Vision, Modeling, and Visualization (2015)
D. Bommes, T. Ritschel and T. Schultz (Eds.)

# Temporally Consistent Wide Baseline Facial Performance Capture via Image Warping

M. Kettern[1], A. Hilsmann[1,2], P. Eisert[1,2]

[1]Fraunhofer HHI, Berlin, Germany
[2]Humboldt University, Berlin, Germany

**Figure 1:** *Sample results of our tracking aproach with two wide baseline input cameras*

**Abstract**

*In this paper, we present a method for detailed temporally consistent facial performance capture that supports any number of arbitrarily placed video cameras. Using a suitable 3D model as reference geometry, our method tracks facial movement and deformation as well as photometric changes due to illumination and shadows. In an analysis-by-synthesis framework, we warp one single reference image per camera to all frames of the sequence thereby drastically reducing temporal drift which is a serious problem for many state-of-the-art approaches. Temporal appearance variations are handled by a photometric estimation component modeling local intensity changes between the reference image and each individual frame. All parameters of the problem are estimated jointly so that we do not require separate estimation steps that might interfere with one another.*

## 1. Introduction

Facial performance capture is a very important topic in computer vision and graphics and has been actively researched for several decades. While marker-based approaches have matured and are readily available in many commercial systems, dense marker-less facial performance capture still poses numerous problems. While many approaches yield visually impressive results, temporal drift, especially in sequences with large motions and deformations is a key problem hindering the use of these methods in real-life applications. Moreover, most approaches use several separate steps for performance capture (e.g. reconstruction of one temporally unaligned mesh per frame - pixel tracking in image space - mesh alignment for temporal consistency - refinement for drift prevention), all of which have their requirements in order to yield good results and may even influence each other's accuracy.

In this paper, we present an integrated approach to temporally consistent facial performance capture that largely reduces temporal drift and does not require a separate 3D reconstruction of the facial geometry in each frame. The basic idea is to use an image-based *analysis-by-synthesis*-approach, synthesizing each frame of the sequence by warping and modifying a single reference image per camera corresponding to the motion and deformation of the underlying tracking model as well as the estimated illumination and shading. Contrary to most other approaches, our method does not require image correspondences between these cam-



DOI: 10.2312/vmv.20151263

eras, thus their placement may be optimized for coverage of the face and recording volume in order to enable capturing natural performances containing large head movements and rotations as well as to create representations that can be rendered from a broad range of viewpoints. Ambiguities arising from points that are covered by only one or even no camera at all are resolved by employing suitable smoothness terms.

**Contributions.** In this paper, we present an analysis-by-synthesis approach to temporally consistent facial performance capture of complex facial expressions even in long sequences and with wide-baseline setups. This is made possible by the following developments:

- An analysis-by-synthesis approach that is highly robust against temporal drift since all variations in appearance are modeled by warping and modifying a single reference image per camera
- Our approach overcomes the drastic variations in appearance resulting from different expressions by the integration of a photometric component into the tracking
- Our approach does not require separate steps but rather allows to model the captured performance directly in terms of a semantically consistent, deforming 3D model

We present a discussion of the most relevant related methods in section 2, followed by the detailed description of our approach in section 3. We present results and experimentally validate the performance of our method in section 4, followed by a conclusion.

## 2. Related Work

Over the last two decades, performance capture has matured as a research topic. Most commercial solutions rely on marker-based approaches, e.g. [Wil90, BBA*07] due to their robustness. However, markers are visible to the standard cameras recording the facial action, which makes the textures captured together with the performance useless without a vast amount of inpainting work. Another problem of marker-based approaches is that they only allow for reconstruction of the movement of a sparse point set on the facial surface and thus often fail to capture the subtleties of good facial acting.

Model-based methods allow to obtain semantically consistent mesh sequences even from monocular video streams [EG98, BBPV03, GVWT13] but the model geometry is either very coarse or has to be manually adapted for the target person by a 3D artist.

In order to track the facial geometry in 3D space without explicit deformation constraints, most approaches require a calibrated multi-camera or stereo capture setups and controlled lighting conditions. Additionally, temporal drift often needs to be addressed in explicit separate correction steps as detailed in the following. Under highly controlled studio conditions, [BPL*05] used the optical flow estimated for several well-placed cameras to deform a laser-scan model of an actor and capture highly detailed face textures at the same time. Temporal drift is reduced by computing the optical flow forwards and backwards. In [BHPS10], multiple stereo camera pairs are used which cover overlapping portions of the face to enhance capture resolution and optical flow computation for skin regions exhibiting few textural details above the level of skin-pores. An initial mesh is created by merging the depth maps obtained from the stereo pairs and propagated along pre-computed optical flow fields. In order to prevent temporal drift, an additional correction step based on the optical flow of the sequence of extracted and merged textures is applied.

A solution for a single stereo pair is presented in [VWB12] where a template mesh is computed from stereo correspondences and deformed along separately estimated scene flow fields [VBZ*10]. Temporal drift is reduced by a motion refinement step in which the mesh is updated to reduce the reprojection error between each frame and its successor.

A more extensive treatment of temporal drift can be found in [BHB*11] where the image sequences are divided by anchor frames automatically selected based on their similarity to a handpicked key-frame. The motion is tracked by a multiresolution forward-backward block matching approach. To overcome temporal drift, the authors introduce a "track-to-first" principle as a refinement step where each frame is individually matched to the key-frame. 3D geometry is reconstructed for each frame separately using [BBB*10], and temporal consistency is achieved by aligning the key frame reconstruction to the following reconstructions, guided by the estimated image motion fields. An improvement in reconstruction and tracking quality by factoring out surface shading using ambient occlusion has been proposed in [BBZG12].

