temporally coherent and dense motion field in video and correcting the facial geometry to obtain a more accurate model-to-video alignment, see Fig. 2 (d).

S3 **Dynamic shape refinement (Sec. 8).** In a final step, we reconstruct fine-scale, time-varying facial detail, such as wrinkles and folds. We do this by estimating the unknown lighting and exploiting shading for shape refinement, see Fig. 2 (e).

Notation. A frame in the monocular video corresponding to time stamp t will be denoted by f^t, with f^{t_0} being the starting frame. We reconstruct a spatio-temporally coherent sequence of triangular face meshes M^t, consisting of a fixed set of n vertices with Euclidean coordinates $\boldsymbol{X}^t$ and their connecting edges. The outcome of the subsequent computational steps in our algorithm are the *tracked mesh* M_{b}^t (S1), the *corrected mesh* M_{c}^t (S2) and the final *refined mesh* M_{r}^t (S3), all based on the same vertex set and connectivity.

4 A Personalized Blend Shape Model

We use a *blend shape model* as a parametric morphable 3D representation of the face. Blend shapes [Pighin and Lewis 2006] are additive deformations on top of a neutral face shape that are able to span a large variation of natural expressions and are widely used in facial animation. If we denote by $\boldsymbol{n} \in \mathbb{R}^{3n}$ the neutral shape containing the coordinates of the n vertices of a face mesh in rest, a new facial expression $\boldsymbol{e}$ can be obtained by the linear combination:

$$\boldsymbol{e}\left(\beta_j\right) = \boldsymbol{n} + \sum_{j=1}^{k} \beta_j \, \boldsymbol{b}_j \ , \tag{1}$$

where $\boldsymbol{b}_j \in \mathbb{R}^{3n}$, with $1 \leq j \leq k$, are the blend shape displacements and $0 \leq \beta_j \leq 1, \forall j$ are the k *blending weights*.

We create an actor specific face model by starting from a generic, artist-created, professional blend shape model [4] ($k = 78$) and performing a non-rigid registration of the neutral shape to a binocular stereo reconstruction [Valgaerts et al. 2011] of the actor's face in rest. Please note that any generic blend shape model preferred by an artist and any laser scanning or image-based face reconstruction method [5] can be used instead. Registration is based on manually

[3] www.image-metrics.com

[4] obtained from Faceware Technologies www.facewaretech.com

[5] www.facegen.com

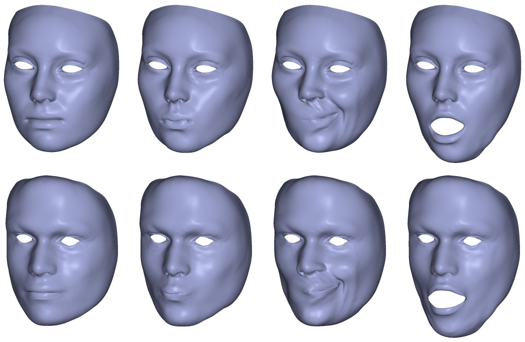

Figure 3: *Personalized blend shape models.* Top: *Three poses from the generic model, including the neutral pose.* Bottom: *The same dimensions after transferring them to the target actor of Fig. 2.*

matching 29 3D landmarks on the eyes, nose and mouth, followed by a global correspondence search and Laplacian regularized shape deformation [Sorkine 2005]. Once the neutral shape is registered, the blend shapes of the generic model are transferred using the same procedure. The obtained face models have a person specific shape, but the same semantic dimensions over all actors. Although our straightforward registration approach has proven sufficient for our application, additional person-specific semantics can be included by using extra scans of different expressions [Li et al. 2010]. Since all personalized blend shape models are derived from the same generic model, they share the same number of vertices (200k) and triangulation (henceforth shared by all meshes in this paper). Fig. 3 shows four corresponding poses for the generic model and the derived personalized model for the actor in Fig. 2 (more examples are shown in the supplementary material). Note that the produced blend shape models lack high frequency shape detail, such as wrinkles.

An alternative parametric representation is a PCA model, which removes possible linear dependencies between the blend shapes. As opposed to uncontrolled PCA dimensions, however, blend shapes are semantically meaningful and correspond to the localized regions of influence on the face that artists are used to work with.

5 2D Facial Feature Tracking

A sparse 2D facial feature tracker serves as the base of our method, but its performance falls short of our accuracy requirements. To meet our needs, we augment it with a new optical flow-based correction approach using automatically selected key frames.

5.1 Sparse Feature Tracking

Our system utilizes a non-rigid face tracking algorithm proposed by Saragih [2011], which tracks a sparse set of $m = 66$ consistent facial feature points, such as the eyes, brows, nose, mouth and face outline, see Fig. 2 (b). The tracking algorithm is based on a 3D point distribution model (PDM), which linearly models non-rigid shape variations around a set of 3D reference locations $\bar{\boldsymbol{X}}_i$, $i = 1, \ldots, m$, and composes them with a global rigid transformation:

$$\boldsymbol{x}_{\mathrm{f},i} = sPR\left(\bar{\boldsymbol{X}}_i + \Phi_i \boldsymbol{q}\right) + \boldsymbol{t} \quad \text{with} \quad P = \left[\begin{smallmatrix}1 & 0 & 0\\ 0 & 1 & 0\end{smallmatrix}\right] . \qquad (2)$$

Here, $\boldsymbol{x}_{\mathrm{f},i}$, $1 \leq i \leq m$, denotes the estimated 2D location of the i-th facial feature in the image and P the orthogonal projection matrix. The PDM parameters are the scaling factor s, the 3D rotation matrix R, the 2D translation vector $\boldsymbol{t}$, and the non-rigid deformation parameters $\boldsymbol{q} \in \mathbb{R}^d$, where $d = 24$ is the dimension of the PDM model. Further, $\Phi_i \in \mathbb{R}^{3\times d}$ denotes a previously learned submatrix of the basis of variation pertaining to the i-th feature. To find the most likely feature locations, the algorithm first calculates a response map for each landmark by local feature detectors trained to differentiate aligned from misaligned locations, and then combines the local detectors in an optimization step which enforces a global prior over their joint motion. Both the trained PDM model and landmark classifiers where provided to us by the authors.

