A Hybrid Approach for Facial Performance Analysis and Editing

Wolfgang Paier, Markus Kettern, Anna Hilsmann and Peter Eisert, *Member, IEEE,*

***Abstract*—As of today, fine-grained editing of facial performances in movie and video production requires either retouching every single frame or creating a highly detailed CGI model of the actor, both of which is restricted to high-budget productions. In this paper, we present an example based approach for facial performance editing that achieves realistic results with standard equipment and very little manual intervention. Based on a model-free surface tracking approach, temporally consistent dynamic texture sequences are extracted from multiple video streams. Using such geometry-plus-texture sequences allows transferring facial expressions/performances between videos which enables the editor for example to change the facial expression, while leaving the rest of the video untouched. Moreover, concatenating and/or looping these sequences in a motion-graph like manner gives a convenient way of composing novel facial performances from multiple source videos. Finally, we present a blending method to seamlessly concatenate texture/mesh sequences and to insert a composed facial performance in a target video.**

***Index Terms*—video editing, facial expression re-targeting, deformable surface tracking, image based rendering**

I. Introduction

FACIAL performances are one of the most important aspects for conveying emotion and information in visual media such as movies, videos and games. The human visual system is highly sensitive towards the subtleties of non-verbal facial communication and, in consequence, exceptionally good at spotting artifacts or inconsistencies in facial video sequences that have been altered by any means. While state-of-the-art approaches to facial video editing have crossed the uncanny valley[1] of creating believable facial animations, these approaches are still limited to high-profile productions because of the equipment and manual labor they require. In this paper, we present an approach that allows to carry out fundamental editing tasks with convincing results using standard equipment and very limited manual intervention. Based on a multi-view input stream of two or more synchronized cameras, the system enables an editor to perform the following tasks:

- Replacement of one facial performance with another while leaving the rest of the video, including global head motion of the actor, untouched
- Creating novel facial performances that have not been captured by concatenating pieces of facial actions in a motion-graph like way, as well as merging different parts (eyes, mouth etc.) of different recorded facial actions

The seamless transfer of a piece of recorded facial performance into a target sequence displaying the same actor most importantly requires the possibility to change the head pose in the recorded sequence while retaining as much detail of the recorded performance as possible. It also requires precise knowledge of how and where to place this piece of performance in the target sequence. Furthermore, the transition boundary between original and inserted content must be virtually invisible. Moreover, in order to switch from one sequence of facial performance to another, the transition in time must be a credible facial motion and not contain temporal artifacts like jumping, ghosting or unnatural intermediate expressions. The proposed system is mainly designed to support intra-person performance editing, e.g. shooting several takes of the same scene and then using the captured video footage to remove small errors and create a flawless video during postprocessing. While it is still possible to transfer the facial expression from one person to another the application area for inter-person performance editing is more restricted as the target and source face should be of similar size and proportion. A possible application for inter-person performance editing would be for example to replace the face of a stuntman by the face of the actor or to create a twin of a character.

In order to enable this seamless and realistic exchange of facial performance sequences, we employ a hybrid approach in which an approximate geometric model of the face is tracked by a method for temporally consistent deformable surface tracking, allowing us to create a spatially normalized representation of each sequence of facial performance to be used in editing. These representations are stored in the texture space of the geometry model. Most of the principal deformations are modeled by movement and deformation of the underlying geometry and only details like stretching skin, wrinkles, occlusions and disocclusions as well as fast micro-expressions which are very hard to correctly capture in geometry are represented in the resulting texture sequences. Via the geometry model, several synchronous video streams are integrated into a single texture sequence using a modified version of a state-of-the-art multi view texture extraction method [4]. In the target sequence, the tracked geometry allows the identification of

M. Kettern and A. Hilsmann are with Fraunhofer Heinrich Hertz Institute, Berlin 10587, Germany (e-mail: markus.kettern@hhi.fraunhofer.de, anna.hilsmann@hhi.fraunhofer.de).

W. Paier and P. Eisert is with Fraunhofer Heinrich Hertz Institute, Berlin 10587, Germany and also with the Humboldt University, Berlin 10099, Germany (e-mail: wolfgang.paier@hhi.fraunhofer.de, peter.eisert@hhi.fraunhofer.de).

[1]As early as 1970, Masahiro Mori envisioned the concept of what was later termed the 'uncanny valley': As robots or other avatars mimicking human appearance and behavior become more and more realistic, the emotional response of most observers will rapidly drop from positive reactions to rejection and repulsion at a certain point and only at a very high level of realism again turn back to acceptance. Since then, the concept has been profoundly based on several psychological explanations as well as confirmed by several studies [1], [2], [3].

the head pose which must be assumed by inserted content. On the other hand, the tracked geometry of the sequence to be inserted allows to render this content correctly into the image. Moreover, the tracking provides semantically consistent models throughout the sequences, allowing the calculation of a similarity measure for pairs of single frames from the recorded sequences, thus providing means to chose an optimal transition point between inserted and original sequence or for concatenating several sequences in order to form a longer one.

A. Contributions

In this paper we present an approach for facial re-animation that makes use of image-based rendering techniques to transfer facial actions/expressions from multiple source videos to a target video. These facial actions are concatenated seamlessly, thus enabling an editor to create photo-realistic modifications to an existing video sequence without much additional effort. We improved upon the system presented in [4] by employing a new deformable surface tracking method [5] which allows greater flexibility in re-rendering facial videos to a different viewpoint with a different head pose. Furthermore, we extended the blending methods used in [4] to work with dynamic geometry-plus-texture sequences.

II. Related Work

A. Facial Geometry Tracking

Capturing and tracking facial performances is an important topic in computer vision and many of its applications. One of the very first research efforts towards facial performance capture in 1972 [6] was already dedicated to the creation of realistic facial CGI models which would then be animated by a method related to today's blend shapes. In more than four decades, facial performance capture and tracking has been actively researched and has developed in two main directions, namely marker-based and marker-less performance capture. Marker-based capture has matured over the last two decades and is available in numerous industrial products as it is highly robust and computationally tractable. Shortcomings of marker-based systems are the loss of expression details due to the fact that the sparse feature information that is actually tracked cannot fully represent the subtle details that comprise a credible facial performance. These have to be added back either manually or by using highly detailed parametric geometry models driven by the markers [7].

Since the reuse of the captured textures is essential for our editing approach, we have to resort to a marker-less tracking approach. The tracking output can e.g. be used to drive a parametric model of the actor's face [8], [9]. In many publications, tracking information is used to drive an animatable CGI head model (e.g. [10]). The challenge of exploiting tracking information in order to edit facial performances is unrelated to these works which rather use tracking as a complex input mechanism. Several recent approaches aim at reconstructing facial geometry and dynamics from monocular input [11], [12] but are bound to use generic parametric face models as priors to overcome the problems arising with such unconstrained visual input which decreases the flexibility in adapting to the subjects facial geometry.

As we want to capture and edit the full performance in all its details, we employ a dense surface tracking method that estimates the motion and deformation of a whole surface as represented by a triangle mesh instead of sparse feature locations. Most approaches to this type of facial performance capture rely on computing a 3D-reconstruction of the face in each single frame, either using depth sensors [13], [14] or directly from the multi-view or stereo video stream [15], [16], [17]. In these approaches, the tracking problem is approached by computing correspondences between semantically inconsistent meshes (one per frame), often using image correspondences between the frames (sparse or dense) as a guidance. One of the main problems of these approaches is the accumulation of drifting errors between the texture and the geometry layer. Since the relation between images and meshes is basically reset at each frame, semantic consistency between the layers has to be enforced by additional steps.

Similar to our approach, [18] use an analysis-by-synthesis approach in order to avoid this form of drift as much as possible. Tracks of feature points are employed in order to deform a template mesh using Laplacian smoothness constraints (among other terms).

In [5], we proposed an analysis-by-synthesis tracking approach which is the basis of the method used in this work. A deforming template mesh is used to track the motion and deformation of a face using optical flow and several layers of Laplacian mesh regularization. In order to increase robustness against occlusions, disocclusions as well as local and global illumination changes in the analysis-by synthesis approach, a photometric component is incorporated to the tracking model. Similar photometric components have already been used in 2D deformable tracking [19].