These state-of-the-art methods for dense markerless facial performance capture divide the tracking process into several separate steps: Motion field estimation in image space, possibly per-frame geometry reconstruction and finally deformation of a template mesh using the estimated motion fields and/or reconstructions. Furthermore, all these methods contain an explicit separate treatment of temporal drift which is one of the most important problems in deformable surface tracking. Similar to our approach, several methods for tracking unstructured 3D data such as point clouds or depth maps use a deforming template shape together with suitable additional constraints (e.g. smoothness) [WJH*07, dAST*08, WLVP09].

We use an image-based analysis-by-synthesis approach, where motion estimation is based on warping a reference frame in order to synthesize each subsequent frame. Thereby, the relation between the deforming mesh and the underlying pixel information remains constant. This approach, however, can usually only be applied to short image sequences with small lighting and shading variance, be-

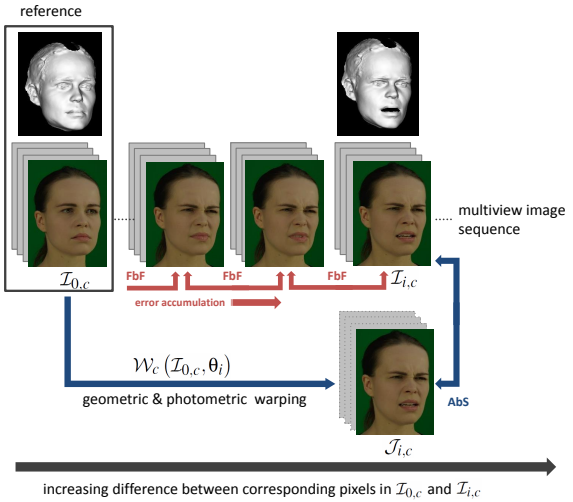


**Figure 2:** *Methodology overview and comparison of our analysis-by-synthesis (AbS) method with a standard frame-by-frame (FbF) tracking approach*

cause these can lead to increasing intensity differences between the warped reference and the current frame. This is especially important in facial performance capture as complex facial expressions can lead to drastic local shading variations. To handle this issue, we explicitly model shading and illumination variations which allows us to modify the reference image not only by geometric distortion but also photometrically. The benefits of compensating illumination and appearance changes in analysis-by-synthesis tracking have been shown e.g. by [WSVT13] for full-body stereo tracking. Our approach is partly inspired by work on 2D deformable surface augmentation where shading variations are explicitly modeled and estimated to achieve temporal consistency and enable realistic re-texturing [HE08, HE09].

## 3. Method

Input to our approach are calibrated and synchronized multi-view video sequences. As an initialization, a reference time-point is selected and a suitable 3D model of the target face, e.g. captured from a laser scan or image-based modeling approaches [BBE14], is aligned to the camera frames by matching sparse landmarks. Note that as our method does not require a small baseline camera setup, we rely on this initialization step to provide correct geometry. If the camera setup allows conducting 3D reconstructions from the captured frames, it is also possible to estimate the geometry directly from the captured sources.

The key idea of our approach is to use the same reference for motion estimation throughout the whole sequence instead of relying on motion estimated between consecutive frames. This is achieved by warping the reference frame of each camera according to the current motion hypothesis in order to resemble the current frame as closely as possible. In this work, "warping" not only means applying geometric transformations to an image but also locally changing its intensity, according to the photometric component as described below. Figure 2 illustrates our method in contrast to a standard frame-by-frame tracking approach.

Our approach consists of a two-component energy minimization problem for each frame minimizing an intensity-based error between the synthesized and the real images for each frame. A data term models geometric as well as photometric variations between the images. The geometric component models rigid motion as well as deformation of the face, whereas the photometric component models intensity variations, as induced by shading and illumination changes between the images. Additionally, several regularization terms minimize the influence of noise and outliers in the image-based estimation process.

### 3.1. Parameter Estimation

In the following, we will index the cameras used for capture by $c$, the time points by $i$ and the $K$ vertices of the mesh used for tracking the surface by $k$. The image of camera $c$ at timepoint $i$ will be denoted by $\mathcal{I}_{i,c}$ and without loss of generality we will assume the reference frame to have been captured at time point 0.

For estimating the motion and deformation of the face from reference image $\mathcal{I}_{0,c}$ of camera $c$ to one of its successors $\mathcal{I}_{i,c}$, we aim at minimizing the difference between $\mathcal{I}_{i,c}$ and a rendered image $\mathcal{J}_{i,c} = \mathcal{W}_c(\mathcal{I}_{0,c}, \theta_i)$, where $\mathcal{W}_c$ is a view-dependent warping function that applies all geometric as well as photometric changes to the reference image $\mathcal{I}_{0,c}$, as induced by the estimated tracking parameters $\theta_i$ between the time points 0 and $i$. The parametrization of this warp function is given by

$$\theta_i = \begin{bmatrix} \mathbf{r}_i \\ \mathbf{t}_i \\ \mathbf{u}_i \\ \varphi_i \end{bmatrix} \tag{1}$$

where $\mathbf{r}_i$ represents the 3 degrees of freedom of object rotation, $\mathbf{t}_i$ is its translation in world coordinate space, $\mathbf{u}_i$ is a vector containing $x,y,z$-offsets for each vertex representing the object's deformation and $\varphi_i$ is a vector containing one value per vertex for the photometric adaption of the key-frame texture. Since the object to be tracked is represented by a triangle mesh, the rendering can easily be sourced out to the GPU where it can be performed extremely fast even for complex meshes.