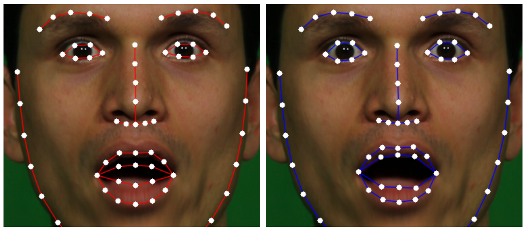

Figure 4: *Facial features before correction (*Left*) and after correction (*Right*).*

5.2 Automatic Key Frame Selection

There may be localization errors in the detected features, especially for expressions on which the tracker was not trained. Tab. 1 quantifies this effect by listing the mean distance of the detected features from their manually annotated ground truth locations for a selection of expressions from the experiments in Sec 9. To account for such errors, we correct the feature locations using accurate optical flow between *key frames*, i.e., frames for which the localization of the facial features detected by the face tracker is considered reliable.

Appearance descriptor. Key frames are selected by comparing the facial appearance of each frame with the appearance of a reference frame that has well localized features, such as a frame of a neutral pose. In our application, we assume that the starting frame f^{t_0} depicts the actor in rest and we choose it as a reference. We first align all frames in the sequence to the first frame using a 2D affine transformation that maps at best the set of detected features onto the reference shape. Next, we consider three rectangular regions of fixed size around the eyes and mouth, and split them into several tiles for which the appearance is represented as a histogram of local binary patterns (LBP). LBPs are very effective for expression matching and identification tasks [Ahonen et al. 2006] and encode the relative brightness around a pixel by assigning a binary value to each neighboring pixel, depending on whether it is brighter or not. The result is an integer value between 0 and 2^l for each center pixel, where l is the size of a circular neighborhood. To increase the discriminative power of appearance matching, we combine a uniform LBP code for $l = 8$ [Kemelmacher-Shlizerman et al. 2010] with a non-uniform code for $l = 4$, resulting in an LBP histogram of 75 bins for each tile. Finally, we concatenate all histograms within a region of interest to a single descriptor H for the whole region.

Appearance matching. In a first pass, an initial set of key frames is selected as those frames in the sequence that are closest to the neutral expression according to the distance metric:

$$d_{\mathrm{app}}(f^{t_0}, f^t) = \sum_{i=1}^{3} d_{\chi^2}\left(H_i(f^{t_0}), H_i(f^t)\right) \quad , \qquad (3)$$

where d_{χ^2} is the chi-squared distance between two histograms and H_i the appearance descriptor for the eyes and mouth regions. The amount of initial key frames is chosen as 2.5% of the sequence length, which can be thought of as a probability estimate of finding

Table 1: *Key frame-based feature correction: Mean distance (in pixels) of the* 66 *tracked facial feature points to their manually annotated ground truth location for a selection of expressions from the sequences shown in the experiments of Sec. 9.*

Sequence	Feature Tracking	Key Frame Correction
11 expressions of seq. 1 (Fig. 7)	5.38 ± 1.47	3.83 ± 1.05
11 expressions of seq. 2 (Fig. 10)	6.72 ± 1.44	4.60 ± 0.70
10 expressions of seq. 3 (Fig. 10)	6.36 ± 1.65	4.13 ± 0.88
Overall	6.15 ± 1.52	4.19 ± 0.88
Overall, only mouth and eyes	7.24 ± 2.22	4.35 ± 1.46

a neutral expression in the video and at the same time corresponds to an average inter-key-frame-distance of 40 frames. In a second pass, we select clips between consecutive key frames with a length of more than 50 and divide these by adding more key frames. These in-between key frames are selected in the same way using the distance metric (3), but this time we use an appearance descriptor H for a small region around each of the m detected facial features. Unlike the initial key frames, in-between key frames may have non-neutral expressions since we only seek similar texture patterns around facial features and not within whole facial regions. The division threshold of 50 frames is chosen to limit drift by optic flow (see Sec 5.3) over longer clips. In our experiments, the resulting average key frame distance was 22, with an average maximum of almost 90.

5.3 Optical Flow-based Feature Correction

If we assume that we have a key frame at time $t = T$, we compute the feature locations at times $t > T$ as:

$$\boldsymbol{x}_i^t = \lambda_i \, \boldsymbol{x}_{\mathrm{f},i}^t + (1 - \lambda_i) \, \boldsymbol{x}_{\mathrm{o},i}^t \quad \text{for} \quad 1 \leq i \leq m \ , \tag{4}$$

where $0 \leq \lambda_i \leq 1$ is a weighting factor. In this expression, $\boldsymbol{x}_\mathrm{f}^t$ is the feature position (2) obtained by the *facial feature tracker* (Sec. 5.1) at time t, and $\boldsymbol{x}_\mathrm{o}^t$ is the feature location estimated by *optical flow*:

$$\boldsymbol{x}_\mathrm{o}^t = \boldsymbol{x}^T + \sum_{T \leq i < t} \boldsymbol{w}^i \ . \tag{5}$$

Here, $\boldsymbol{x}^T$ denotes the feature position in the key frame f^T and $\boldsymbol{w}^t$ is the forward optical flow vector from t to $t+1$ in $\boldsymbol{x}_\mathrm{o}^t$. Optical flow is estimated in a variational framework by minimizing an energy consisting of a data and a smoothness term similar to those of Eq. (10) and Eq. (11). In practice, we also compute the backward optical flow from $t+1$ to t and use it to back-trace the feature position from the next key frame. The influence of the forward and backward optical flows is varied smoothly over time, with the forward (backward) flow having more weight near the previous (next) key frame. This avoids an accumulation of drift errors and ensures smooth feature trajectories at key frames. A related key frame approach for dense tracking was adopted by Beeler et al. [2011].

Improvements in feature location after optical flow-based feature correction are clearly noticeable for very expressive regions, such as the mouth and the eyes in Fig. 4. Tab. 1 further shows that the overall feature localization improves after our correction step.

6 Coarse Expression and Pose Estimation

We now align the 3D blend shape model to the 2D sparse feature locations found in each frame: We solve an optimization problem to find the pose and facial expression parameters of the 3D blend shape model such that it reprojects onto the tracked 2D features. This is done in three steps.