B. Performance Synthesis

Creating synthetic human characters and especially human faces is one of the most challenging tasks in computer graphics. Geometric and reflective properties are hard to model, furthermore, humans are very good at interpreting faces, such that even slight deviations from the expected visual appearance are perceived as wrong or unnatural facial expressions. As a consequence, many approaches apply image based rendering techniques to create photo realistic renderings of humans [20], [21], [22], [23], [24], [25], [26], [4]. For example Borshukov et al. [20] use a highly sophisticated capture setup consisting of 8 infrared cameras plus 3 synchronized, high definition color cameras to capture an actor's facial performance in an ambient lighting setup. Based on 70 small retro reflective markers, they drive a previously scanned 3D model of the actor's face to obtain a time-consistent animation mesh and to extract dynamic textures. Mesh sequences and dynamic textures are then used to perform a motion graph like animation of the actor's face. They achieve highly realistic results, but in contrast to [20] we also want to provide a framework which is not only affordable by high-budget productions. In [27], Dale et al. present an affordable system that is capable of transferring the facial

performance of a person between video sequences. They use a multilinear face model to track the actor's facial geometry which enables them to transfer a facial performance even between different persons. Finally, they use a dynamic programming approach for retiming and gradient domain blending to seamlessly integrate a facial performance in the target video. Recently Thies et al. [28] presented an impressive system for realtime expression transfer and facial reenactment. They use a linear model that represents identity, facial expression and albedo which allows for a realistic expression transfer between different individuals captured from different viewing angles. To cope with changing light situations, they use a Spherical Harmonics basis to model diffuse lighting caused by distant light sources. A facial performance is transferred by modifying the facial expression parameter of the target sequence. While their approach produces impressive results, they focus more on realtime performance and a compact representation of the captured facial performance. Linear models are typically a good choice for realtime applications as they represent a scene with a compact parameter set but they also tend to miss fine details. A similar system was proposed by Xu et al. in [29]. They developed a method for facial performance transfer with additional user constraints. Based on a multiscale mesh decomposition they are able to transfer facial performances from one person to another or even to an animal, as well as performing user guided manipulation of fine details (e.g. wrinkles). Though both previous works also tackle the problem of facial performance transfer our approach differs as we are focusing on creating novel facial videos by seamless concatenation of multiple performances and integration in a target video.

Pushing the idea of image based rendering further, Xu et al. [30] presented a system for the synthesis of novel full body performances from multi-view video. They use performance capture to obtain pose and geometry for each video frame. Based on this data, a synthetic video performance is rendered according to a user provided query viewpoint and skeleton-sequence, even if the exact body pose is not represented in the database. However, they also explain that while the approach is appropriate for skeletal animation, facial animation has to be handled separately.

As humans are very sensitive to inconsistencies in the appearance of other human faces, we specifically concentrate on facial re-animation. Inspired by the aforementioned advances in image- and video-based rendering, our approach is based on real video footage to achieve photo realistic results. Video-based facial animation has only recently found attention in the literature. Li et al. [31] proposed a purely image based system for facial expression transfer based on video databases for both, the source and target character. They implemented a new similarity measure to find candidate frames in the target subject's video database. Additionally, they use a correspondence map and optical flow field to create a synthetic version of the query frame by warping the target's neutral expression. Then they combine the synthetic and the retrieved images to form a realistic and temporally consistent video of the target person performing the queried facial performance. While this approach can be used to synthesize a new performance of a frontal face it's necessary to have already a temporally consistent source sequence that can be mimicked by the system. We rather want to provide a tool to synthesize a realistic video, by concatenating facial expressions from multiple source videos which are not temporally consistent, independent and were captured potentially by multiple cameras.

Paier et al. [26] presented a system for facial re-targeting, i.e. transferring short sequences of a facial performance between different videos. Based on rigid facial geometry tracking, this method can be used to fine tune the timing of facial performances or to exchange similar facial expressions as long as the viewpoint in source and target sequence is very similar. In [4], they improved their system by using a low complexity blend-shape model (i.e. 1 blend shape to open/close the mouth) for tracking the rigid head pose and large scale deformations of the jaw. In contrast to [26], we use a deformable surface tracking approach that allows much bigger changes in viewpoint because of the more accurately approximated geometry. Thus it grants greater freedom in the choice of sequences used for performance replacement. Moreover, we also focus on synthesizing novel sequences from several short clips of facial performance with credible transitions in both geometry and texture layer being created on demand. Therefore, we developed a convenient heuristic for automatic transition search and extended the blending method presented in [4] for the usage with textured geometry streams.

Our performance synthesis strategy is also related to the idea of motion graphs [32], [33], [34] that have already been successfully used for skeletal or surface-based animation of human characters. We capture several video sequences of a facial performance and split them up into short clips that contain single actions or facial expressions (e.g. smile, talk, looking surprised or angry). These clips are transformed to our hybrid representation and may then be concatenated in almost any order to compose a novel facial performance. Similar to [35], [36], [4], we also find smooth transitions between different facial sequences because directly switching from one sequence to another would create obvious artifacts in the synthesized facial video (e.g. sudden change of facial expression and sudden changes of illumination). For this purpose, we use a geometric image registration technique to compensate for changes in the facial expression as well as a modified cross dissolve to smoothly blend all remaining color differences.

III. Method Overview

The presented approach consists of two main steps: facial performance analysis and performance synthesis/editing. We use the video streams of two or more synchronous and calibrated cameras as input. Sequences to be edited are chosen and one frame per sequence (with neutral facial expression) is picked as a key frame. Furthermore, a static 3D model of the actor's head is needed as initial geometric proxy for deformable surface capture. This model can be acquired using any 3D-reconstruction method available, the models used in this paper were created using a standard multiview-stereo approach [37]. In order to enable mouth animation, a mouth

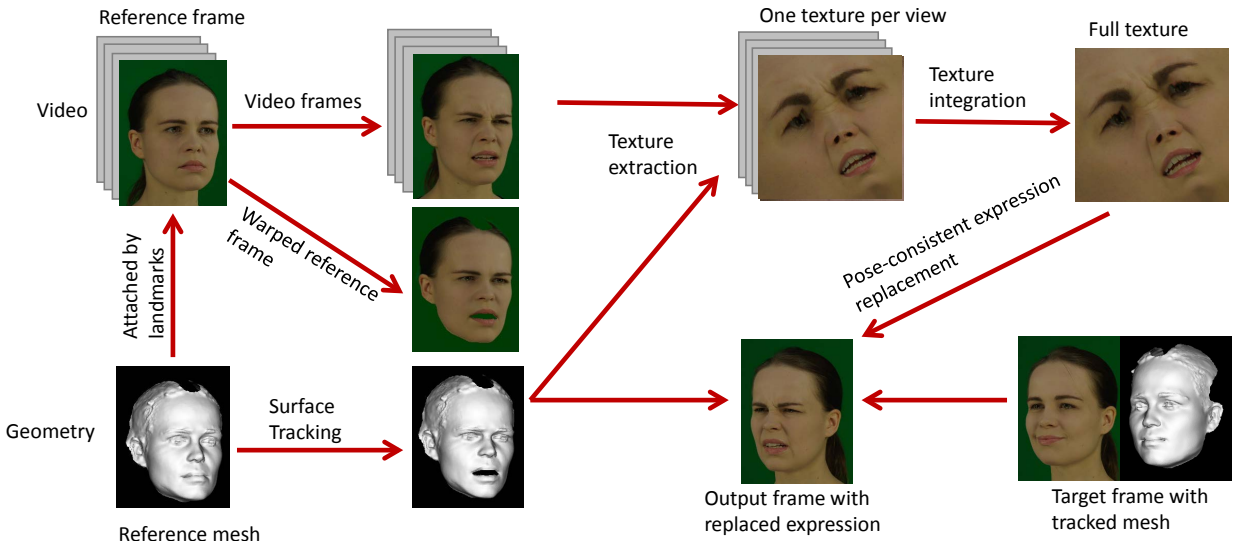

Figure 1. System overview: The reference mesh is aligned to the reference video frame via landmarks and is tracked using our deformable surface tracking method. Then, dynamic textures are extracted and integrated over all available views. Finally, the tracking information in the target video is used to correctly place and render the desired expression into the target video frame.

seam is cut into the model using standard editing software (e.g. blender). For fitting the model to the images at the key frames, about 10 vertices are manually picked as landmarks. Their counterparts in the images can be found by a facial feature detector such as [38].

During performance analysis, the geometric proxy mesh is tracked along the input streams using the method described in section IV. Afterwards, dynamic textures are extracted by unwrapping the input streams along the texture coordinates of the temporally consistent model. Several video streams can be integrated into one dynamic texture sequence as described in section V.

In the editing step, the processed facial performance sequences can be used to manipulate a target video sequence where the actor has been tracked using the same tracking model and method. Possible editing tasks include changing the facial expression as a whole or just exchanging parts of the face (e.g. the eyes) but also concatenation and/or looping of different source sequences in order to create a new facial performance sequences that were not captured previously. Based on the tracked meshes and extracted dynamic textures, the desired facial performance is synthesized as detailed in section VI. As the presented system supports multiview video footage, the performances can be rendered from different viewpoints, enabling replacement in a wide range of target sequences. Finally, several blending techniques are employed to seamlessly fuse the concatenated/looped facial performances at geometry as well as texture level and to smoothly integrate the resulting facial performance in the target video.

Figure 1 presents a symbolic overview of the system including tracking the mesh in the video stream, extracting and integrating the dynamic textures, and replacing the expression in a target video by rendering the texture into the frame.