The residual vector for measuring the distance between images $\mathcal{I}_{i,c}$ and $\mathcal{J}_{i,c}$ is given for each pixel $\mathbf{p}$ of $\mathcal{I}_{i,c}$ by

$$r_{i,c}^{(img)}(\mathbf{p}, \theta_i) = \left(\mathcal{I}_{i,c}(\mathbf{p}) - \mathcal{J}_{i,c}(\mathbf{p})\right) \tag{2}$$

$$\mathcal{J}_{i,c} = \mathcal{W}_c(\mathcal{I}_{0,c}, \theta_i) \tag{3}$$

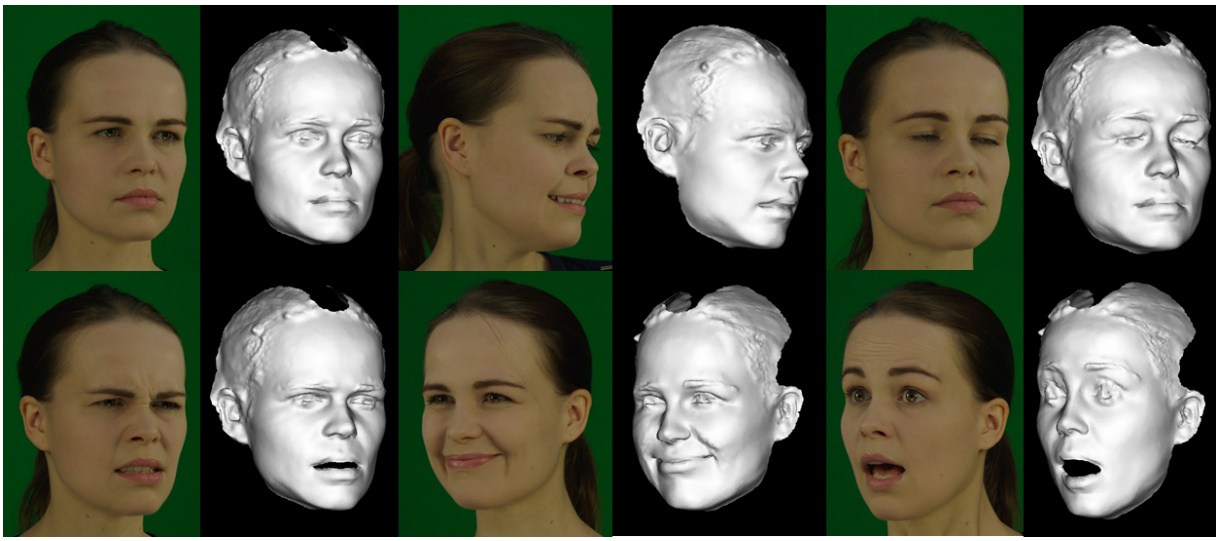

**Figure 3:** *Side-by-side comparison of input frames and flat-shaded tracked geometry*

and is computed for all pixels $\mathbf{p} \in \Omega$, the image region covered by the output of $\mathcal{W}_c\left(\mathcal{I}_{0,c}, \theta_i\right)$, the rendered model with the tracking parameters applied.

The final cost function for the data term is then given by

$$\mathcal{E}_i^{(img)} = \sum_c \Phi\left(\mathbf{r}_{i,c}^{(img)}\right) \tag{4}$$

where $\Phi$ is a suitable kernel function, e.g. the Square-norm or a robust norm-like function.

### 3.2. Geometric and Photometric Components

The position $\mathbf{v}_i^{(k)}$ of a vertex with index $k$ of the mesh parametrized by $\theta_i$ is given relative to its position $\mathbf{v}_0^{(k)}$ at time 0 by

$$\mathbf{v}_i^{(k)} = \mathbf{R}_i\left(\mathbf{v}_0^{(k)} + \mathbf{u}_i^{(k)}\right) + \mathbf{t}_i \tag{5}$$

where $\mathbf{R}_i$ is a rotation matrix and $\mathbf{t}_i$ a translation vector which together define the rigid transformation of the mesh, and $\mathbf{u}_i^{(k)}$ is an offset vector which describes the local deformation for each vertex individually. The rotation $\mathbf{R}_i$ is parametrized by $\mathbf{r}_i = [r_x\, r_y\, r_z]^T$, which are the first elements of $\theta_i$.