6.1 Coupling the 2D and 3D Model

To couple the sparse 2D features to their corresponding 3D vertices on the blend shape model, we render an OpenGL shaded neutral face in front of a black background and estimate the feature locations with the tracker of Sec. 5.1. After minimal manual correction of the detected features (see the supplementary material), we establish the fixed set of corresponding 3D feature vertices, henceforth denoted as F. This step only needs to be completed once for the generic model since all personalized models are derived from it.

6.2 Expression Estimation

Given a set of facial feature locations $\boldsymbol{x}_i^t$, $1 \leq i \leq m$ estimated in the current frame f^t, and a personalized blend shape model $\boldsymbol{e}\left(\beta_j\right)$, it is our task to estimate the current facial expression in terms of the blending weights β_j^t, $1 \leq j \leq k$. This expression transfer problem can be formulated in a constrained least squares sense as:

$$\min_{\beta_j^t} \sum_i^m \left\| \left(s^t R^t P^\top \boldsymbol{x}_i^t + \boldsymbol{t}^t \right) - \boldsymbol{X}_{F,i}\left(\beta_j^t\right) \right\|^2 , \tag{6}$$

$$\text{s.t.} \quad 0 \leq \beta_j^t \leq 1 \quad \text{for} \quad 1 \leq j \leq k \ , \tag{7}$$

where $\boldsymbol{X}_{F,i} \in F$ are the coordinates of the feature vertices of the blend shape model. The orthogonal weak perspective projection matrix P is the same as in Eq. (2) and s^t, $R^t \in \mathbb{R}^{3\times3}$ and $\boldsymbol{t}^t \in \mathbb{R}^3$ denote a scaling factor, a rotation matrix, and a translation vector which align the reprojected feature locations with the feature vertices of the blend shape model in a weak perspective setting. Since the alignment transformations are unknown, we solve the above quadratic programming problem iteratively: First we optimize for $\{s, R, \boldsymbol{t}\}^t$ using a current estimate for the blending weights, after which we solve for β_j^t in a second step keeping the transformations fixed. We terminate when the change in β_j^t falls below a threshold.

Solving for the transformations. Finding the least squares solution of $\{s, R, \boldsymbol{t}\}^t$ to expression (6) for a constant set of blending weights is equivalent to aligning two 3D point sets, which can be solved in closed form by SVD [Arun et al. 1987].

Solving for the blending weights. Once the alignment transformations have been computed, we search for an optimal combination of the linear weights β_j^t which minimizes the difference in shape between the point sets $\left(s^t R^t P^\top \boldsymbol{x}_i^t + \boldsymbol{t}^t\right)$ and $\boldsymbol{X}_{F,i}\left(\beta_j^t\right)$, $1 \leq i \leq m$, subject to the box constraints (7). By rewriting the blend shape model (1) as:

$$\boldsymbol{e}\left(\beta_j\right) = \left(1 - \sum_{j=1}^k \beta_j\right) \boldsymbol{n} + \sum_{j=1}^k \beta_j \left(\boldsymbol{n} + \boldsymbol{b}_j\right) \ , \tag{8}$$

and defining $\beta_0 = 1 - \sum_{i=j}^k \beta_j$, we obtain an instance of a convex quadratic programming problem with box constraints and a linear equality constraint. This can be solved efficiently by methods based on sequential minimal optimization [6] [Platt 1998]. As opposed to the alignment step, we found experimentally that the blending weight optimization is more robust if only performed over the X- and Y-coordinates, so for this step we discard depth information.

6.3 3D Pose Estimation

To retrieve the head pose under a full perspective projection, we update the positions of the 3D feature vertices in F using the computed blending weights, and feed them together with the tracked 2D

[6] http://cmp.felk.cvut.cz/~xfrancv/libqp/html/

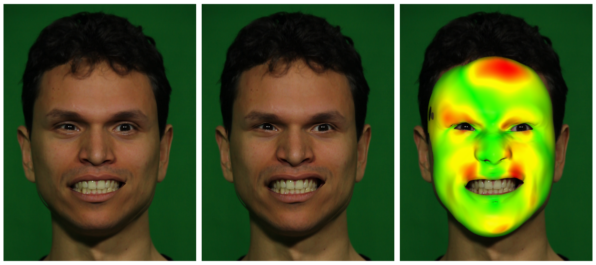

Figure 5: *Dense expression and pose correction.* Left: *Overlay of the tracked blend shape model of Fig. 2 (c), textured with the starting frame.* Middle: *Textured overlay of the tracking-corrected face mesh of Fig. 2 (d). This synthetic frame is closer to the target frame in Fig. 2 (a).* Right: *Per-vertex correction color coded on the corrected mesh, where red means large correction and green means small correction.*

facial feature locations to a pose estimation algorithm [David et al. 2004]. It approximates the perspective projection by a series of scaled orthographic projections and iteratively estimates the global pose parameters for the given set of 2D-to-3D correspondences.

Expression and pose estimation are iterated until convergence, resulting in a *tracked face mesh* M_{b}^t with associated blending weights and pose parameters. However, M_{b}^t lies within the space spanned by the blend shape model and lacks high-frequency face detail that appears in the video. These shortcomings will be tackled next.

7 Dense Expression and Pose Correction

After coarse expression and pose estimation, there may remain residual errors in the facial expression and head pose which can lead to misalignments when overlaying the 3D model with the video, see Fig. 5. The first reason for this error is that the used parametric blend shape model has a limit in expressibility and is not able to exactly reproduce a target expression that is not spanned by its basis of variation. The second reason is that the optimization of the previous section is performed over a fixed set of sparse feature vertices and excludes vertices that lie in other facial regions such as the cheeks or the forehead. To obtain an accurately aligned 3D mesh, we correct the initially estimated expression and pose over all vertices.

7.1 Temporally Coherent Corrective Flow

To correct the expression and pose of the face mesh M_{b}^t, obtained by blend shape tracking, we assign a *fixed* color to each vertex using projective texturing and blending from the starting frame f^{t_0}. Projecting M_{b}^t back onto the image plane at every time t results in the synthetic image sequence f_{s}, depicted in Fig. 6. To ensure optimal texturing for the results presented in Sec. 9, we manually improved the detected feature locations in the starting frame.