IV. Facial Geometry Tracking

Many approaches to realistic facial performance editing rely on static meshes or very low-dimensional blend shape models for providing an approximate geometry model for the face [26], [39]. While providing highly robust tracking methods, these approaches are limited in terms of the variability of facial deformations they can represent geometrically and thus have to represent all variations beyond their limited geometric expression space in the texture. However, when changing the viewpoint, regions where the true geometry is approximated too coarsely will exhibit visual artifacts since the texture is being rendered onto the wrong 3D location. On the other hand, many state-of-the-art methods for performance capture yield details on the level of wrinkles and pores. However, most of them rely on one individual, temporally unaligned 3D reconstruction per frame and treat tracking as a mesh correspondence problem where the images are used to guide the solution. Over time, nearly all approaches tend to accumulate drift between texture and mesh which has to be specially treated.

We employ a freely deformable 3D tracking approach regularized by imposing Laplacian smoothness penalties on the deformation offsets, similar to [18], [19]. In order to prevent the texture from floating over the geometry, we resort to an analysis-by-synthesis approach for tracking: the mesh deformations for each frame are estimated by minimizing an image-based cost function between the frame and the first frame of the sequence warped by the accumulated deformations estimated for all previous frames. In this way, the relation between geometry and texture stays constant throughout the whole sequence, which is necessary since we want to exploit this semantic consistency when using our texture-based approach for performance editing. Only appearance changes that the tracker cannot adapt the geometry to have to be represented in the texture. These changes usually include wrinkles and

other high-frequency details that are not present in the key frame as well as disocclusions in the mouth and eye regions.

One of the main problems of analysis-by-synthesis methods is posed by changes in illumination, e.g. occurring from movement and rotation of the head relative to the light sources. The interaction between human facial surfaces and lighting is highly complex and hard to model realistically. However, since we rely on the recorded textures to reproduce fine details such as this interaction, we mainly need our tracking algorithm to be robust against the effects induced by illumination changes which can severely hamper tracking quality since they introduce low-frequency brightness changes the tracker will try to compensate for by adding unwanted deformations. In order to overcome this problem, we use an additional photometric warping component that adapts the brightness of the warped key frame to better match the levels of the target frame. Since this adaption is done for each vertex, this component can reproduce nearly arbitrary modulations of appearance and thus has to be regularized on its own if we want to prevent it from incorporating appearance changes that are caused by geometric deformations which should be estimated by the tracker. The photometric component also makes the tracking more robust against high-frequency details like wrinkles that we aim at expressing in the texture layer. Figure 2 displays some example tracking results from a data set recorded with two cameras.

A. Tracking Methodology

Throughout the rest of the paper, we will use i as index for the individual views/cameras, k for indexing the vertices $\mathbf{v}_k$ of the tracking mesh with a total of K vertices stored in the $K\times 3$-matrix $\mathbf{V}$, and superscript t as index for time points. The image of camera i at time t is given by $\mathcal{I}_i^t$. The purpose of our analysis-by-synthesis-based geometry tracking is to find one set of vertex positions $\mathbf{V}^t$ per time instant which minimizes an image-based error function

$$\epsilon_{img}^t = \sum_i \Phi\left(r_i^t\right) \quad (1)$$

$$r_i^t(\mathbf{p}) = \mathcal{I}_i^t(\mathbf{p}) - \mathcal{J}_i^t(\mathbf{p}) \quad (2)$$

$$\mathcal{J}_i^t = \mathcal{W}_i\left(\mathcal{I}_i^0, \mathbf{V}^t\right) \quad (3)$$

where Φ is a suitable norm-like function, $\mathbf{p}$ denotes an image pixel position, and $\mathcal{W}_i\left(\mathcal{I}_i^0, \mathbf{V}^t\right)$ is a view-dependent *warping function* which deforms the first image of the sequence, $\mathcal{I}_i^0$, according to the vertex positions $\mathbf{V}^t$. Thus, $\mathcal{J}_i^t$ is a warped version of the key frame $\mathcal{I}_i^0$, created by means of the deforming tracking mesh as explained below. Finally, $r_i^t(\mathbf{p})$ is the *residual function* corresponding to ϵ_i^t which is evaluated on all pixels covered by the tracking mesh in image $\mathcal{J}_i^t$.

B. Mesh-based Image Warping

We assume that $\mathbf{V}^0$, the tracking mesh in its starting form, is aligned to the images $\mathcal{I}_i^0$ for all views i which can be achieved e.g. by a rigid matching of manually selected landmarks on the mesh to the output of a facial feature detector like [38]. The quantity we want to estimate in geometry tracking are the positions of the vertices in time point t, as given by $\mathbf{V}^t$.

If we rasterize a hypothesis for the mesh $\mathbf{V}^t$ onto the image plane of camera i, each pixel $\mathbf{p}$ is associated with one triangle $T(\mathbf{p})$ and its barycentric coordinates β respective to this triangle are given by

$$\mathbf{p} = \sum_{k\in T(\mathbf{p})} \beta_k^t \mathbf{u}_k^t \quad (4)$$

where $\mathbf{u}_k^t$ is the projection of vertex k at its original position $\mathbf{v}_k^t$ onto the image plane of camera i.

In order to infer the color of pixel $\mathbf{p}$, we sample the key frame $\mathcal{I}_i^0$ at the position defined by (4):

$$\mathcal{J}_i^t(\mathbf{p}) = \mathcal{I}_i^0(\mathbf{p}') \quad (5)$$

$$\mathbf{p}' = \sum_{k\in T(\mathbf{p})} \beta_k^t \mathbf{u}_k^0 \quad (6)$$

Since large parts of the motion in a natural performance may be due to movement of the complete head instead of deformation, we aim at explaining as much as possible of the observed changes by a rigid motion of the head. In this way, we obtain two layers of motion: rigid head movement and facial deformation. This is useful for performance editing because in most cases we will want to exchange the facial deformation in a sequence while keeping the original head pose trajectory. To this end, we compose $\mathbf{V}^t$ from a rigid and a non-rigid component by

$$\mathbf{V}^t = \mathbf{R}^t\left(\mathbf{V}^0 + \triangle\mathbf{V}^t\right) + \mathbf{T}^t \quad (7)$$

where $\mathbf{R}^t$ is a rotation matrix which will be parametrized by Euler angles r_x, r_y, r_t, $\triangle\mathbf{V}^t$ are the vertex offsets representing the deformation layer, and $\mathbf{T}^t$ is a rigid offset matrix that adds the same vector $\mathbf{t}^t$ to each vertex. In order to estimate these quantities, we employ two tracking steps per frame. In the first step, we estimate the rigid head pose changes by inserting

$$\hat{\mathbf{V}}^t = \mathbf{R}^t\left(\mathbf{V}^0 + \triangle\mathbf{V}^{t-1}\right) + \mathbf{T}^t \quad (8)$$

into equation (3) and minimizing the corresponding error function (1) over the 6 parameters of the rigid head motion r_x, r_y, r_t and $\mathbf{t}^t = \left[t_1^t\, t_2^t\, t_3^t\right]^T$. This is done efficiently using standard dense Gauss-Newton optimization where we add a small amount (0.001) of zero-order Thikonov parameter regularization in order to keep the numerically volatile rotational component stable.

In the second step, we minimize the remaining image difference between $\mathcal{I}_i^t$ and $\mathcal{J}_i^t$ by estimating the vertex offset vector $\triangle\mathbf{V}^t$ in (7) and keeping $\mathbf{R}^t$ and $\mathbf{T}^t$ fixed with the additional components described in the following.

C. Photometric Component

Since we use the key frame $\mathcal{I}_i^0$ as warping reference throughout the whole sequence, changes in illumination and other appearance variations like disocclusions and wrinkles can lead to deformation artifacts since there is no geometric modification that will allow these structures to appear in the rendered image. Similar to [40] we compensate for these variations by introducing a photometric component to our warping function $\mathcal{W}$. This is achieved by attaching an intensity

factor s_k^t to each vertex and changing the image sampling equation (5) to

$$\mathcal{J}_i^t(\mathbf{p}) = \mathcal{I}_i^0(\mathbf{p}') \sum_{k \in T(\mathbf{p})} \beta_k^t s_k^t \tag{9}$$

Since the photometric component has one value per vertex, its resolution is typically lower than the image resolution. It is able to express arbitrary appearance variations up to a certain frequency which depends on the size of the mesh triangles in the image. The impact of higher frequencies on the geometry estimation will be minimal since they will be contained within a single triangle in at least one direction (e.g. wrinkles) and thus be smoothed over by the inherent regularization of this mesh-based approach. The initial values s_k^0 are set to 1 and the values for the subsequent frames are estimated jointly with the vertex offsets $\triangle\mathbf{V}^t$.