Let $\mathbf{x}$ denote the point on the mesh surface corresponding to an image pixel $\mathbf{p}$ in the synthetic image $\mathcal{J}_i$. If $T(\mathbf{x})$ is the mesh triangle containing $\mathbf{x}$, its position can be expressed by its barycentric coordinates:

$$\mathbf{x} = \sum_{k \in T(\mathbf{x})} \mathbf{v}_i^{(k)} \beta_i^{(k)}(\mathbf{x}) \tag{6}$$

where $\beta_i^{(k)}(\mathbf{x})$ is the barycentric coordinate of $\mathbf{x}$ with respect to vertex $k$. The color of $\mathcal{J}_i(\mathbf{p})$ for a simple warp-based rendering approach would be given by

$$\hat{\mathcal{J}}_{i,c}(\mathbf{p}) = \mathcal{I}_0\left(\sum_{k \in T(\mathbf{x})} \Psi_c\left(\mathbf{v}_0^{(k)}\right) \beta_0^{(k)}(\mathbf{x})\right) \tag{7}$$

where $\Psi_c$ is the camera projection function for view $c$. In order to account for intensity variations during the sequence to be tracked, we extend (7) by multiplying with an additional photometric component per vertex:

$$\mathcal{J}_{i,c}(\mathbf{p}) = \hat{\mathcal{J}}_{i,c}(\mathbf{p}) \sum_{k \in T(\mathbf{x})} \beta_i^{(k)}(\mathbf{x})\, \varphi_i^{(k)} \tag{8}$$

$$= \mathcal{W}_c\left(\mathcal{I}_{0,c}, \theta_i, \mathbf{p}\right) \tag{9}$$

where $\varphi_i^{(k)}$ is the photometric component of $\mathcal{W}_c$ corresponding to vertex $k$. Note that the photometric component is treated as view-independent in this work so that all components of the estimated parameters $\theta_i$ are independent of the number of views and the view positions.

### 3.3. Regularization via the Mesh Laplacian

In order to obtain smooth surface deformations, decrease noise and drift, as well as to resolve ambiguities (e.g. at vertices visible in only one or even no camera), we employ a twofold regularization approach based on the mesh Laplacian which penalizes both strong variations in local mesh geometry over time as well as divergence from the starting mesh. The *Laplacian differential* [Sor05] of a vertex $\mathbf{v}_i^{(k)}$ describes its position as relative to its one-ring (the set of direct neighbors) $N(k)$. In this work, we use the uniform Laplacian for which this differential is given by

$$\hat{\mathbf{d}}_i^{(k)} = \mathbf{v}_i^{(k)} - \frac{1}{|N(k)|} \sum_{j \in N(k)} \mathbf{v}_i^{(j)} \tag{10}$$

These differentials, however, are not invariant to a rotation of the mesh [Sor04] and thus, we rotate them by the inverse rotational component of the rigid transformation estimated for the corresponding frame, yielding

$$\mathbf{d}_i^{(k)} = \mathbf{R}_i^T \hat{\mathbf{d}}_i^{(k)} \tag{11}$$

The residual for the regularization term enforcing smooth surface deformations and decreasing the influence of noise is given by the difference between the Laplacian differentials of the current mesh and the ones of the mesh used in the previous frame:

$$\mathbf{r}_i^{(def)} = \begin{bmatrix} \mathbf{d}_i^{(0)} \\ \vdots \\ \mathbf{d}_i^{(K-1)} \end{bmatrix} - \begin{bmatrix} \mathbf{d}_{i-1}^{(0)} \\ \vdots \\ \mathbf{d}_{i-1}^{(K-1)} \end{bmatrix} \tag{12}$$

Similarly, residual $\mathbf{r}_i^{(acc)}$ for preventing error accumulation in the mesh geometry is defined as the difference between the Laplacian differentials of the current mesh and the ones of the mesh in frame 0. The regularization penalty thus amounts to

$$\mathcal{E}_i^{(reg)} = \Phi\left(\lambda_1 \mathbf{r}_i^{(def)} + \lambda_2 \mathbf{r}_i^{(acc)}\right) \tag{13}$$

where $\lambda_1, \lambda_2$ are weight factors which control the regularization process and are dependent on the mesh resolution (we used 2 and 20, respectively, in our experiments).

The photometric component is also regularized by a mesh-based Laplacian term which uses the differentials given by

$$\mathbf{c}_i^{(k)} = \varphi_i^{(k)} - \frac{1}{|N(k)|} \sum_{j \in N(k)} \varphi_i^{(j)} \tag{14}$$

and directly penalizes them such that

$$\mathbf{r}_i^{(regp)} = \begin{bmatrix} \mathbf{c}_i^{(0)} \\ \vdots \\ \mathbf{c}_i^{(K-1)} \end{bmatrix} \tag{15}$$

$$\mathcal{E}_i^{(regp)} = \Phi\left(\mathbf{r}_i^{(regp)}\right) \tag{16}$$

### 3.4. Optimization Strategy

In order to ensure quick convergence and to bridge large motions between successive frames, we employ a coarse-to-fine optimization scheme with a downsampling factor of 0.5. On each resolution level, we first compute a rigid fit of the model using the image-based error (4) without the regularization terms and only the first six elements of the parameter vector $\theta_i$ in (1). Afterwards, we jointly refine the rigid position and compute the deformation parameters by minimizing the error over the full parameter vector $\theta_i$. This approach favors rigid motion over deformation, thereby stabilizing the tracking and minimizing local drift in the computed vertex offsets