The idea behind our correction step is to compute the dense optical flow field that minimizes the difference between a synthetic frame f_{s}^t and its corresponding true target frame f^t, and then use the flow to deform the mesh. This corrective optical flow is denoted as $\boldsymbol{w}_1$ in Fig. 6. Computing such corrective optical flow independently for each time t introduces temporal artifacts in the corrected mesh geometry due to the lack of coherence over time in the optical flow estimation (see the second supplementary video for an illustration of such temporal artifacts). However, if we assume that M_{b}^t deforms coherently over time, the synthetic sequence will be smooth over time and, since the true sequence is smooth by construction, the corrective flow $\boldsymbol{w}_1$ between f_{s}^t and f^t has to vary smoothly over time as well.

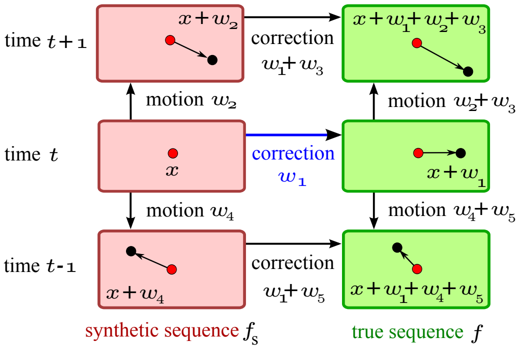

Figure 6: *Temporally coherent corrective flow estimation.*

To impose temporal smoothness on $\boldsymbol{w}_1$, we include frames at $t+1$ and $t-1$ and introduce a new optical flow method for the six-frame scenario depicted in Fig. 6. Exploiting the dependencies between the correspondences, the problem can be parametrized w.r.t. the reference frame f_{s}^t by $\boldsymbol{w}_1$ and four additional flows: $\boldsymbol{w}_2$ and $\boldsymbol{w}_4$ describing the face motion in the synthetic sequence, and $\boldsymbol{w}_3$ and $\boldsymbol{w}_5$ describing the temporal change in the corrective flow $\boldsymbol{w}_1$. Note that $\boldsymbol{w}_1+\boldsymbol{w}_3$ and $\boldsymbol{w}_1+\boldsymbol{w}_5$ represent the corrective flows in corresponding image points at $t+1$ and $t-1$ and so we can impose temporal coherence through the flow changes $\boldsymbol{w}_3$ and $\boldsymbol{w}_5$.

To estimate all unknown flows simultaneously, we minimize an energy consisting of data, smoothness, and similarity constraints:

$$E = \int_{\Omega} \left(\sum_{i=1}^{7} E_{\mathrm{data}}^i + \sum_{i=1}^{5} \alpha_i E_{\mathrm{smooth}}^i + \sum_{i=1}^{2} \gamma_i E_{\mathrm{sim}}^i \right) \mathrm{d}\boldsymbol{x} \ . \quad (9)$$

Data constraints. The data terms in energy (9) impose photometric constancy between corresponding points along the seven connections drawn in Fig. 6. For brightness constancy, the first data term between f_{s}^{t+1} and f^{t+1}, for example, takes the form:

$$E_{\mathrm{data}}^1 = \Psi_{\mathrm{d}} \left(|f^{t+1}(\boldsymbol{x}+\boldsymbol{w}_1+\boldsymbol{w}_2+\boldsymbol{w}_3) - f_{\mathrm{s}}^{t+1}(\boldsymbol{x}+\boldsymbol{w}_2)|^2 \right), \quad (10)$$

with $\Psi_{\mathrm{d}}(s^2) = \sqrt{s^2+(0.001)^2}$ the robust regularized L_1 penalizer. The remaining six data terms are constructed in the same way and all constraints are extended with a gradient constancy assumption and color information for improved matching accuracy.

Smoothness constraints. Similar in spirit to the scene flow scenario in [Valgaerts et al. 2012], we use a structure-aware regularization for the flows $\boldsymbol{w}_1$, $\boldsymbol{w}_2$ and $\boldsymbol{w}_4$ to improve the optical flow estimation in semantically meaningful regions of the face, e.g.:

$$E_{\mathrm{smooth}}^1 \ = \ \Psi_{\mathrm{s}} \left(|\nabla \boldsymbol{w}_1^{\top} \boldsymbol{r}_1|^2 \right) + \Psi_{\mathrm{s}} \left(|\nabla \boldsymbol{w}_1^{\top} \boldsymbol{r}_2|^2 \right) \ , \quad (11)$$

where $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$ are smoothing directions along and across flow structures and $\Psi_{\mathrm{s}}(s^2) = 2\lambda_{\mathrm{s}}^2 \sqrt{1+(s/\lambda_{\mathrm{s}})^2}$, with $\lambda_{\mathrm{s}} = 0.1$, a discontinuity-preserving function. As opposed to the corrective and motion flows, we regularize $\boldsymbol{w}_3$ and $\boldsymbol{w}_5$ much stronger:

$$E_{\mathrm{smooth}}^3 \ = \ |\nabla \boldsymbol{w}_3|^2 \quad \text{and} \quad E_{\mathrm{smooth}}^5 \ = \ |\nabla \boldsymbol{w}_5|^2 \ . \quad (12)$$

This quadratic regularization of the flow changes ensures that the corrective flow $\boldsymbol{w}_1$ varies smoothly over time.

Similarity constraints. Finally, we enforce the corrective flows $\boldsymbol{w}_1$, $\boldsymbol{w}_1+\boldsymbol{w}_3$ and $\boldsymbol{w}_1+\boldsymbol{w}_5$ to be similar to each other, i.e., we strongly penalize the magnitude of the flow changes:

$$E_{\mathrm{sim}}^1 = |\boldsymbol{w}_3|^2 \quad \text{and} \quad E_{\mathrm{sim}}^2 = |\boldsymbol{w}_5|^2 \; . \tag{13}$$

The respective terms (13) and (12) can be related to first and second order smoothness constraints along optical flow trajectories, as described in [Volz et al. 2011]. Contrary to their approach, we exploit the circular dependencies in our specific set-up for the purpose of coherently correcting one image sequence w.r.t. another.

