Face2Face: Real-time Face Capture and Reenactment of RGB Videos

Justus Thies
Technical University Munich
Boltzmannstr. 3
Garching, Germany
justus.thies@tum.de

Michael Zollhöfer
Stanford University
353 Serra Mall
Stanford, CA, USA
zollhoefer@cs.stanford.edu

Marc Stamminger
University of
Erlangen-Nuremberg
Cauerstr. 11
Erlangen, Germany
marc.stamminger@fau.de

Christian Theobalt
Max-Planck-Institute for
Informatics
Campus E1.4
Saarbrücken, Germany
theobalt@mpi-inf.mpg.de

Matthias Nießner
Technical University Munich
Boltzmannstr. 3
Garching, Germany
niessner@tum.de

ABSTRACT

Face2Face is an approach for real-time facial reenactment of a monocular target video sequence (e.g., Youtube video). The source sequence is also a monocular video stream, captured live with a commodity webcam. Our goal is to animate the facial expressions of the target video by a source actor and re-render the manipulated output video in a photo-realistic fashion. To this end, we first address the under-constrained problem of facial identity recovery from monocular video by non-rigid model-based bundling. At run time, we track facial expressions of both source and target video using a dense photometric consistency measure. Reenactment is then achieved by fast and efficient deformation transfer between source and target. The mouth interior that best matches the re-targeted expression is retrieved from the target sequence and warped to produce an accurate fit. Finally, we convincingly re-render the synthesized target face on top of the corresponding video stream such that it seamlessly blends with the real-world illumination. We demonstrate our method in a live setup, where Youtube videos are reenacted in real time. This live setup has also been shown at SIGGRAPH Emerging Technologies 2016 [20], where it won the Best in Show Award.

1. INTRODUCTION

In recent years, real-time markerless facial performance capture based on commodity sensors has been demonstrated. Impressive results have been achieved, both based on RGB as well as RGB-D data. These techniques have become increasingly popular for the animation of virtual CG avatars in video games and movies. It is now feasible to run these face capture and tracking algorithms from home, which is the foundation for many VR and AR applications, such as teleconferencing.

In this paper, we employ a new dense markerless facial performance capture method based on monocular RGB data, similar to state-of-the-art methods. However, instead of transferring facial expressions to virtual CG characters, our main contribution is monocular *facial reenactment* in real-time. In contrast to previous reenactment approaches that run offline, our goal is the *online* transfer of facial expressions of a source actor captured by an RGB sensor to a target actor. The target sequence can be any monocular video; e.g., legacy video footage downloaded from Youtube with a facial performance. We aim to modify the target video in a photo-realistic fashion, such that it is virtually impossible to notice the manipulations. Faithful photo-realistic facial reenactment is the foundation for a variety of applications; for instance, in video conferencing, the video feed can be adapted to match the face motion of a translator, or face videos can be convincingly dubbed to a foreign language.

In our method, we first reconstruct the shape identity of the target actor using a new global non-rigid model-based bundling approach based on a prerecorded training sequence. As this preprocess is performed globally on a set of training frames, we can resolve geometric ambiguities common to monocular reconstruction. At runtime, we track both the expressions of the source and target actor's video by a dense analysis-by-synthesis approach based on a statistical facial prior. We demonstrate that our RGB tracking accuracy is on par with the state of the art, even with online tracking methods relying on depth data. In order to transfer expressions from the source to the target actor in real-time, we propose a novel transfer functions that efficiently applies deformation transfer [18] directly in the used low-dimensional expression space. For final image synthesis, we re-render the target's face with transferred expression coefficients and composite it with the target video's background under consideration of the estimated environment lighting. Finally, we introduce a new image-based mouth synthesis approach that generates a realistic mouth interior by retrieving and warping best matching mouth shapes from the offline sample sequence. It is important to note that we maintain the appearance of the target mouth shape; in contrast, existing methods either copy the source mouth region onto the target [23]or a generic teeth proxy is rendered [8, 19], both of which leads to inconsistent results. Fig. 2 shows an overview of our method.

The original version of this paper is entitled "Face2Face: Real-time Face Capture and Reenactment of RGB Videos" and was published in Proc. Computer Vision and Pattern Recognition (CVPR), 2016, IEEE.