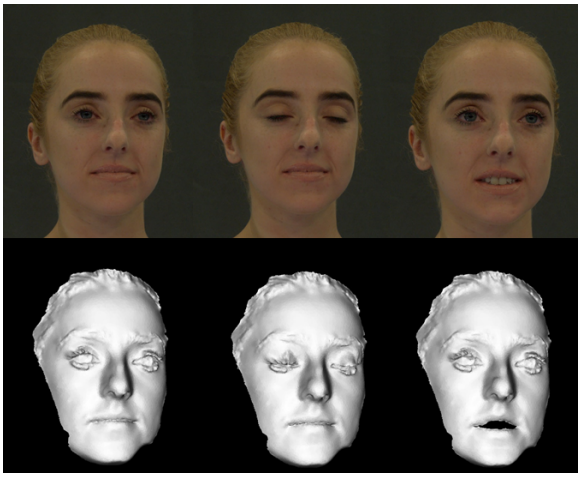

**Figure 4:** *Results form Dataset B with 4 cameras and 4K camera resolution, challenging eye movement and eyelash geometry*

$\mathbf{u}_i$. We use the Charbonnier penalty function

$$\Phi(\mathbf{r}) = \sqrt{\mathbf{r}^T \mathbf{r} + \varepsilon^2} \tag{17}$$

which is a robust error norm reducing the influence of outliers to the error function. In the data term, this makes the optimization more robust against noise in the data, while in the smoothness term, it allows for discontinuities in the deformation and photometric parameters. The overall cost function is given by

$$\mathcal{E}_i = \mathcal{E}_i^{(img)} + \mathcal{E}_i^{(reg)} + \gamma \mathcal{E}_i^{(regp)} \tag{18}$$

where $\gamma$ is used to weight the regularization of the photometric component and has been set to 0.1 in our experiments.

The optimization is done in an iterative fashion with the single steps calculated by a generalized Gauss-Newton update rule

$$\theta_{i+1} = \theta_i - \triangle\theta_i \tag{19}$$

$$J_\varepsilon^T \operatorname{diag}\left(\frac{d^2\Phi}{d\mathbf{r}^2}\right) J_\varepsilon \triangle\theta_i = J_\varepsilon^T \frac{d\Phi}{d\mathbf{r}} \tag{20}$$

where $J_\varepsilon$ is the complete Jacobian matrix of the overall residual in the toral error function (18).

This generalized Gauss-Newton update directly takes the derivatives of the kernel function $\Phi$ into account which are $\frac{d\Phi}{d\mathbf{r}} = \mathbf{r}$ and $\frac{d^2\Phi}{d\mathbf{r}^2} = \mathbf{1}$ in the case of the $L_2^2$-norm. Note that this approach is related but not equal to iteratively reweighted least squares estimation [Gre84] and is more general in the sense that it uses the true second derivative of the kernel. If the computed update step leads to an error increase, i.e. $\mathcal{E}_i(\theta_{i+1}) > \mathcal{E}_i(\theta)$, we start a line search in order to generate updates $\theta_{i+1}(\alpha) = \theta_i - \alpha\triangle\theta_i$, $\alpha < 1$ that could still decrease the error.

Since we aim at minimizing the error function (18) with a quadratic optimization algorithm, we need the Jacobian matrices of the residual functions for all error terms involved. If we use RGB color images and the mesh has $K$ vertices, the Jacobian of $\mathbf{r}_i^{(img)}$ from equation (4) is a sparse $(3|\Omega|) \times (6+4K)$ matrix with its first 6 columns fully occupied and the following $4K$ columns being sparse. This matrix contains 3 rows for each pixel $\mathbf{p}$, one for each color channel, and each row will be given by

$$\frac{\partial \mathcal{J}_{i,c}(\mathbf{p})}{\partial \theta_i} = \begin{bmatrix} \mathbf{Q}_i(\mathbf{x})\,\mathbf{z} \\ \mathbf{z} \\ \mathbf{B}_i(\mathbf{p})\,\mathbf{z} \\ \hat{\mathcal{J}}_{i,c}(\mathbf{p})\,\mathbf{b}_i(\mathbf{p}) \end{bmatrix}^T \tag{21}$$

where $\mathbf{Q}_i(\mathbf{x})$ is the Jacobian of the rotation of $\mathbf{x}$ by $\mathbf{R}_i$, and

$$\mathbf{z}^T = \nabla \mathcal{J}_{i,c}(\mathbf{p}) \frac{d\Psi_c}{d\mathbf{x}} \tag{22}$$

is the $1 \times 3$ row vector denoting the product of the image gradient of $\mathcal{J}_{i,c}$ at $\mathbf{p}$ and the $2 \times 3$ Jacobian matrix of the projection function $\Psi_c$ with respect to $\mathbf{x}$. In practice, we blend the image gradient of $\mathcal{J}_{i,c}$ with the gradient of the target image as suggested in [HS80] to obtain

$$\nabla \mathcal{J}^*_{i,c} = \frac{1}{2}\left(\nabla \mathcal{J}_{i,c} + \nabla \mathcal{I}_{i,c}\right) \tag{23}$$

Vector $\mathbf{b}_i(\mathbf{x})$ represents the barycentric coordinates of $\mathbf{x}$ as a sparse $K \times 1$ vector which contains one row per vertex. If vertex $k$ is an element of $T(\mathbf{x})$, the corresponding row of $\mathbf{b}_i(\mathbf{x})$ is set to $\beta_i^{(k)}(\mathbf{x})$. Matrix $\mathbf{B}_i(\mathbf{p})$ is a $3K \times 3$-matrix containing one $3 \times 3$-block for each row of $\mathbf{b}_i(\mathbf{x})$ and is given by

$$\mathbf{B}_i(\mathbf{p}) = \left[\left(\mathbf{D}^{(0)}\right)^T \cdots \left(\mathbf{D}^{(K-1)}\right)^T\right]^T \tag{24}$$

$$\mathbf{D}^{(k)} = \operatorname{diag}\left(\begin{bmatrix} b_i^{(k)}(\mathbf{x}) \\ b_i^{(k)}(\mathbf{x}) \\ b_i^{(k)}(\mathbf{x}) \end{bmatrix}\right) \tag{25}$$

where $b_i^{(k)}(\mathbf{x})$ is the $k$-th element of $\mathbf{b}_i(\mathbf{x})$. Since all elements of $\mathbf{b}_i(\mathbf{x})$ are zero except for the three elements corresponding to the vertices of triangle $T(\mathbf{x})$, $\mathbf{B}_i(\mathbf{p})$ is sparse.