The total energy (9) is minimized over all flows by a coarse-to-fine multiresolution strategy using a non-linear multigrid method. Computation can be sped up by using the forward and backward optical flows used for feature tracking in Sec. 5 as initialization.

7.2 Optical Flow-based Mesh Deformation

We correct the geometry of M_{b}^t by projecting the estimated optical flow $\boldsymbol{w}_1$ back onto the mesh and retrieving a corrective 3D motion vector for each vertex. Since our monocular setting has an inherent depth ambiguity, it is impossible to recover the correct motion in the Z-direction (i.e., in depth). However, we experienced that correcting each vertex in X- and Y-directions parallel to the image plane produces realistic and expressive results. Denoting the 3D motion field parallel to the image plane by $\boldsymbol{W}^t$, we propagate each vertex to its new position in the *corrected face mesh* M_{c}^t. To ensure a smooth deformation, we minimize the Laplacian-regularized energy:

$$E = \left\| LX_{\mathrm{c}}^t - LX_{\mathrm{b}}^t \right\|^2 + \mu^2 \sum_{i \in C^t} \left\| \boldsymbol{X}_{\mathrm{c},i}^t - (\boldsymbol{X}_{\mathrm{b},i}^t + \boldsymbol{W}_i^t) \right\|^2 , \tag{14}$$

where L is the Laplacian matrix with cotangent weights of M_{b}^t [Sorkine 2005], X_{c}^t and X_{b}^t matrices collecting the positions of all vertices $\boldsymbol{X}_i^t$ in M_{c}^t and M_{b}^t, $1 \leq i \leq n$, and μ a weight. The set C^t is a uniformly subsampled selection of currently visible vertices.

We perform the steps of Sec. 7.1 and Sec. 7.2 once per frame, but they could be applied iteratively. Note that they take us slightly outside the 3D shape space spanned by the blend shape model and yield an extremely accurate alignment of the mesh with the video. The alignment before and after correction is shown in Fig. 5.

8 Dynamic Shape Refinement

In a final step, we capture and add fine-scale surface detail to the tracked mesh, such as emerging or disappearing wrinkles and folds. Our approach is based on the *shape-from-shading* framework under general unknown illumination that was proposed in [Valgaerts et al. 2012] for the binocular reconstruction case. Based on an estimate of geometry and albedo, the method first estimates the unknown incident lighting at the current time step and then uses the known lighting to deform the geometry such that the rendered shading gradients and the image shading gradients agree. Essentially, this method inverts the rendering equation to reconstruct the scene, which is easier in a setting with multiple cameras where the fact that a surface is seen from several viewpoints constrains the solution space better.

To adjust this approach to the monocular case, we estimate the unknown illumination from a larger temporal baseline to compensate for the lack of additional cameras. In our setting, we assume that the illumination conditions do not change over time but that a ground truth light probe is not available. Therefore, we first estimate lighting, albedo, and refined surface geometry of the tracked face mesh for the first 10 frames of every video using the exact same approach as [Valgaerts et al. 2012]. In our monocular case, since the estimation is much more under-constrained and error-prone, we only use this result as an initialization. In a second step, we jointly use the initial albedo and fine scale geometry to estimate a single environment map that globally fits to all time steps. We then use this static light environment and estimate the dynamic geometry detail at each time step [Valgaerts et al. 2012]. The result of dynamic shape refinement is the final *refined face mesh* M_{r}^t. To remove temporal flicker in the visualization of the results, we update the surface normals by averaging them over a temporal window of size 5 and adapt the geometry to the updated normals [Nehab et al. 2005].

9 Results

We evaluate the performance of our approach on four video sequences of different actors with lengths ranging from 560 (22s) to 1000 frames (40s). Three videos are recorded with a Canon EOS 550D camera at 25 fps in HD quality (1920×1088 pixels) and one video with a GoPro outdoor camera at 30 fps in HD quality.

Performance capture results. The first two results are part of a calibrated binocular stereo sequence recorded under uncontrolled indoor lighting by Valgaerts et al. [2012]. We only use one camera output for our method and need one extra frame from the second camera for the blend shape model creation. Results for the first sequence, featuring very expressive gestures and normal speech, are shown in Fig. 7. All meshes consist of the same set of vertices and are produced by tracking and refining the personalized blend shape model of Fig. 3 over 560 frames (22s). The green screen is part of the recording and is not used. The figure shows that we are able to faithfully capture very challenging facial expressions, even for gestures that are not spanned by the blend shape model, e.g., the right column. The third row illustrates that our method effectively reconstructs a space-time coherent 3D face geometry with dynamic fine scale detail. Although the actor's head hardly moves in depth, our method estimates a small global translation component in the camera direction, which we discard for the 3D visualization in the figures and supplementary video. Fig. 10 shows a result for a second sequence of 620 frames (25s), featuring fast and expressive motion. Our results capture a high level of shape, motion, and surface detail.

Fig. 10 shows an additional result for a third sequence, newly recorded under similar conditions as the first two. The sequence depicts a recitation of a theatrical play and is extremely challenging due to is length of 1000 frames (40s), its diversity in facial expressions, and its fast and shaky head motion. The overlays in the figure show that we are able to estimate the X- and Y-components of the head pose very accurately and retrieve very subtle facial expressions, demonstrating the applicability of our method for demanding real world applications. We also captured an actor's facial performance outdoors with a lightweight GoPro camera. Despite the low quality of the video and the uncontrolled setting, we obtain accurate tracking results and realistic face detail, see Fig. 8. We recommend watching all results in our supplementary video. The video also shows a limiting head pose with extreme pitch for the GoPro sequence, and demonstrates how our algorithm fully recovers.

For all results, λ_i was 0.1 for the mouth features, 0.5 for the eye features and 0.2 for the remaining features. For the Canon results, $\alpha_1 = 500$, $\alpha_2 = \alpha_4 = 600$, and $\alpha_3 = \alpha_5 = 300$, and for the GoPro result $\alpha_2 = \alpha_4 = 700$, and $\alpha_3 = \alpha_5 = 400$. Further, $\gamma_1 = \gamma_2 = 50$ and $\mu = 0.5$. For improved accuracy around the eye lids, the eyes of the blend shape model were filled before tracking, but not visualized in the final results. Eye filling is only done once in the generic model and does not change any of our method steps.