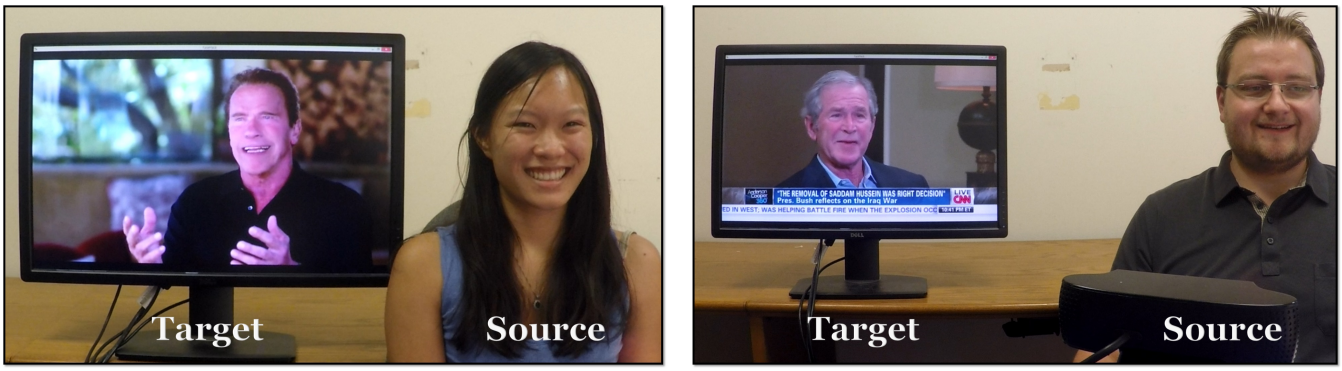

Figure 1: Proposed online reenactment setup: A monocular target video sequence (e.g., from Youtube) is reenacted based on the expressions of a source actor who is recorded live with a commodity webcam.

We demonstrate highly-convincing transfer of facial expressions from a source to a target video in real time. We show results with a live setup where a source video stream, which is captured by a webcam, is used to manipulate a target Youtube video. In addition, we compare against state-of-the-art reenactment methods, which we outperform both in terms of resulting video quality and runtime (we are the first real-time RGB reenactment method). In summary, our key contributions are:

- dense, global non-rigid model-based bundling,
- accurate tracking, appearance, and lighting estimation in unconstrained live RGB video,
- person-dependent expression transfer using subspace deformations,
- and a novel mouth synthesis approach.

2. RELATED WORK

Offline RGB Performance Capture.

Recent offline performance capture techniques approach the hard monocular reconstruction problem by fitting a blendshape or a multi-linear face model to the input video sequence. Even geometric fine-scale surface detail is extracted via inverse shading-based surface refinement. Shi et al. [16] achieve impressive results based on global energy optimization of a set of selected keyframes. Our model-based bundling formulation to recover actor identities is similar to their approach; however, we use robust and dense global photometric alignment, which we enforce with an efficient data-parallel optimization strategy on the GPU.

Online RGB-D Performance Capture.

Weise et al. [25] capture facial performances in real-time by fitting a parametric blendshape model to RGB-D data, but they require a professional, custom capture setup. The first real-time facial performance capture system based on a commodity depth sensor has been demonstrated by Weise et al. [24]. Follow up work focused on corrective shapes [2], dynamically adapting the blendshape basis [11], non-rigid mesh deformation [6]. These works achieve impressive results, but rely on depth data which is typically unavailable in most video footage.

Online RGB Performance Capture.

While many sparse real-time face trackers exist, e.g., [15], real-time dense monocular tracking is the basis of realistic online facial reenactment. Cao et al. [5] propose a real-time regression-based approach to infer 3D positions of facial landmarks which constrain a user-specific blendshape model. Follow-up work [4] also regresses fine-scale face wrinkles. These methods achieve impressive results, but are not directly applicable as a component in facial reenactment, since they do not facilitate dense, pixel-accurate tracking.

Offline Reenactment.