The Jacobians of both, $\mathbf{r}_i^{(def)}$ and $\mathbf{r}_i^{(acc)}$ from equation (13), with respect to the vertex offsets $\mathbf{u}_i$ are given by sparse $3K \times 3K$-matrices which contain the coefficients for computing the Laplacian differentials, multiplied by $\mathbf{R}_i^T$.

## 4. Results and Experimental Evaluation

For the results we used data from two different real capture sessions. Dataset A (dark haired woman, green background) was captured using two synchronized and calibrated cameras with a resolution of $1920 \times 1080$ and 60 frames per second. Dataset B (blond hair, grey background) was captured with 4 cameras at 4K resolution (figure 4). The reference model was derived with an image-based reconstruction method [SKHE11], using 7 pairs of D-SLR cameras. Figure 3 displays a side-by-side comparison of example input frames and the tracking results in order to illustrate the versatility of the method for both tracking complex deformations as well as substantial off-plane rotations (e.g. top row, center pair).

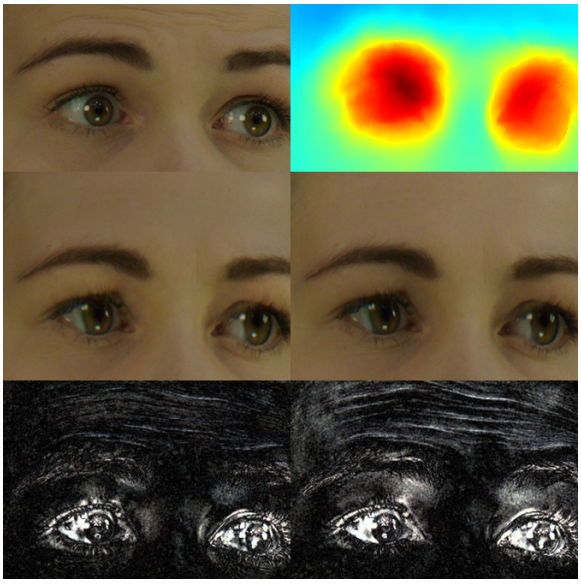

**Figure 5:** *Effect of the photometric component on the rendering (detail): target frame (top left), value map of the photometric component (top right), warped reference frame with (center left) and without (center right) the photometric component applied during rendering, absolute difference of warped images and target image (bottom)*

In order to experimentally confirm the performance of our approach, we conducted several tests putting our method next to other approaches to face tracking realized in the same framework for a direct comparison.

**Effects of photometric component.** Figure 5 illustrates the effect of the photometric component on rendering the warped reference frame. The top row shows the target frame and a value map of the photometric component. The center row shows the warped reference frame with (left) and without (right) photometric component being applied. The bottom row displays the absolute difference images between the target frame and the synthesized frames from the center row. These images illustrate that the photometric component has accounted for several brightness changes during the tracking, especially at the eyelids and the forehead.

The effects of the photometric component on the tracking itself is illustrated in figure 6 where a result image of a tracking pass without the photometric component (left) is compared to the corresponding image created by tracking with the photometric component (right). While prominent

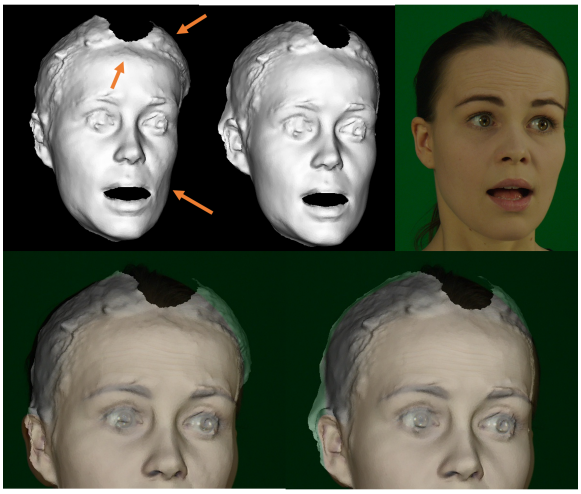

**Figure 6:** *Comparison of tracking results for our analysis-by-synthesis approach with the photometric component being disabled (left) and enabled (right). Upper row: tracking results as rendered meshes, target frame (right). The arrows point at regions where local brightness changes have caused errors in the estimated deformation. Bottom row: overlays of the rendered mesh and the target frame*

cues like eyes, lips and eyebrows have been tracked correctly without the photometric component, less textured regions are more sensitive to brightness changes on the surface, such that these effects may lead to errors if not treated properly. The bottom row contains overlays of the rendered tracking mesh and the target images for visual comparison.