D. Regularization

In order to increase robustness towards outliers occurring e.g. at highlights, disocclusions and especially in weakly textured regions, we add several regularization terms in order to prevent the occurrence of geometric artifacts and also to keep the mesh as closely to its initial shape as possible while following all relevant cues for deformation. The regularization is achieved by penalizing deviations in the *Laplacian differential vectors* $\boldsymbol{\delta}^t$ which we define using the Graph Laplacian of the mesh [41]:

$$\boldsymbol{\delta}_k^t = \mathbf{v}_k^t - \frac{1}{|N_k|} \sum_{j \in N_k} \mathbf{v}_j^t \tag{10}$$

$$\boldsymbol{\delta}^t = \begin{bmatrix} \boldsymbol{\delta}_0^t \\ \vdots \\ \boldsymbol{\delta}_{K-1}^t \end{bmatrix} \tag{11}$$

where N_k denotes the 1-neighborhood of vertex $\mathbf{v}_k^t$. The error function we use for geometric regularization is given by

$$\epsilon_{reg}^t(\mathbf{V}^t) = \lambda_1 \Phi(\boldsymbol{\delta}^t - \boldsymbol{\delta}^0) + \lambda_2 \Phi(\boldsymbol{\delta}^t - \boldsymbol{\delta}^{t-1}) \tag{12}$$

where the first term penalizes deviations from the initial shape and the second term regularizes the mesh development over time. Note that the values for λ_1, λ_2 may change with the mesh resolution. Similarly, we regularize the photometric component over time using Laplacian differentials

$$\rho_k^t = s_k^t - \frac{1}{|N_k|} \sum_{j \in N_k} s_j^t \tag{13}$$

$$\boldsymbol{\rho}^t = \begin{bmatrix} s_0^t \\ \vdots \\ s_{K-1}^t \end{bmatrix} \tag{14}$$

$$\epsilon_{phot}^t(\mathbf{s}^t) = \lambda_3 \Phi(\boldsymbol{\rho}^t - \boldsymbol{\rho}^{t-1}) \tag{15}$$

E. Optimization and Optical Flow

The complete objective function for the deformation estimation step of our geometric tracking is given by the sum of (1), (12) and (15), parametrized by the vertex offsets $\triangle\mathbf{V}^t$ and the values for the photometric component $\mathbf{s}^t$:

$$\epsilon^t(\triangle\mathbf{V}^t, \mathbf{s}^t) = \epsilon_{img}^t(\mathbf{V}^t, \mathbf{s}^t) + \epsilon_{reg}^t(\mathbf{V}^t) + \epsilon_{phot}^t(\mathbf{s}^t) \tag{16}$$

In order to weaken the influence of outliers and to improve localization of discontinuities in the regularized meshes, we use the Charbonnier penalty function

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

The objective function is minimized using *iteratively reweighted least squares* which is an approximation of a Gauss-Newton scheme for robust estimation [42], [43]. Since this optimization scheme exploits the Jacobian matrix in order to generate a system of (weighted) normal equations in each iteration, we need to compute the Jacobian of our objective function. The partial derivatives of all transformations can be computed analytically except for the x and y derivatives of the images. These are approximated by the image gradients which relates our tracking approach to optical flow estimation.

As suggested in [44], we use a blend of the gradients of the rendered image $\mathcal{J}_i^t$ and the target image $\mathcal{I}_i^t$ to replace the gradient of $\mathcal{J}_i^t$ in derivative computations:

$$\nabla\mathcal{J}_i^{t*}(\mathbf{p}) = \frac{1}{2}\left(\nabla\mathcal{J}_i^t(\mathbf{p}) + \nabla\mathcal{I}_i^t(\mathbf{p})\right) \tag{18}$$

Since the motion of each pixel is dependent only on the estimated motion of the mesh triangle containing it and we have one residual value per pixel (or three if we use RGB images), the system of normal equations for optimization is sparse. In order to solve it efficiently, we resort to a state-of-the-art sparse Cholesky matrix factorization.

V. Dynamic Textures

The presented robust face tracking method provides a stream of temporally consistent triangle meshes that accurately follow the subject's facial geometry even in presence of large deformations. However, since fine scale details (e.g. small deformations, wrinkles, shades) cannot be captured efficiently by geometry alone we extract a dynamic texture (i.e. a sequence of textures) for each video sequence. The dynamic texture does not only contain static fine scale details like pores, it also captures fine scale motions, wrinkles or shades that would be lost when using a static texture. Another advantage of using dynamic textures is the convenient integration into existing rendering engines and the natural support for temporally consistent editing tasks, as they are motion compensated.

A. Extraction of Texture sequences

Based on the tracking results we want to extract a sequence of textures from the multiview video stream that optimally represents the subject's face texture during the whole video sequence. Extracting dynamic textures is a typical multi-label problem (see [45], [46], [25], [26], [4]), i.e. assign a source camera to each triangle from which it should receive its texture. To ensure a high visual quality, we enforce three quality criteria:

- high spatial resolution

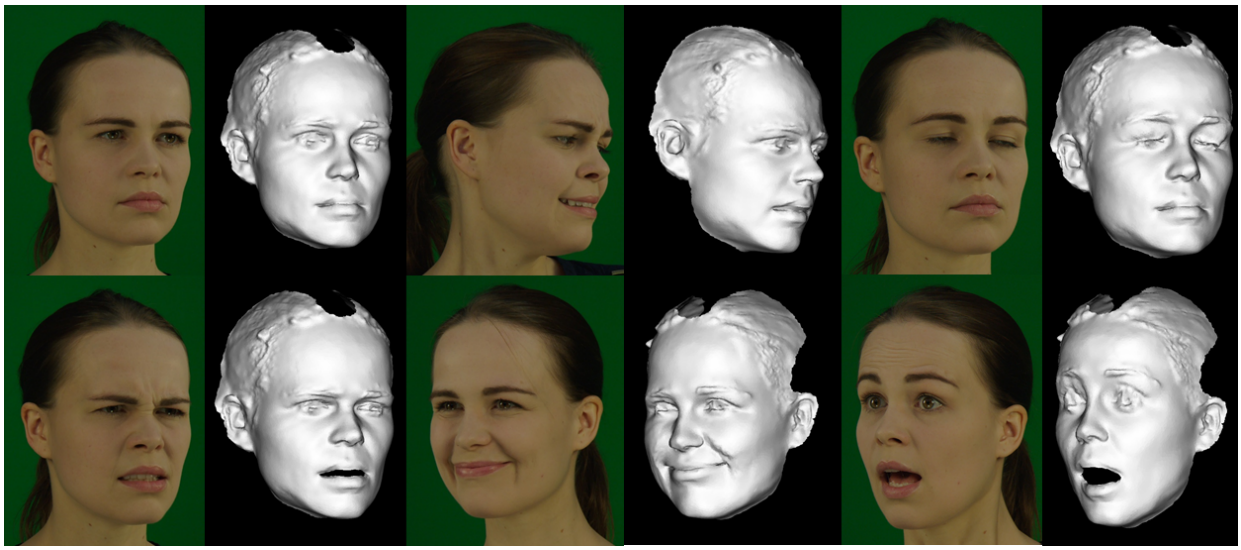

Figure 2. Example results of deformable face tracking: Target frames and flat shaded renderings of the tracking mesh.

- spatial consistency: low visibility of spatial seams, i.e. no visible seams within a texture
- temporal consistency: low visibility of temporal seams, no flickering seams between consecutive textures (e.g. constantly changing source camera near a seam)

Formulating these requirements as a discrete optimization task yields the following error function:

$$\begin{aligned}\epsilon(I) = &\sum_{t}^{T}\sum_{n}^{N}\mathcal{D}(n, i_n^t) \\ &+ \lambda \sum_{n,m\in\mathcal{N}} \mathcal{V}(i_n^t, i_m^t) \\ &+ \eta\mathcal{U}(i_n^t, i_n^{t-1})\end{aligned} \tag{19}$$

where I denotes the set of source camera labels for each face region n. All triangles in a region n receive their texture from the same source camera. This implicitly increases the spatial smoothness of the texture mosaic but more importantly drastically reduces the computational complexity in case of long video sequences with high polygon count 3D models.

The first term $\mathcal{D}(n, i)$ measures the spatial resolution of a region n textured by camera i:

$$\mathcal{D}(n,i) = \begin{cases} 1 - \mathcal{A}(n,i) & n \text{ is visible} \\ \infty & n \text{ is occluded} \end{cases}, \tag{20}$$

$$\mathcal{A}(n,i) = \frac{area(n,i)}{\sum_j area(n,j)}, \tag{21}$$

with $\mathcal{A}(n, i)$ being the area of n projected on the image plane of camera i relative to the sum of $area(n, i)$ over all possible cameras to ease the choice of the weighting factors η and λ.

The second term $\mathcal{V}(i_n, i_m)$ in (19) penalizes spatially non-smooth solutions and increases the overall cost (19) by the sum of color differences along the border $e_{n,m}$ of two adjacent regions n and m that are textured from two cameras i_n and i_m.

$$\mathcal{V}(i_n, i_m) = \begin{cases} 0 & i_n = i_m \\ \Pi_{e_{n,m}} & i_n \neq i_m \end{cases} \tag{22}$$

$$\Pi_{e_{n,m}} = \int_{e_{n,m}} \|\mathcal{I}_{i_n}(x) - \mathcal{I}_{i_m}(x)\|\, dx \tag{23}$$

Finally, a temporal smoothness term $\mathcal{U}(i^t, i^{t-1})$ (24) increases the overall cost by a constant η for each region n with a changing source camera i between two consecutive time steps.

$$\mathcal{U}(i^t, i^{t-1}) = \begin{cases} 0 & i^t = i^{t-t} \\ 1 & i^t \neq i^{t-t} \end{cases} \tag{24}$$

Without such a term, the resulting dynamic textures are not temporally consistent, i.e. the source camera of a certain region can change arbitrarily between two consecutive texture frames resulting in visually disturbing flickering in the extracted texture sequence.