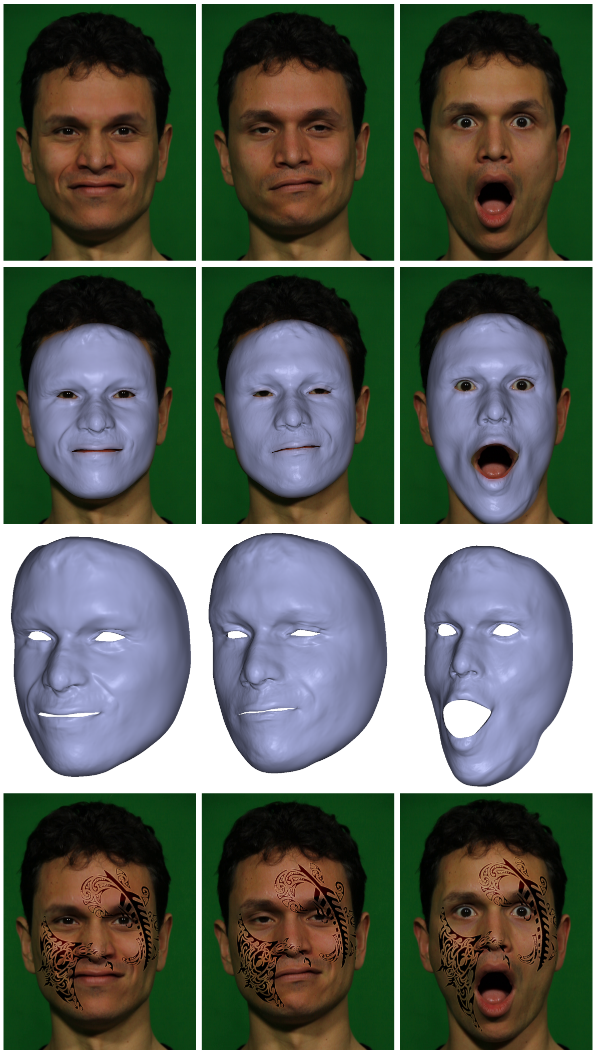

Figure 7: *Results for expressive facial motion.* Top to bottom: *The input frame, the corresponding blended overlay of the reconstructed mesh, a 3D view of the mesh, and an example of applying virtual face texture using the estimated geometry and lighting.*

Comparison with binocular reconstruction. In Fig. 9 we compare our results with a binocular facial performance capture method [Valgaerts et al. 2012]. In the middle and right panes we show our reconstructed face mesh for the target frame of Fig. 2 and its deviation w.r.t. the corresponding binocular result on the left. The colored error plots in the figure and in the supplementary video depict the Euclidean distance between the nearest visible vertices on the binocular and monocular meshes, and are produced by aligning the starting meshes of the sequence using rigid ICP and tracking them while discarding the small translation in the depth direction. As can be derived from the color scale, the meshes are very close in geometry and pose. Over all 560 frames, the average distance between the meshes was 1.71mm, with an average maximum distance of 7.45mm. Differences mainly appear near the lips, cheeks and forehead, where the dense expression correction of Sec. 7 cannot refine in depth. The supplementary document further reports this comparison for the first sequence of Fig. 10 (2.91mm average distance and 9.82mm average maximum distance) and illustrates that our monocular tracking is, in some cases, less susceptible to occlusions and drift, and is overall more robust to extreme head motions.

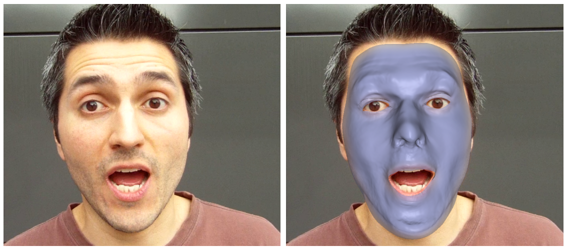

Figure 8: *Outdoor result captured with a hand-held GoPro camera.*

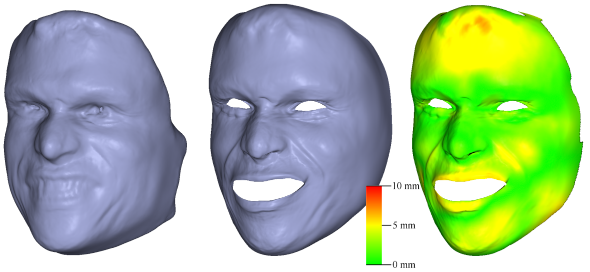

Figure 9: *Comparison with the binocular method of [Valgaerts et al. 2012].* Left to right: *Binocular reconstruction for the frame in Fig. 2. Our reconstruction. Euclidean distance (see error scale).*

Virtual face texture. Our capturing process introduces very little perceivable drift (see checkerboard texture in the video), so it is well suited for video augmentation tasks such as adding virtual textures or tattoos [7], see Figs. 1 and 7. To this end, we render the texture as a diffuse albedo map on the moving face and light it with the estimated incident illumination. The texture is rendered in a separate channel and overlaid with the input video using Adobe Premiere. Our detailed reconstruction and lighting of the deformation detail is important to make the shading of the texture correspond to the shading in the video, giving the impression of virtual make-up.

Run time. For the Canon sequences, the tracking and tracking correction run at a respective speed of 10s and 4m per frame, while the shading-based refinement has a run time of around 5m per frame. All three steps run fully automatically and can be started in parallel with a small frame delay. The only tasks requiring user intervention are the creation of the personalized blend shape model (Sec. 4, 20m), the one-time 2D-to-3D model coupling (Sec. 6.1, 10m) and the texturing of the blend shape model (Sec. 7.1, 10m).

Discussion and limitations. Our face tracking and refinement is automatic, but creating the personalized blend shape model and improving the 2D features in the first frame for texturing rely on a small amount of user interaction. This is because each of these tasks corresponds to a hard computer vision sub-problem. Currently, our 2D tracking and key frame selection start from a rest pose, but they could start from any frame with reliably detected features. A rest pose texture is also used for the optical flow-based correction, although a non-rest texture could be used as well (albeit harder).