**Drift prevention via Analysis-by-Synthesis.** One of the main contributions of this paper is that the presented tracking is highly robust against temporal drift. While this may be evident when contemplating the use of the key-frame $\mathcal{I}_0$ as the source of all synthesized frames $\mathcal{J}_t$, a simple comparison with a standard frame-by-frame approach shows that this choice indeed strongly decreases temporal drift. The method we use for comparison is built within the same tracking framework, with the only difference that we use frame $\mathcal{I}_{t-1}$ as the rendering source for $\mathcal{J}_t$, instead of $\mathcal{I}_0$, allowing to directly infer the influence of the reference chosen for warping. Figure 7 displays the estimated geometry for frame 30 of a challenging sequence with quick changes in expression and pose. Equal weights have been used for all smoothness terms. The estimated geometry of both approaches seems visually valid although our proposed single-reference approach (center) has followed the deformation more closely (e.g. lip shape). As shown in the second row (overlay of the tracked geometry and the target image), however, the position of the mesh has already drifted by a substantial amount for the frame-by-frame tracking approach. As expected, the tracking results when using $\mathcal{I}_0$ as the reference frame throughout the sequence do not exhibit any visible drift. Adding a backward warping component like the one being used to alleviate drift in [BPL*05] did not significantly decrease drift in our experiment.

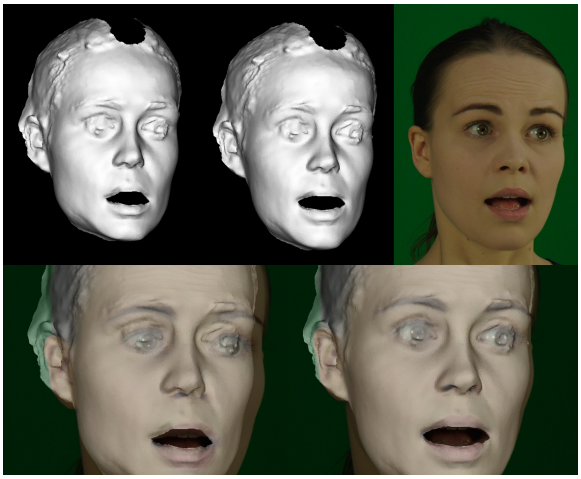

**Figure 7:** *Comparison of estimated geometry for frame-by-frame tracking (left) versus our approach with a single reference frame (right). Upper row: tracking results, target frame (right). Lower row: overlay of tracked geometry and target image. The shifting effect of temporal drift in the frame-by-frame approach is clearly visible.*

Table 1 shows the mean squared error (MSE) between a target frame and the corresponding synthesized instance of frame $\mathcal{I}_0$ for the different tracking methods used in our experiments, indicating the consistency of each tracking method. The top row shows the error for our approach using $\mathcal{I}_0$ as the reference frame for the whole sequence, the center row for conventional frame-by-frame tracking and the bottom row for frame-by-frame tracking with an additional backwards warping term. The table illustrates that the proposed method yields the best results in this comparison. The photometric component provides an additional error decrease. Surprisingly, forward-backward estimation of the optical flow (bottom row) error did not yield better results than simple forward frame-by-frame tracking in this experiment. The slight error increase when applying the photometric component to a frame-by-frame approach results from the increased adaptability between pairs of successive frames which in this case tends to amplify drift.

## 5. Conclusion

We have presented an analysis-by-synthesis approach to temporally consistent facial performance capture. Since our method uses a single reference frame (per camera) which is warped to synthesize all subsequent frames, it is robust against temporal drift as has been validated experimentally by comparison with an approach that uses pairs of subsequent frames for tracking. Bradley et al [BHPS10] correctly

| Warping | no PC | PC |
|---|---|---|
| $\mathcal{I}_0^{\rightarrow}$ | 0.0041 / 0.0026 | **0.0025 / 0.0020** |
| $\mathcal{I}_{i-1}^{\rightarrow}$ | 0.0154 / 0.0165 | 0.0166 / 0.0182 |
| $\mathcal{I}_{i-1}^{\rightarrow}, \mathcal{I}_i^{\leftarrow}$ | 0.0164 / 0.0163 | 0.0166 / 0.0169 |

**Table 1:** *Comparison of MSE error between synthesized and target frame with different tracking approaches, for left / right camera. Rows: warping reference frame and direction. Columns: photometric component disabled / enabled*

observe that "If it were possible to accurately compute flow between the first video image and every other frame, there would be no accumulation of error. Unfortunately, temporally distant video images in a capture sequence are usually too dissimilar to consider this option." The proposed method tackles this dissimilarity problem by adding a photometric component which allows to estimate brightness changes resulting from deformation, movement and self-shadowing, which are then applied to the reference frame when synthesizing a target image.

The image warping used for image synthesis is directly induced by the deformations applied to the reference model for each time point. This makes our method an integrated, single-step approach as opposed to most state-of-the-art methods that use at least one stage for tracking pixel motion and another stage for following this motion with a tracking mesh. Also, a drift correction step is needed in most state-of-the-art methods but is not necessary in our approach.

In future work, we aim to extend our method by illumination estimation to allow for more detailed estimation of geometric deformations, e.g. at wrinkles, by analyzing their self-shadowing behavior. In order to use the results for applications such as free-viewpoint rendering, we will also add a texture synthesis component which will stitch the textures captured by the individual cameras into one complete texture representing the area covered by all cameras together.