Vlasic et al. [23] perform facial reenactment by tracking a face template, which is re-rendered under different expression parameters on top of the target; the mouth interior is directly copied from the source video. Image-based offline mouth re-animation was shown in [3]. Garrido et al. [7] propose an automatic purely image-based approach to replace the entire face. These approaches merely enable self-reenactment; i.e., when source and target are the same person; in contrast, we perform reenactment of a different target actor. Recent work presents virtual dubbing [8], a problem similar to ours; however, the method runs at slow offline rates and relies on a generic teeth proxy for the mouth interior. Li et al. [12] retrieve frames from a database based on a similarity metric. They use optical flow as appearance and velocity measure and search for the k-nearest neighbors based on time stamps and flow distance. Saragih et al. [15] present a real-time avatar animation system from a single image. Their approach is based on sparse landmark tracking, and the mouth of the source is copied to the target using texture warping.

Online Reenactment.

Recently, first online facial reenactment approaches based on RGB-(D) data have been proposed. Kemelmacher-Shlizerman et al. [10] enable image-based puppetry by querying similar images from a database. They employ an appearance cost metric and consider rotation angular distance. While they achieve impressive results, the retrieved stream of faces is not temporally coherent. Thies et al. [19] show the first online reenactment system; however, they rely on depth data and use a generic teeth proxy for the mouth region. In this paper, we address both shortcomings: 1) our method is the first real-time RGB-only reenactment technique; 2) we synthesize the mouth regions exclusively from the target sequence (no need for a teeth proxy or direct source-to-target copy).

Follow-up Work.

The core component of the proposed approach is the dense face reconstruction algorithm. It has already been adapted for several applications, such as head mounted display removal [22], facial projection mapping [17], and avatar digitization [9]. FaceVR [22] demonstrates self-reenactment for head mounted display removal, which is particularly useful for enabling natural teleconferences in

virtual reality. The FaceForge [17] system enables real-time facial projection mapping to dynamically alter the appearance of a person in the real world. The avatar digitization approach of Hu et al. [9] reconstructs a stylized 3D avatar that includes hair and teeth, from just a single image. The resulting 3D avatars can for example be used in computer games.

3. USE CASES

The proposed facial tracking and reenactment has several use-cases that we want to highlight in this section. In movie productions the idea of facial reenactment can be used as a video editing tool to change for example the expression of an actor in a particular shot. Using the estimated geometry of an actor, it can also be used to modify the appearance of a face in a post-process, e.g., changing the illumination. Another field in post-production is the synchronization of an audio channel to the video. If a movie is translated to another language, the movements of the mouth do not match the audio of the so called dubber. Nowadays, to match the video, the audio including the spoken text is adapted, which might result in a loss of information. Using facial reenactment instead, the expressions of the dubber can be transferred to the actor in the movie and thus the audio and video is synchronized. Since our reenactment approach runs in real time, it is also possible to setup a teleconferencing system with a live interpreter that simultaneously translates the speech of a person to another language.

In contrast to state-of-the-art movie production setups that work with markers and complex camera setups, our system presented in this paper only requires commodity hardware without the need for markers. Our tracking results can also be used to animate virtual characters. These virtual characters can be part of animation movies, but can also be used in computer games. With the introduction of virtual reality glasses, also called head mounted displays (HMDs), the realistic animation of such virtual avatars, becomes more and more important for an immersive game-play. FaceVR [22] demonstrates that facial tracking is also possible if the face is almost completely occluded by such an HMD. The project also paves the way to new applications like teleconferencing in VR based on HMD removal.

Besides these consumer applications, you can also think of numerous medical applications. For example one can build a training system that helps patients to train expressions after a stroke.

4. METHOD OVERVIEW

In the following, we describe our real-time facial reenactment pipeline (see Fig. 2). Input to our method is a monocular target video sequence and a live video stream captured by a commodity webcam. First, we describe how we synthesize facial imagery using a statistical prior and an image formation model (see Sec. 5). We find optimal parameters that best explain the input observations by solving a variational energy minimization problem (see Sec. 6). We minimize this energy with a tailored, data-parallel GPU-based *Iteratively Reweighted Least Squares* (IRLS) solver (see Sec. 7). We employ IRLS for off-line non-rigid model-based bundling (see Sec. 8) on a set of selected keyframes to obtain the facial identity of the source as well as of the target actor. This step jointly recovers the facial identity, expression, skin reflectance, and illumination from monocular input data. At runtime, both source and target animations are reconstructed based on a model-to-frame tracking strategy with a similar energy formulation. For reenactment, we propose a fast and efficient deformation transfer approach that directly operates in the subspace spanned by the used statistical prior (see Sec. 9). The mouth interior that best matches the re-targeted