We use the alpha-expansion method [47] to efficiently find a close-to-optimum approximate solution for the objective function. Finally, a global color matching [48] together with Poisson blending [49] modified for the usage in texture mosaics are employed to conceal remaining seams.

VI. Concatenation of Sequences

The methods presented in the two previous sections create an independent set of mesh-plus-texture sequences where each sequence represents a certain action or facial expression. In this section, these independent sequences are brought into connection by defining transition rules between individual sequences for later animation.

By looping and concatenating several sequences, new and more complex sequences can be synthesized. By rendering such a sequence to a target video, we can exchange the facial

performance of a subject in a video. This type of animation strategy is related to motion graph based animation techniques [32], [33], [34], [20]. In the context of motion graphs, edges in the graph correspond to facial actions, and vertices to expression states. Since the extracted sequences have been captured separately and in a different order, simple concatenation would create obvious visual artifacts during transitions between two sequences. These artifacts appear in geometry as well as in texture due to different facial expressions and changing illumination (e.g. caused by head movement), see figure 4. We do not want to rely on dedicated transition sequences in order to switch from one facial action to another as this would drastically increase the number of necessary source sequences for editing. Instead, good transition points between sequences are defined automatically in a given search window at the end of the current sequence and the beginning of the next sequence. The user defines the search window based on the relevance of the content (i.e. the important content should remain outside the search window) and on how accurate the user wants to determine the transition (e.g. a large transition window gives less control over the transition point while a small transition window allows to closely specify when to switch from one sequence to another). In order to find a suitable transition frame pair i, j a heuristic based on geometry and texture information is used (25).

$$\mathcal{H}(t,q) = \frac{\|\triangle \mathbf{V}^t - \triangle \mathbf{V}^q\|}{|\mathbf{V}|} + \lambda \frac{\|\mathcal{T}_t - \mathcal{T}_q\|}{|\mathcal{T}|} \qquad (25)$$

Although geometric similarity gives a very good hint for suitable transitions frames, textural information is still valuable as it can distinguish between expression states that are not reflected in the geometry (e.g. wrinkles and disocclusions).

With t being the frame in the current sequence and q being the frame in the next sequence, $\mathcal{H}$ is a heuristic based on the mean color difference of face textures $\mathcal{T}$ and the mean difference of vertex offsets $\triangle \mathbf{V}$ between the tested frames t, q provided by the deformable face tracking. For the calculation of the texture match, down sampled versions of size 256x256 are used to speed up the search, while still providing enough details. The following two subsections describe the blending strategies for geometry as well as textures sequences.

A. Geometry Blending

As the geometry deforms over time, simple concatenation of independent mesh sequences leads to sudden jumps during the transition. Therefore, the vertex offsets $\triangle \mathbf{V}^t$ are blended linearly during the transition over a fixed number of frames (e.g. $n = 30$). We calculate a blending speed factor α for each vertex to account for the fact that not all regions of the face move equally fast. While, for example, the mouth can move rather quickly, other regions (e.g. forehead, temporal region) should not expose a noticeable motion/deformation at all. However, due to slight inaccuracies, even rigid areas in the mesh will deform slightly over time which leads to visible motions during the transition. The blending speed α will distribute the morphing process of rigid vertices over a large number of frames (making it hardly noticeable), while less rigid regions (e.g. mouth) can be blended over a short time period. The blending speed is estimated for each vertex based on the average difference of vertex offsets (i.e. vertex motion adjusted for the rigid motion component) $\left\|\triangle \mathbf{v}_k^{t+1} - \triangle \mathbf{v}_k^t\right\|$ between consecutive frames. In order to get a spatially smooth α, the change in a vertex offset contributes to the α for all vertices in its neighbourhood (see figure 3). To gain more control over the non-uniformity, α can be estimated for example using only the fastest, slowest or all motions of a vertex (see figure 3). Finally, α is normalized so that the slowest vertex reaches its target position after n steps and the fastest vertex reaches its final position after $\frac{n}{\alpha}$ frames.

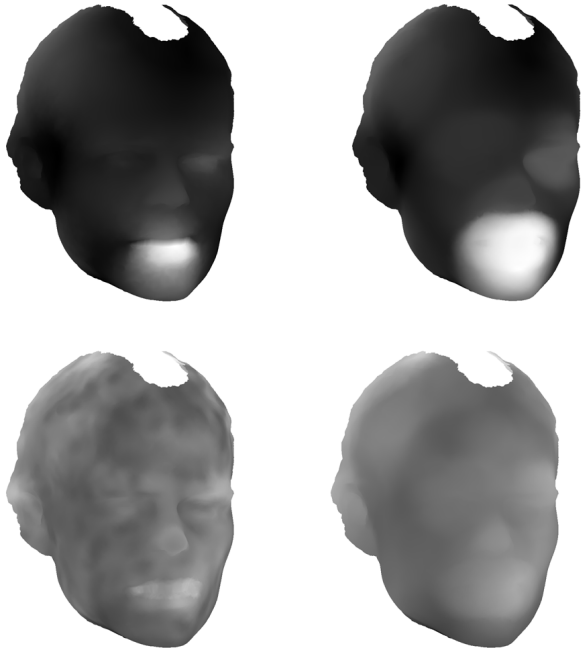

Figure 3. 3D head model rendered with α weights as intensities. To improve visibility,α weights are scaled to the domain [0,1] in this image. Left column: neighbourhood radius = 0.005, right column: radius = 0.02. Using only slow motion samples creates an almost uniform blending speed (bottom row). Using fast motion samples to estimate α results in a strongly non uniform blending speed (top row).

B. Texture Blending

Between texture sequences, a two stage blending strategy is employed: first, the spatial misalignment between the last frame $\mathcal{T}_{last,t-1}$ of the previous texture sequence and the first frame $\mathcal{T}_{first,t}$ of the next sequence is corrected, before the remaining color/intensity differences are blended by a cross dissolve.

1) Spatial Texture Blending: The spatial misalignment is estimated by calculating a 2D warp $\mathcal{W}(\mathcal{T}, \Phi)$ that maps $\mathcal{T}_{last,t-1}$ on $\mathcal{T}_{first,t}$, minimizing

$$\underset{\Phi}{argmin} \left\|\mathcal{T}_{first,t} - \mathcal{W}(\mathcal{T}_{last,t-1}, \Phi)\right\|^2 + \lambda \mathcal{E}_{reg}(\Phi), \qquad (26)$$

with $\mathcal{E}_{reg}$ being a regularizing term weighted by a scalar factor λ. Similar to [19], the image deformation of $\mathcal{T}_{last,t-1}$ with regard to $\mathcal{T}_{first,t}$ is modeled as a regular deforming 2D

Figure 4. Impact of geometric warp during transition. Bottom-left: previous frame, bottom-right: current frame. Middle-left: 50% cross dissolve without geometric warp (artifacts around the lips and the eyes), middle-right: 50% cross dissolve after geometric warp. Top-left: color difference before geometric warp, top-right: after geometric warp (no strong edges are visible around eyes and mouth.).

control mesh with Barycentric interpolation between vertex positions, i.e. the warping function is parametrized by a vector Φ containing the control vertex displacements.

Based on the estimated warp, we gradually deform the motion in the last frames of $\mathcal{T}_{\dots,t-1}$ and the first of $\mathcal{T}_{\dots,t}$ to ensure that the transition frames of both sequences are identical. This deformation process is distributed over several frames. In our experiments, a rather high number of frames $n = 60$ (at 59 fps) is used to perform the spatial deformation because the low additional motion per frame makes it barely noticeable.

2) Anisotropic Cross Dissolve: The spatial texture alignment largely reduces ghosting artifacts during blending (see figure 4). However, small details (e.g. wrinkles that appear or disappear), changing light conditions and remaining misalignment that could not be fully compensated by the image warping (see figure 4) still cause visible colour differences between $\mathcal{T}_{last,t-1}$ and $\mathcal{T}_{first,t}$. Though the remaining discrepancies are not disturbing in the still image, they become apparent when replaying the texture sequences. Therefore, an additional cross dissolve blending is performed in parallel to the spatial warping. Again, the cross dissolve is distributed over a large number of frames to achieve a slow and smooth transition. The number of frames has to be chosen carefully: if too few frames are used for the transition, the resulting transition can become apparent due to sudden changes in shading or specularities. On the other hand, if the number of frames is too large, ghosting artifacts can appear because the cross dissolve adds high frequency details while the face deforms (e.g. specularities on the closed eye, lip line on a opened mouth, etc.).