Our results are very detailed and expressive, but not completely free from artifacts. As the dynamic texture in the video illustrates, small tracking inaccuracies can still be observed, e.g., around the teeth and lips. Small tangential floating of the vertices may also be present, as observed in the virtual texture overlays and the dy-

[7] Design taken from `www.deviantart.com/` under a CC license

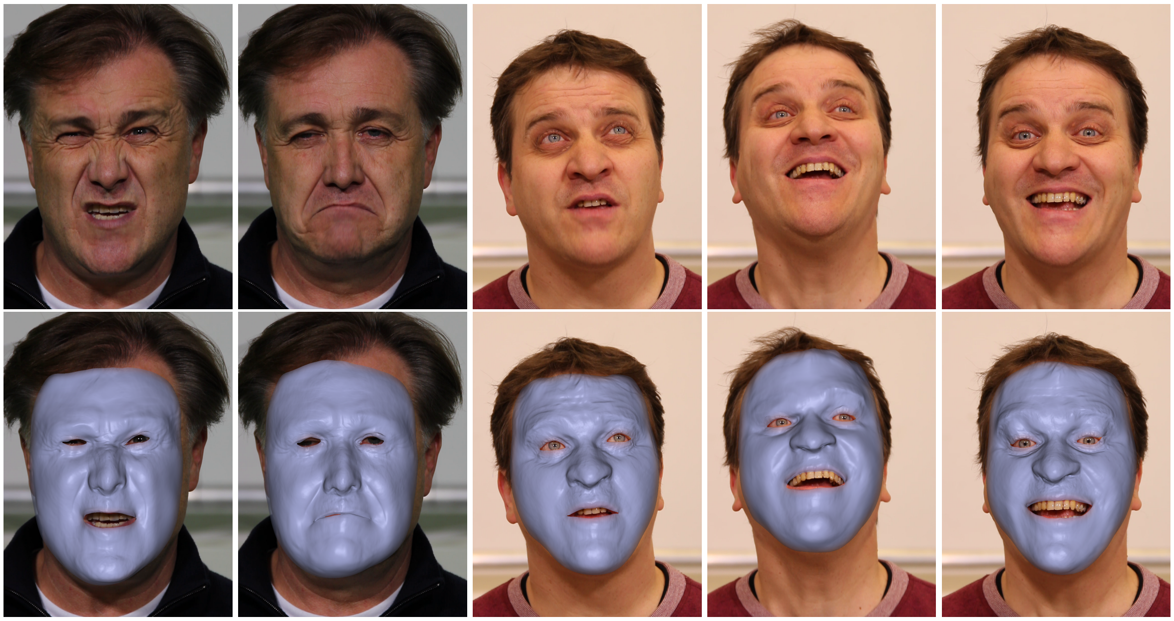

Figure 10: *Performance capture results for very expressive and fast facial gestures and challenging head motion for up to 1000 frames.*

namic texture in the UV domain. For the GoPro result, artifacts around the nose are visible due to the challenging low-quality input (noise, rolling shutter, colour saturation). Extremely fast motion can be problematic for feature tracking with optical flow and our method currently does not handle light changes as it violates the optical flow assumptions. Under strong side illumination, which causes cast shadows, the shading-based refinement may fail, but for general unknown lighting (indoor ceiling, bright outdoor diffuse), we are able to produce good results for scenarios deemed challenging in previous works. Partial occlusions (hand, glasses, hair) are difficult to handle with our dense optical flow optimization.

The inverse problem of estimating depth from a single image is far more challenging than in a multi-view setting, and depending on the camera parameters, even notable depth changes of the head may lead to hardly perceivable differences in the projected image. Consequently, even though our 3D geometry aligns well with the 2D video, there may be temporal noise in the estimated depth, which we filter out for the 3D visualizations. This limitation may stem from the use of a 2D PDM model and a 3D blend shape model that have a different dimensionality and expression range. We will work towards a better coupling of these models for 3D pose estimation.

10 Conclusion

We presented a method for monocular reconstruction of spatio-temporally coherent 3D facial performances. Our system succeeds for scenes captured under uncontrolled and unknown lighting, and is able to reconstruct very long sequences, scenes showing very expressive facial gestures, and scenes showing strong head motion. Compared to previously proposed monocular approaches, it reconstructs facial meshes of very high detail and runs fully automatically aside from a small manual initialization. It also fares very well in comparison to a recent state-of-the-art binocular facial performance capture method. Our approach combines novel 2D and 3D tracking and reconstruction methods, and estimates blend shape parameters that can be directly used by animators. We demonstrated its performance quantitatively and qualitatively on several face data sets, and also showcased its application to editing the appearance of faces.

Acknowledgements

We gratefully acknowledge all our actors for their participation in the recordings and thank the reviewers for their helpful comments. We thank James Tompkin for his suggestions and corrections.

References

AHONEN, T., HADID, A., AND PIETIKAINEN, M. 2006. Face description with local binary patterns: Application to face recognition. *IEEE TPAMI 28*, 12, 2037–2041.

ALEXANDER, O., ROGERS, M., LAMBETH, W., CHIANG, M., AND DEBEVEC, P. 2009. The Digital Emily Project: photoreal facial modeling and animation. In *ACM SIGGRAPH Courses*, 12:1–12:15.

ARUN, K. S., HUANG, T. S., AND BLOSTEIN, S. D. 1987. Least-squares fitting of two 3-D point sets. *IEEE TPAMI 9*, 5, 698–700.

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

BICKEL, B., BOTSCH, M., ANGST, R., MATUSIK, W., OTADUY, M., PFISTER, H., AND GROSS, M. 2007. Multi-scale capture of facial geometry and motion. *ACM TOG (Proc. SIGGRAPH) 26*, 33:1–33:10.

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

BLANZ, V., BASSO, C., VETTER, T., AND POGGIO, T. 2003. Reanimating faces in images and video. *CGF (Proc. EUROGRAPHICS) 22*, 641–650.

BORSHUKOV, G., PIPONI, D., LARSEN, O., LEWIS, J. P., AND TEMPELAAR-LIETZ, C. 2003. Universal capture: image-based facial animation for "The Matrix Reloaded". In *ACM SIGGRAPH 2003 Sketches*, 16:1–16:1.