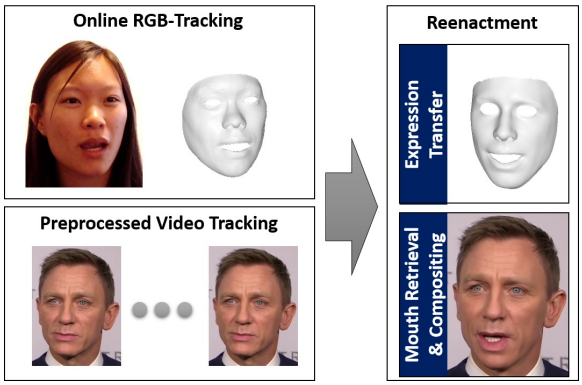

Figure 2: An overview of our reenactment approach: In a preprocessing step we analyze and reconstruct the face of the target actor. During live reenactment, we track the expression of the source actor and transfer them to the reconstructed target face. Finally, we composite a novel image of the target person using a mouth interior of the target sequence that best matches the new expression.

expression is retrieved from the input target sequence (see Sec. 10) and is warped to produce an accurate fit. We demonstrate our complete pipeline in a live reenactment setup that enables the modification of arbitrary video footage and perform a comparison to state-of-the-art tracking as well as reenactment approaches (see Sec. 11). In Sec. 12 we show the limitations of our proposed method.

Since we are aware of the implications of a video editing tool like Face2Face, we included a section in this paper that discusses the potential misuse of the presented technology (see Sec. 13). Finally, we conclude with an outlook on future work (see Sec. 14).

5. SYNTHESIS OF FACIAL IMAGERY

The synthesis of facial imagery is based on a multi-linear face model (see the original Face2Face paper for more details). The first two dimensions represent facial identity – i.e., geometric shape and skin reflectance – and the third dimension controls the facial expression. Hence, we parametrize a face as:

$$\mathcal{M}_{\text{geo}}(\boldsymbol{\alpha}, \boldsymbol{\delta}) = \boldsymbol{a}_{\text{id}} + E_{\text{id}} \cdot \boldsymbol{\alpha} + E_{\text{exp}} \cdot \boldsymbol{\delta} \,, \quad (1)$$

$$\mathcal{M}_{\text{alb}}(\boldsymbol{\beta}) = \boldsymbol{a}_{\text{alb}} + E_{\text{alb}} \cdot \boldsymbol{\beta} \,. \quad (2)$$

This prior assumes a multivariate normal probability distribution of shape and reflectance around the average shape $\boldsymbol{a}_{\text{id}} \in \mathbb{R}^{3n}$ and reflectance $\boldsymbol{a}_{\text{alb}} \in \mathbb{R}^{3n}$. The shape $E_{\text{id}} \in \mathbb{R}^{3n \times 80}$, reflectance $E_{\text{alb}} \in \mathbb{R}^{3n \times 80}$, and expression $E_{\text{exp}} \in \mathbb{R}^{3n \times 76}$ basis and the corresponding standard deviations $\sigma_{\text{id}} \in \mathbb{R}^{80}$, $\sigma_{\text{alb}} \in \mathbb{R}^{80}$, and $\sigma_{\text{exp}} \in \mathbb{R}^{76}$ are given. The model has 53K vertices and 106K faces. A synthesized image $C_{\mathcal{S}}$ is generated through rasterization of the model under a rigid model transformation $\Phi(\boldsymbol{v})$ and the full perspective transformation $\Pi(\boldsymbol{v})$. Illumination is approximated by the first three bands of Spherical Harmonics (SH) [13] basis functions, assuming Labertian surfaces and smooth distant illumination, neglecting self-shadowing.