Therefore, an anisotropic cross dissolve was implemented that allows for multiple blending speeds within the same texture. For example, a fast blending (e.g. 4 frames) is used in regions with high frequency differences (e.g. eyes and mouth) whereas slow blending speed (e.g. 40 frames) is applied in smooth regions with mainly low frequency differences (e.g. skin regions). In case of small misalignments, the fast cross dissolve does not create disturbing effects, actually blending small misalignments with cross dissolve results in a sensation of movement [35]. This small but fast movement is barely noticeable in contrast to a slowly appearing or disappearing ghosting effect caused by an ordinary cross dissolve. We implemented the anisotropic cross dissolve by providing an additional speed-up factor b for each texel. For this purpose, a static binary map $\mathcal{B}$ is used to mark regions of increased blending speed. To ensure a smooth spatial transition between regions of different blending speeds, $\mathcal{B}$ is blurred in order to create intermediate regions where b changes gradually from slow to fast. For our experiments, a single binary map was created manually in texture space.

C. Editing of Facial Videos

The general editing approach from the user's point of view is plotted in figure 5. Firstly, the user selects a sequence of frames in the target video which should be modified (e.g. because the facial expression should be more/less intense, retimed or re-arranged in some other way). Now, the edited facial region (e.g. mouth and nose, see figure 6) has to be selected once for example by creating a mask in the first modified frame. Finally, the user can design a new performance for the marked facial region by concatenating/looping facial expressions taken from one or more suitable source videos where the actor shows the desired facial expression/action. The user editable parameters are basically the blending speed of texture and geometry, the selection of the modified facial region and obviously the arrangement of source sequences.

During the editing step, we assume that the geometric proxy mesh is already tracked in the target as well as in the source sequence(s) using the method described in section IV. Based on user input, the extracted mesh-plus-texture sequences are concatenated seamlessly (using the methods described in section VI) and rendered to the target video using the rigid motion parameters $\mathbf{R}^t, \mathbf{T}^t$ of the target video and the deformation parameter $\triangle\mathbf{V}^t$ and texture of the source sequence(s). In order to embed the synthetic sequence seamlessly in the target video

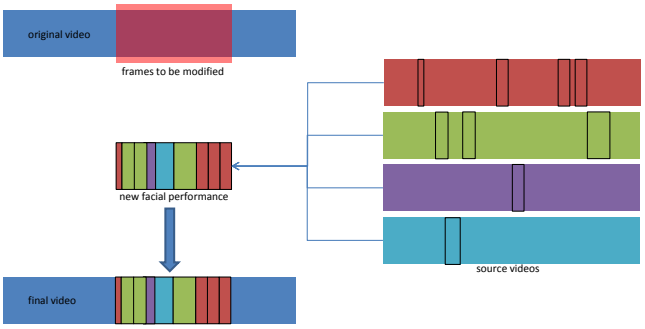

Figure 5. This figure gives a schematic overview of the editing process. Top: original video with the processed frame sequence (red). Middle: desired sequences are taken from different source videos (red, green, purple, cyan) and concatenated/looped to form a new facial performances. Bottom: the new facial performance is inserted in the target video and all originally independet sequences are blended at the transitions (black edges) to form a novel facial performance that faithfully fits in the target video.

we perform a fade-in/out which blends smoothly from the target video to the synthetic sequence and back. This is done by pre/appending a very short sequence (i.e. one frame) to the synthetic sequence. These very short sequences are created directly from the target video and hence exactly represent the facial expression in the target video right before and after the modified part. The fade in/out is then computed between these additional one-frame sequences and the synthetic sequence using the method described in section IV. In order to decide which parts of the target video are modified, we use a binary mask in texture space. The mask is created by the user once for example using a brush-tool marking modified face areas directly in one or multiple target video frames. Based on the tracked geometry the selected areas in the video are used to fill the binary mask in texture space (see figure 6). The advantage of this approach is that the binary mask remains static as the texture space does not change over time although the head is moving and the geometry might change. Finally, the created mask is used to insert the rendered facial performances into the target video (see figure 6 and 7) using Poisson image editing [49].

VII. Results and Discussion

A. Experimental Results

In this section we present result images of the proposed editing technique (figure 7). For our experiments, we captured several facial performances of an actress with 2 calibrated UHD-cameras (Sony-F55) at 59Hz. Additionally, she was captured with a multi-view still camera rig consisting of 14 D-SLR cameras to obtain a reference head geometry in a neutral expression, using the method presented in [37].

We used a semiautomatic mesh unwrapping technique to add texture coordinates to the 3D head model. Then, several short sequences were selected from the recorded footage and the geometry was tracked in each sequence using the deformable tracking method described in section IV. For each tracked video sequence a dynamic texture was created using

Figure 6. This figure shows an example of partial face editing. In the bottom row, the face is overlayed with the binary editing mask (bright parts are modified, all other parts remain the same). On the left side the editing mask is shown in texture space, while on the right side the mask is projected into image space using the tracked 3D model. In the top row, the result of the partial editing is shown: modified version on the left and original version on the right side.

the method presented in section V. To ensure seamless transitions between concatenated or looped facial performances we optimized textures/geometry over the last 30/15 frames of each sequence.

A non-optimized implementation of our system needs approx. 30 seconds per frame for tracking, takes several seconds for one texture mosaic and approximately 10 seconds for the generation of an optimized texture sequence transition. This is sufficient as we consider the main purpose of our presented system as an offline processing tool.

Providing a quality measure of the synthesized videos is complicated: first, the plausibility of a synthesized video depends on the skills of the animator. Although the impact may be smaller than in traditional animation techniques, because the animator needs less technical skills (compared to using rigged models), showing for example contrasting emotions in the same face might still result in an implausible result. Also, there is no accurate ground truth available as we are creating new video sequences using multiple recorded source sequences. Thus, we have to resort to visual evaluation of the presented components, e.g. to check if they introduce artifacts (e.g. due to inaccurate motion caused by the tracking system, obvious transitions caused by the blending methods) in the synthesized video.

The results show that by using the proposed deformable tracking method, a single static 3D reference model is sufficient to accurately track the facial geometry of the captured subject. Large and medium scale deformations are captured in geometry while fine scale motions and small details are

Figure 7. This figure shows some images from video streams that were edited with the proposed technique. Bottom row shows the original images, middle row shows the manipulated version and top row shows a zoomed-in version of the manipulated images.

captured in dynamic textures. The presented warping/blending techniques are used to create seamless transitions between independent facial performances. The geometry blending works well for small and medium deformations. Therefore, a transition search is employed to find an optimal transition point. Remaining motions/differences in texture space are compensated using image warping and anisotropic cross dissolve.

The aforementioned techniques allow transferring facial performances from one video to another even with different head orientation between source and target sequence. Additionally, the presented approach requires only little additional data: one static 3D model of the actor's head and calibrated cameras are sufficient to apply the presented facial performance editing technique. We also aim at keeping manual effort as low as possible: the user is required to select a few vertices in the 3D model as landmarks, their counterparts in the video sequences can be found by a facial feature detector.

More convincing results can be found in the supplemental material: we show several editing results of 3 individuals and a comparison of the proposed blending techniques as well as possible limitations.

B. Limitations

Strong lighting variations may present a limitation of the image based concatenation and rendering methods, as these variations are conserved in the texture (figure 8). We addressed this for example by capturing under homogeneous illumination to capture only textural changes caused by changing facial expressions.

Another possible solution would be performing a global relighting based on the tracked geometry (e.g. as demonstrated in [50]) to adapt the synthetic performance to a target scene. Occlusions of the face are currently not detected since the facial expression can only be recovered in visible areas. This means the user of our system has to choose suitable source sequences when he/she creates the composite sequence. However, even if a face is partly occluded it is still possible to use the visible part (e.g. only eyes, only mouth) as a source for expression transfer. Using a model based approach to hallucinate the geometry and texture seems tempting but we actually believe that a most-probable solution is not necessarily what an editor wants to improve/refine a facial expression with. Target sequences where the face is partially occluded could be handled using an occlusion detection based on geometry and/or texture to detect and preserve occluded areas while regions belonging to the face will be modified. Self occlusion, on the other hand, is minimized as our system supports multiple cameras which also gives more freedom to the actor, as he can move the head freely without the need to look directly into a designated camera.

The speed and duration of a facial performance typically varies from shot to shot. In some cases it may be desirable to modify the speed of a facial expression for example to intensify the displayed emotion. This can for example be achieved by taking a slower/faster version from a different shot but this

Figure 8. This figure shows additional experiments. The left side shows samples from an inter-person expression transfer: target(left), result(middle), source(right). The right side shows results of an expression transfer in case of different light conditions between source and target sequence. We added synthetic light to the tracked video sequence and used this as expression source: target(left), result(middle) source(right).

may also cause a gap in the final video. Simply playing the inserted sequence at lower/higher speed or repeating/leaving out the frames is not an option. A possible solution could be for example to fill the gap e.g. by inserting a short loopable sequence that fits the current facial expression as we did in our experiments. Our system assumes that source and target sequences have the same frame rate which we do not consider a strong limitations, however if someone wants to make use of source videos from different capture sessions with different frame rates it would be necessary to use existing tools for frame rate conversion.