BOUAZIZ, S., WANG, Y., AND PAULY, M. 2013. Online modeling for realtime facial animation. *ACM TOG (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 TOG (Proc. SIGGRAPH) 29*, 4, 41:1–41:10.

BRAND, M., AND BHOTIKA, R. 2001. Flexible flow for 3D nonrigid tracking and shape recovery. In *Proc. CVPR*, 315–322.

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

CHAI, J.-X., XIAO, J., AND HODGINS, J. 2003. Vision-based control of 3D facial animation. In *Proc. SCA*, 193–206.

CHUANG, E., AND BREGLER, C. 2002. Performance-driven facial animation using blend shape interpolation. Tech. Rep. CS-TR-2002-02, Stanford University.

COOTES, T. F., EDWARDS, G. J., AND TAYLOR, C. J. 2001. Active appearance models. *IEEE TPAMI 23*, 6, 681–685.

DALE, K., SUNKAVALLI, K., JOHNSON, M. K., VLASIC, D., MATUSIK, W., AND PFISTER, H. 2011. Video face replacement. *ACM TOG (Proc. SIGGRAPH Asia) 30*, 6, 130:1–130:10.

DANTONE, M., GALL, J., FANELLI, G., AND GOOL, L. V. 2012. Real-time facial feature detection using conditional regression forests. In *Proc. CVPR*, 2578–2585.

DAVID, P., DEMENTHON, D., DURAISWAMI, R., AND SAMET, H. 2004. SoftPOSIT: Simultaneous pose and correspondence determination. *IJCV 59*, 3, 259–284.

DECARLO, D., AND METAXAS, D. 1996. The integration of optical flow and deformable models with applications to human face shape and motion estimation. In *Proc. CVPR*, 231–238.

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. CA*, 68–79.

FURUKAWA, Y., AND PONCE, J. 2009. Dense 3D motion capture for human faces. In *Proc. CVPR*, 1674–1681.

GUENTER, B., GRIMM, C., WOOD, D., MALVAR, H., AND PIGHIN, F. 1998. Making faces. In *Proc. SIGGRAPH*, 55–66.

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

KEMELMACHER-SHLIZERMAN, I., SANKAR, A., SHECHTMAN, E., AND SEITZ, S. M. 2010. Being John Malkovich. In *Proc. ECCV*, 341–353.

LI, H., ROIVAINEN, P., AND FORCHEIMER, R. 1993. 3-D motion estimation in model-based facial image coding. *IEEE TPAMI 15*, 6, 545–555.

LI, H., WEISE, T., AND PAULY, M. 2010. Example-based facial rigging. *ACM TOG (Proc. SIGGRAPH) 29*, 3, 32:1–32:6.

LI, K., XU, F., WANG, J., DAI, Q., AND LIU, Y. 2012. A data-driven approach for facial expression synthesis in video. In *Proc. CVPR*, 57–64.

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

NEHAB, D., RUSINKIEWICZ, S., DAVIS, J., AND RAMAMOORTHI, R. 2005. Efficiently combining positions and normals for precise 3D geometry. *ACM TOG 24*, 3, 536–543.

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

PIGHIN, F., SZELISKI, R., AND SALESIN, D. 1999. Resynthesizing facial animation through 3D model-based tracking. In *Proc. CVPR*, 143–150.

PLATT, J. C. 1998. Sequential minimal optimization: A fast algorithm for training support vector machines. Tech. Rep. MSR-TR-98-14, Microsoft Research.

SARAGIH, J. M., LUCEY, S., AND COHN, J. F. 2011. Deformable model fitting by regularized landmark mean-shift. *IJCV 91*, 2, 200–215.

SORKINE, O. 2005. Laplacian mesh processing. In *EUROGRAPHICS STAR report*, 53–70.

VALGAERTS, L., BRUHN, A., MAINBERGER, M., AND WEICKERT, J. 2011. Dense versus sparse approaches for estimating the fundamental matrix. *IJCV 96*, 2, 212–234.

VALGAERTS, L., WU, C., BRUHN, A., SEIDEL, H.-P., AND THEOBALT, C. 2012. Lightweight binocular facial performance capture under uncontrolled lighting. *ACM TOG (Proc. SIGGRAPH Asia) 31*, 6, 187:1–187:11.

VLASIC, D., BRAND, M., PFISTER, H., AND POPOVIĆ, J. 2005. Face transfer with multilinear models. *ACM TOG (Proc. SIGGRAPH) 24*, 3, 426–433.

VOLZ, S., BRUHN, A., VALGAERTS, L., AND ZIMMER, H. 2011. Modeling temporal coherence for optical flow. In *Proc. ICCV*, 1116–1123.

WANG, Y., HUANG, X., SU LEE, C., ZHANG, S., LI, Z., SAMARAS, D., METAXAS, D., ELGAMMAL, A., AND HUANG, P. 2004. High resolution acquisition, learning and transfer of dynamic 3-D facial expressions. *CGF 23*, 677–686.

WEISE, T., LEIBE, B., AND GOOL, L. J. V. 2007. Fast 3D scanning with automatic motion compensation. In *Proc. CVPR*.

WEISE, T., LI, H., GOOL, L. J. V., AND PAULY, M. 2009. Face/Off: live facial puppetry. In *Proc. SIGGRAPH/Eurographics Symposium on Computer Animation*, 7–16.

WEISE, T., BOUAZIZ, S., LI, H., AND PAULY, M. 2011. Realtime performance-based facial animation. *ACM TOG (Proc. SIGGRAPH) 30*, 77:1–77:10.

WILLIAMS, L. 1990. Performance-driven facial animation. In *Proc. SIGGRAPH*, 235–242.

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 TOG 29*, 17:1–17:11.

XIAO, J., BAKER, S., MATTHEWS, I., AND KANADE, T. 2004. Real-time combined 2D+3D active appearance models. In *Proc. CVPR*, 535 – 542.

ZHANG, L., NOAH, CURLESS, B., AND SEITZ, S. M. 2004. Spacetime faces: high resolution capture for modeling and animation. *ACM TOG (Proc. SIGGRAPH) 23*, 548–558.