Synthesis is dependent on the face model parameters $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, $\boldsymbol{\delta}$, the illumination parameters $\boldsymbol{\gamma}$, the rigid transformation $\mathbf{R}$, $\mathbf{t}$, and the camera parameters $\boldsymbol{\kappa}$ defining Π. The vector of unknowns $\boldsymbol{\mathcal{P}}$ is the union of these parameters.

6. ENERGY FORMULATION

Given a monocular input sequence, we reconstruct all unknown parameters $\mathcal{P}$ jointly with a robust variational optimization. The proposed objective is highly non-linear in the unknowns and has the following components:

$$E(\boldsymbol{\mathcal{P}}) = \underbrace{w_{col}E_{col}(\boldsymbol{\mathcal{P}}) + w_{lan}E_{lan}(\boldsymbol{\mathcal{P}})}_{data} + \underbrace{w_{reg}E_{reg}(\boldsymbol{\mathcal{P}})}_{prior} \quad . \quad (3)$$

The data term measures the similarity between the synthesized imagery and the input data in terms of photo-consistency E_{col} and facial feature alignment E_{lan}. The likelihood of a given parameter vector $\boldsymbol{\mathcal{P}}$ is taken into account by the statistical regularizer E_{reg}. The weights w_{col}, w_{lan}, and w_{reg} balance the three different sub-objectives. In all of our experiments, we set $w_{col} = 1$, $w_{lan} = 10$, and $w_{reg} = 2.5 \cdot 10^{-5}$. In the following, we introduce the different sub-objectives.

Photo-Consistency. In order to quantify how well the input data is explained by a synthesized image, we measure the photometric alignment error on pixel level:

$$E_{\text{col}}(\boldsymbol{\mathcal{P}}) = \frac{1}{|\mathcal{V}|} \sum_{\boldsymbol{p} \in \mathcal{V}} \| C_{\mathcal{S}}(\boldsymbol{p}) - C_{\mathcal{I}}(\boldsymbol{p}) \|_2 \ , \quad (4)$$

where $C_{\mathcal{S}}$ is the synthesized image, $C_{\mathcal{I}}$ is the input RGB image, and $\boldsymbol{p} \in \mathcal{V}$ denote all visible pixel positions in $C_{\mathcal{S}}$. We use the $\ell_{2,1}$-norm instead of a least-squares formulation to be robust against outliers. In our scenario, distance in color space is based on ℓ_2, while in the summation over all pixels an ℓ_1-norm is used to enforce sparsity.

Feature Alignment. In addition, we enforce feature similarity between a set of salient facial feature point pairs detected in the RGB stream:

$$E_{\text{lan}}(\boldsymbol{\mathcal{P}}) = \frac{1}{|\mathcal{F}|} \sum_{\boldsymbol{f}_j \in \mathcal{F}} w_{\text{conf},j} \left\| \boldsymbol{f}_j - \Pi(\Phi(\boldsymbol{v}_j) \right\|_2^2 \ . \quad (5)$$

To this end, we employ a state-of-the-art facial landmark tracking algorithm by [14]. Each feature point $\boldsymbol{f}_j \in \mathcal{F} \subset \mathbb{R}^2$ comes with a detection confidence $w_{\text{conf},j}$ and corresponds to a unique vertex $\boldsymbol{v}_j = \mathcal{M}_{geo}(\boldsymbol{\alpha}, \boldsymbol{\delta}) \in \mathbb{R}^3$ of our face prior. This helps avoiding local minima in the highly-complex energy landscape of $E_{\text{col}}(\boldsymbol{\mathcal{P}})$.

Statistical Regularization. We enforce plausibility of the synthesized faces based on the assumption of a normal distributed population. To this end, we enforce the parameters to stay statistically close to the mean:

$$E_{\text{reg}}(\boldsymbol{\mathcal{P}}) = \sum_{i=1}^{80} \left[\left(\frac{\boldsymbol{\alpha}_i}{\sigma_{\text{id},i}} \right)^2 + \left(\frac{\boldsymbol{\beta}_i}{\sigma_{\text{alb},i}} \right)^2 \right] + \sum_{i=1}^{76} \left(\frac{\boldsymbol{\delta}_i}{\sigma_{\text{exp},i}} \right)^2 \ . \quad (6)$$

This commonly-used regularization strategy prevents degenerations of the facial geometry and reflectance, and guides the optimization strategy out of local minima [1].