When editing facial performances is crucial to maintaining a silhouette consistent with the displayed facial expression. The tracked 3D face model can for example be used to open the mouth because the opened mouth simply occludes more background pixels producing no artifacts. However, trying to close an open mouth cannot reveal background pixel that were originally occluded by actor during the video shot and creates an implausible result.

The presented approach relies purely on synthetic transitions which is advantageous as it reduces the number of necessary source sequences. However in some cases (e.g. strong deformations) it might still be necessary to use captured transitions in order to realistically change from one facial performance/expression to another. We use a deformable surface tracking method to augment the captured video footage with time consistent 3D information. While these accurately capture deformations of the facial geometry (see figure 2) we use a coarse approximation of teeth and mouth interior by simply closing the gap between upper and lower lip (see figure 2) with a single row of triangles. This can cause artifacts when changing the viewpoint in a sequence with an opened mouth and should be considered when facial performances are edited or composed.

The presented approach mainly aims at intra-person editing tasks. This means, video footage of one actor is used to modifiy facial shots of the same actor. For example to perform tasks like replacing/removing unwanted facial expressions, re-timing facial actions or concatenating multiple sequences to create a new facial performance. Inter-person expression transfer on the other hand is currently not fully supported. It is for example possible to exchange facial expressions between different persons for a complete shot (e.g. replacing the stuntman's face using video footage of the main actor). However, as we are using an image based rendering approach the whole appearance of a source actor is transferred too and not only the expression (figure 8). While this might still be usefull for expression transfer between similar looking people (e.g. with color matching to adjust differences in the skin tone and a non-uniform scaling to adjust the head size/shape) it will in general be difficult to achieve plausible results.

C. Conclusion

This paper presents an inexpensive but effective technique for editing facial video sequences. We use captured video material to perform example based re-animation of human faces in videos which enables even untrained editors to achieve photo realistic results without the need for complex manual intervention. We also do not rely on additional expensive hardware for video capture and 3D model creation (e.g. can be done with off-the-shelf hardware) which makes it attractive for low cost production. The used model free tracking method adds accurate and time consistent 3D information to the recorded video sequences without creating much overhead

in terms of model creation and capturing. A single, static 3D reference model of the actor's head is sufficient and the tracking is performed on the original video footage (i.e. no retro reflective markers are necessary).

Using the 3D data, temporally consistent dynamic textures are extracted from the video sequences which allows incorporating the video data from one or multiple video streams into a single texture stream. This way, a facial performance is completely encapsulated in a geometry-plus-texture stream which combines the photo realism of real image data with the ability to modify viewpoint and re-arrange recorded performances. To edit a video, the user simply transfers a facial performance or parts of it (e.g. eyes, mouth) from one shot to another. Furthermore, by concatenating and looping short sequences, the system also enables the user to synthesize novel and more complex performances.

D. Future Work

Currently, we modify the inner region of the face without correcting the silhouette. With this giving a proof for the effectiveness our approach, it would be an interesting extension to provide a realistic silhouette adaption according to the new facial performance. In order to provide a convenient workflow we use an automatic method that seamlessly concatenates different sequences: a heuristic is used to find a suitable transition point and remaining differences are compensated using texture and geometry blending. To push this idea further, we want to investigate ways of creating more realistic transitions, even between sequences with a large difference at the transition point, based on example motions taken from the captured geometry data. In order to make the system more robust with respect to illumination changes, an illumination estimation will be added to the geometry tracking which will then allow to compensate for these changes in the dynamic textures. Also, evaluating how well this can be achieved using our photometric warping component would be an interesting challenge.

E. Acknowledgment

The work presented in this paper has been partially funded by the Seventh Framework Programme EU projects RE@CT (FP7-ICT-288369) and by the BMBF project 3DGIM (03ZZ0407).

REFERENCES

[1] M. Mori, "Bukimi no tani [The uncanny valley]," *Energy*, vol. 7, no. 4, pp. 33–35, 1970.

[2] M. Mori, K. F. MacDorman, and N. Kageki, "The uncanny valley [from the field]," *IEEE Robot. Automat. Mag.*, vol. 19, no. 2, pp. 98–100, 2012.

[3] J. Seyama and R. S. Nagayama, "The uncanny valley: Effect of realism on the impression of artificial human faces," *Presence: Teleoper. Virtual Environ.*, vol. 16, no. 4, pp. 337–351, Aug. 2007.

[4] W. Paier, M. Kettern, H. Anna, and P. Eisert, "Video-based facial re-animation," in *Proceedings of the 12th European Conference on Visual Media Production*, ser. CVMP '15. ACM, 2015.

[5] M. Kettern, A. Hilsmann, and P. Eisert, "Temporally consistent wide baseline facial performance capture via image warping," in *Proceedings of the Vision, Modeling, and Visualization Workshop 2015*, Aachen, Germany, October 2015.

[6] F. I. Parke, "Computer generated animation of faces," *Proceedings of the ACM annual conference - Volume 1*, pp. 451–457, 1972.

[7] O. Alexander, M. Rogers, W. Lambeth, M. Chiang, and P. Debevec, "The digital emily project: Photoreal facial modeling and animation," in *ACM SIGGRAPH 2009 Courses*, ser. SIGGRAPH '09. New York, NY, USA: ACM, 2009, pp. 12:1–12:15.

[8] P. Garrido, L. Valgaert, C. Wu, and C. Theobalt, "Reconstructing Detailed Dynamic Face Geometry from Monocular Video," *ACM Transactions on Graphics*, vol. 32, no. 6, 2013.

[9] C. Cao, D. Bradley, K. Zhou, and T. Beeler, "Real-time high-fidelity facial performance capture," *ACM Transactions on Graphics*, vol. 34, no. 4, pp. 46:1–46:9, Jul. 2015.

[10] C. Cao, Q. Hou, and K. Zhou, "Displaced dynamic expression regression for real-time facial tracking and animation," *ACM Transactions on Graphics*, vol. 33, no. 4, pp. 43:1–43:10, Jul. 2014.

[11] F. Shi, H.-T. Wu, X. Tong, and J. Chai, "Automatic acquisition of high-fidelity facial performances using monocular videos," *ACM Transactions on Graphics*, vol. 33, no. 6, pp. 222:1–222:13, Nov. 2014.

[12] A. E. Ichim, S. Bouaziz, and M. Pauly, "Dynamic 3d avatar creation from hand-held video input," *ACM Trans. Graph.*, vol. 34, no. 4, pp. 45:1–45:14, Jul. 2015.

[13] T. Weise, H. Li, L. Van Gool, and M. Pauly, "Face/Off," *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation - SCA '09*, p. 7, 2009.

[14] H. Li, J. Yu, Y. Ye, and C. Bregler, "Realtime facial animation with on-the-fly correctives," *ACM Trans. Graph.*, vol. 32, no. 4, pp. 42:1–42:10, Jul. 2013.

[15] L. Zhang, N. Snavely, B. Curless, and S. M. Seitz, "Spacetime Faces:˜{H}igh Resolution Capture for Modeling and Animation," *Proc. SIGGRAPH*, pp. 548–558, 2004.

[16] Y. Furukawa and J. Ponce, "Dense 3d motion capture for human faces," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1674–1681, 2009.

[17] T. Beeler, F. Hahn, D. Bradley, B. Bickel, P. Beardsley, C. Gotsman, R. W. Sumner, and M. Gross, "High-quality passive facial performance capture using anchor frames," *ACM Transactions on Graphics*, vol. 30, no. 4, p. 1, 2011.

[18] E. de Aguiar, C. Stoll, C. Theobalt, N. Ahmed, H.-P. Seidel, and S. Thrun, "Performance capture from sparse multi-view video," *ACM Transactions on Graphics*, vol. 27, no. 3, p. 1, 2008.

[19] A. Hilsmann and P. Eisert, "Tracking deformable surfaces with optical flow in the presence of self-occlusions in monocular image sequences," in *CVPR Workshops, Workshop on Non-Rigid Shape Analysis and Deformable Image Alignment (NORDIA)*. IEEE Computer Society, June 2008, pp. 1–6.

[20] G. Borshukov, J. Montgomery, W. Werner, B. Ruff, J. Lau, P. Thuriot, P. Mooney, S. Van Niekerk, D. Raposo, J.-L. Duprat, J. Hable, H. Kihlström, D. Roizman, K. Noone, and J. O'Connell, "Playable universal capture," in *ACM SIGGRAPH 2006 Sketches*, ser. SIGGRAPH '06. New York, NY, USA: ACM, 2006. [Online]. Available: http://doi.acm.org/10.1145/1179849.1179884

[21] C. Lipski, F. Klose, K. Ruhl, and M. Magnor, "Making of who cares hd stereoscopic free viewpoint video," in *European Conference on Visual Media Production (CVMP)*, 2011.