## References


[BBA*07] BICKEL B., BOTSCH M., ANGST R., MATUSIK W., OTADUY M., PFISTER H., GROSS M.: Multi-scale capture of facial geometry and motion. *ACM Transactions on Graphics 26*, 3 (2007), 33. 2

[BBB*10] BEELER T., BICKEL B., BEARDSLEY P., SUMNER B., GROSS M.: High-quality single-shot capture of facial geometry. *ACM Transactions on Graphics 29*, 4 (2010), 1. 2

[BBE14] BLUMENTHAL-BARBY D. C., EISERT P.: High-resolution depth for binocular image-based modeling. *Computers and Graphics (Pergamon) 39*, 1 (2014). 3

[BBPV03] BLANZ V., BASSO C., POGGIO T., VETTER T.: Reanimating Faces in Images and Video. *Computer Graphics Forum 22*, 3 (2003), 641–650. 2

[BBZG12] BEELER T., BRADLEY D., ZIMMER H., GROSS M.: Improved reconstruction of deforming surfaces by cancelling ambient occlusion. *Lecture Notes in Computer Science 7572 LNCS*, PART 1 (2012), 30–43. 2

[BHB*11] BEELER T., HAHN F., BRADLEY D., BICKEL B., BEARDSLEY P., GOTSMAN C., SUMNER R. W., GROSS M.: High-quality passive facial performance capture using anchor frames. *ACM Transactions on Graphics 30*, 4 (2011), 1. 2

[BHPS10] BRADLEY D., HEIDRICH W., POPA T., SHEFFER A.: High resolution passive facial performance capture. *ACM Transactions on Graphics 29*, 4 (2010), 1. 2, 7

[BPL*05] BORSHUKOV G., PIPONI D., LARSEN O., LEWIS J. P., TEMPELAAR-LIETZ C.: Universal capture - image-based facial animation for "The Matrix Reloaded". In *ACM SIGGRAPH 2005 Courses* (2005), p. 16. 2, 7

[dAST*08] DE AGUIAR E., STOLL C., THEOBALT C., AHMED N., SEIDEL H.-P., THRUN S.: Performance capture from sparse multi-view video. *ACM Transactions on Graphics 27*, 3 (2008), 1. 2

[EG98] EISERT P., GIROD B.: Analyzing facial expressions for virtual conferencing. *IEEE Computer Graphics and Applications 18*, 5 (1998). 2

[Gre84] GREEN P. J.: Iteratively Reweighted Least Squares for Maximum Likelihood Estimation , and some Robust and Resistant Alternatives. *Journal of the Royal Statistical Society 46*, 2 (1984), 149–192. 5

[GVWT13] GARRIDO P., VALGAERT L., WU C., THEOBALT C.: Reconstructing Detailed Dynamic Face Geometry from Monocular Video. *ACM Transactions on Graphics 32*, 6 (2013). 2

[HE08] HILSMANN A., EISERT P.: Tracking deformable surfaces with optical flow in the presence of self occlusion in monocular image sequences. *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPR), Workshops* (2008). 3

[HE09] HILSMANN A., EISERT P.: Realistic Cloth Augmentation in Single View Video. In *Vision, Modeling, and Visualization Workshop* (2009). 3

[HS80] HORN B. K., SCHUNCK B. G.: *Determining Optical Flow*. Tech. rep., Cambridge, MA, USA, 1980. 6

[SKHE11] SCHNEIDER D. C., KETTERN M., HILSMANN A., EISERT P.: A Global Optimization Approach to High-detail Reconstruction of the Head. In *Vision, Modeling, and Visualization (2011)* (2011), Eisert P., Hornegger J., Polthier K., (Eds.), The Eurographics Association. 6

[Sor04] Laplacian Surface Editing. In *Eurographics Symposium on Geometry Processing* (2004), SGP '04, ACM, pp. 175–184. 5

[Sor05] SORKINE O.: Laplacian Mesh Processing. *Eurographics - State of the Art Reports*, Section 4 (2005), 53–70. 4

[VBZ*10] VALGAERTS L., BRUHN A., ZIMMER H., WEICKERT J., STOLL C., THEOBALT C.: Joint estimation of motion, structure and geometry from stereo sequences. *Lecture Notes in Computer Science 6314 LNCS*, PART 4 (2010), 568–581. 2

[VWB12] VALGAERTS L., WU C., BRUHN A.: Lightweight binocular facial performance capture under uncontrolled lighting. *ACM Transactions on Graphics* (2012). 2

[Wil90] WILLIAMS L.: Performance-driven facial animation. In *Proceedings of the 17th Conference on Computer Graphics and Interactive Techniques* (1990), SIGGRAPH, ACM. 2

[WJH*07] WAND M., JENKE P., HUANG Q., BOKELOH M., GUIBAS L., SCHILLING A.: Reconstruction of deforming geometry from time-varying point clouds. *Eurographics symposium on Geometry processing* (2007), 49–58. 2

[WLVP09] WEISE T., LI H., VAN GOOL L., PAULY M.: Face/Off: Live Facial Puppetry. *Eurographics Symposium on Computer Animation - SCA* (2009), 7. 2

[WSVT13] WU C., STOLL C., VALGAERTS L., THEOBALT C.: On-set performance capture of multiple actors with a stereo camera. In *ACM Transactions on Graphics* (2013), vol. 32. 3