[22] J. Carranza, C. Theobalt, M. Magnor, and H.-P. Seidel, "Free-viewpoint video of human actors," in *ACM SIGGRAPH*, 2003.

[23] J. Kilner, J. Starck, and A. Hilton, "A comparative study of free-viewpoint video techniques for sports events," in *European Conference on Visual Media Production (CVMP)*, 2006.

[24] A. Schödl and I. A. Essa, "Controlled animation of video sprites," in *SCA '02: Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*. New York, NY, USA: ACM, 2002, pp. 121–127.

[25] D. Casas, M. Volino, J. Collomosse, and A. Hilton, "4d video textures for interactive character appearance," *Computer Graphics Forum*, vol. 33, no. 2, pp. 371–380, 2014.

[26] W. Paier, M. Kettern, and P. Eisert, "Realistic retargeting of facial video," in *Proceedings of the 11th European Conference on Visual Media Production*, ser. CVMP '14. New York, NY, USA: ACM, 2014, pp. 2:1–2:10.

[27] K. Dale, K. Sunkavalli, M. K. Johnson, D. Vlasic, W. Matusik, and H. Pfister, "Video face replacement," *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)*, vol. 30, 2011.

[28] J. Thies, M. Zollhöfer, M. Nießner, L. Valgaerts, M. Stamminger, and C. Theobalt, "Real-time expression transfer for facial reenactment," *ACM Transactions on Graphics (TOG)*, vol. 34, no. 6, 2015.

[29] F. Xu, J. Chai, Y. Liu, and X. Tong, "Controllable high-fidelity facial performance transfer," *ACM Trans. Graph.*, vol. 33, no. 4, pp. 42:1–42:11, Jul. 2014. [Online]. Available: http://doi.acm.org/10.1145/2601097.2601210
[30] F. Xu, Y. Liu, C. Stoll, J. Tompkin, G. Bharaj, Q. Dai, H.-P. Seidel, J. Kautz, and C. Theobalt, "Video-based characters: Creating new human performances from a multi-view video database," in *ACM SIGGRAPH 2011 Papers*, ser. SIGGRAPH '11. New York, NY, USA: ACM, 2011, pp. 32:1–32:10.
[31] K. Li, F. Xu, J. Wang, Q. Dai, and Y. Liu, "A data-driven approach for facial expression synthesis in video," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 57–64.
[32] L. Kovar, M. Gleicher, and F. Pighin, "Motion graphs," in *Proc. of the 29th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '02. New York, NY, USA: ACM, 2002, pp. 473–482.
[33] P. Huang, A. Hilton, and J. Starck, "Human motion synthesis from 3d video," in *IEEE Conf. on Computer Vision and Pattern Recognition*, June 2009, pp. 1478–1485.
[34] D. Casas, M. Tejera, J.-Y. Guillemaut, and A. Hilton, "4d parametric motion graphs for interactive animation," in *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, ser. I3D '12. New York, NY, USA: ACM, 2012, pp. 103–110.
[35] I. Kemelmacher-Shlizerman, E. Shechtman, R. Garg, and S. M. Seitz, "Exploring photobios," *ACM Trans. Graph.*, vol. 30, no. 4, pp. 61:1–61:10, Jul. 2011.
[36] F. Pighin, J. Hecker, D. Lischinski, R. Szeliski, and D. H. Salesin, "Synthesizing realistic facial expressions from photographs," in *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '98. New York, NY, USA: ACM, 1998, pp. 75–84. [Online]. Available: http://doi.acm.org/10.1145/280814.280825
[37] D. Blumenthal-Barby and P. Eisert, "High-resolution depth for binocular image-based modelling," *Computers & Graphics*, vol. 39, pp. 89–100, 2014.
[38] J. Saraigh, S. Lucey, and J. Cohn, "Deformable Model Fitting by Regularized Landmark Mean-Shift," *INTERNATIONAL JOURNAL OF COMPUTER VISION*, vol. 91, no. 2, 2011.
[39] M. Kettern, D. Blumenthal-Barby, and P. Eisert, "High detail flexible viewpoint facial video from monocular input using static geometric proxies," in *Proceedings of the 6th International Conference on Computer Vision / Computer Graphics Collaboration Techniques and Applications*, ser. MIRAGE '13. New York, NY, USA: ACM, 2013, pp. 2:1–2:8.
[40] A. Hilsmann and P. Eisert, "Joint estimation of deformable motion and photometric parameters in single view video," in *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*. IEEE, 2009, pp. 390–397.
[41] O. Sorkine, D. Cohen-Or, Y. Lipman, M. Alexa, C. Rössl, and H.-P. Seidel, "Laplacian Surface Editing," in *Eurographics Symposium on Geometry Processing*, ser. SGP '04. ACM, 2004, pp. 175–184.
[42] P. J. Green, "Iteratively reweighted least squares for maximum likelihood estimation, and some robust and resistant alternatives," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 149–192, 1984.
[43] I. Daubechies, R. DeVore, M. Fornasier, and C. S. Güntürk, "Iteratively reweighted least squares minimization for sparse recovery," *Communications on Pure and Applied Mathematics*, vol. 63, no. 1, pp. 1–38, 2010.
[44] B. K. Horn and B. G. Schunck, "Determining optical flow," Cambridge, MA, USA, Tech. Rep., 1980.
[45] V. Kwatra, A. Schödl, I. Essa, G. Turk, and A. Bobick, "Graphcut Textures: Image and Video Synthesis Using Graph Cuts," in *ACM SIGGRAPH 2003 Papers*, ser. SIGGRAPH '03. New York, NY, USA: ACM, 2003, pp. 277–286.
[46] V. Lempitsky and D. Ivanov, "Seamless Mosaicing of Image-Based Texture Maps," in *CVPR*. IEEE Computer Society, 2007.
[47] Y. Boykov and V. Kolmogorov, "An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 26, no. 9, pp. 1124–1137, 2004.
[48] E. Reinhard, M. Ashikhmin, B. Gooch, and B. Shirley, "Color transfer between images," *IEEE Computer Graphics Applications*, vol. 21, no. 5, pp. 34–41, 2001.
[49] P. Pérez, M. Gangnet, and A. Blake, "Poisson image editing," *ACM Trans. Graph.*, vol. 22, no. 3, pp. 313–318, Jul. 2003.
[50] D. Bradley, W. Heidrich, T. Popa, and A. Sheffer, "High resolution passive facial performance capture," *ACM Trans. on Graphics (Proc. SIGGRAPH)*, vol. 29, no. 3, 2010.

Wolfgang Paier received his B.S. degree in software development and economics from the Technical University of Graz, Austria in 2009 and the M.Sc. degree in computer science from the Free University of Berlin, Germany in 2013. He joined Fraunhofer HHI in 2013 where he has been working on 3D reconstruction, texture synthesis from multiview video and texture based face re-animation. His current research interests focus on texture based facial performance capture and editing techniques.

Markus Kettern recieved his M.A. degree in computer science and musicology in 2009 and has been working as a computer vision and graphics researcher ever since. He started his career at Fraunhofer HHI where he worked on deformable surface tracking, model-based methods and 3D reconstruction as well as camera calibration and digital matting. Today, he is working as a freelance researcher and developer on these topics and beyond.

Anna Hilsmann received her Dipl.-Ing. degree in Electrical Engineering and Information Technology from RWTH Aachen in 2006 and her Dr.-Ing. degree in Computer Science from HU Berlin in 2014. She joined the Computer Vision and Graphics Group at Fraunhofer HHI in 2007 and the Visual Computing Group at HU Berlin in 2011. Since 2015, she is heading the Computer Vision & Graphics Group at Fraunhofer HHI. Her main research interests cover 3D image and video analysis, such as image registration, model-based deformable tracking and 3D reconstruction, as well as synthesis, auch as image- and video-based rendering, animation and editing.

Peter Eisert is Professor for Visual Computing at the Humboldt University Berlin and heading the Vision & Imaging Technologies Department of the Fraunhofer HHI Berlin, Germany. He received the Dipl.-Ing. degree in Electrical Engineering "with highest honors" from the Technical University of Karlsruhe, Germany, in 1995 and the Dr.-Ing. degree "with highest honors" from the University of Erlangen-Nuremberg, Germany, in 2000. In 2001, he worked as a postdoctoral fellow at the Stanford University, USA, on 3D image analysis and synthesis as well as facial animation and computer graphics. In 2002, he joined FhG-HHI, where he is coordinating and initiating numerous national and international 3rd party funded research projects. He has published more than 150 conference and journal papers on the subject of 3D reconstruction, facial expression analysis and synthesis and image-based rendering. He is Associate Editor of the International Journal of Image and Video Processing and in the Editorial Board of the Journal of Visual Communication and Image Representation. His research interests include 3D image analysis and synthesis, face processing, image-based rendering, computer vision, computer graphics, as well as image and video processing.