7. DATA-PARALLEL OPTIMIZATION

The proposed robust tracking objective is a general unconstrained non-linear optimization problem. We use *Iteratively Reweighted Least Squares* (IRLS) to minimize this objective in real-time using a novel data-parallel GPU-based solver. The key idea of IRLS is to transform the problem, in each iteration, to a non-linear least-squares problem by splitting the norm in two components:

$$||r(\boldsymbol{\mathcal{P}})||_2 = \underbrace{(||r(\boldsymbol{\mathcal{P}}_{old})||_2)^{-1}}_{constant} \cdot \ ||r(\boldsymbol{\mathcal{P}})||_2^2 \, .$$

Here, $r(\cdot)$ is a general residual and $\boldsymbol{\mathcal{P}}_{old}$ is the solution computed in the last iteration. Thus, the first part is kept constant during one iteration and updated afterwards. Close in spirit to [19], each single iteration step is implemented using the Gauss-Newton approach. We take a single GN step in every IRLS iteration and solve the corresponding system of normal equations $\mathbf{J}^T\mathbf{J}\boldsymbol{\delta}^* = -\mathbf{J}^T\mathbf{F}$ based on PCG to obtain an optimal linear parameter update $\boldsymbol{\delta}^*$. The Jacobian $\mathbf{J}$ and the systems' right hand side $-\mathbf{J}^T\mathbf{F}$ are precomputed and stored in device memory for later processing as proposed by Thies et al. [19]. For more details we refer to the original paper [21]. Note that our complete framework is implemented using DirectX for rendering and DirectCompute for optimization. The joint graphics and compute capability of DirectX11 enables us to execute the analysis-by-synthesis loop without any resource mapping overhead between these two stages. In the case of an analysis-by-synthesis approach, this is essential for runtime performance, since many rendering-to-compute switches are required. To compute the Jacobian $\mathbf{J}$ we developed a differential renderer that is based on the standard rasterizer of the graphics pipeline. To this end, during the synthesis stage, we additionally store the vertex and triangle attributes that are required for computing the partial derivatives to dedicated rendertargets. Using this information a compute shader calculates the final derivatives that are needed for the optimization.

8. NON-RIGID MODEL-BASED BUNDLING

To estimate the identity of the actors in the heavily underconstrained scenario of monocular reconstruction, we introduce a non-rigid model-based bundling approach. Based on the proposed objective, we jointly estimate all parameters over k key-frames of the input video sequence. The estimated unknowns are the global identity $\{\boldsymbol{\alpha}, \boldsymbol{\beta}\}$ and intrinsics $\boldsymbol{\kappa}$ as well as the unknown per-frame pose $\{\boldsymbol{\delta}^k, \mathbf{R}^k, \mathbf{t}^k\}_k$ and illumination parameters $\{\boldsymbol{\gamma}^k\}_k$. We use a similar data-parallel optimization strategy as proposed for model-to-frame tracking, but jointly solve the normal equations for the entire keyframe set. For our non-rigid model-based bundling problem, the non-zero structure of the corresponding Jacobian is block dense. Our PCG solver exploits the non-zero structure for increased performance (see original paper). Since all keyframes observe the same face identity under potentially varying illumination, expression, and viewing angle, we can robustly separate identity from all other problem dimensions. Note that we also solve for the intrinsic camera parameters of Π, thus being able to process uncalibrated video footage. The employed Gauss-Newton framework is embedded in a hierarchical solution strategy (see Fig. 3). The underlying hierarchy enables faster convergence and avoids getting stuck in local minima of the optimized energy function. We start optimizing on a coarse level and lift the solution to the next finer level using the parametric face model. In our experiments we used three levels with 25, 5, and 1 Gauss-Newton iterations for the coarsest, the medium, and the finest level, respectively. In each Gauss-Newton iteration, we employ 4 PCG steps to efficiently solve the underlying normal equations. Our implementation is not restricted to the number k of used keyframes, but the processing time increases linearly with k. In our experiments we used $k = 6$ keyframes for the estimation of the identity parameters, which results in a processing time of only a few seconds ($\sim$ 20s).

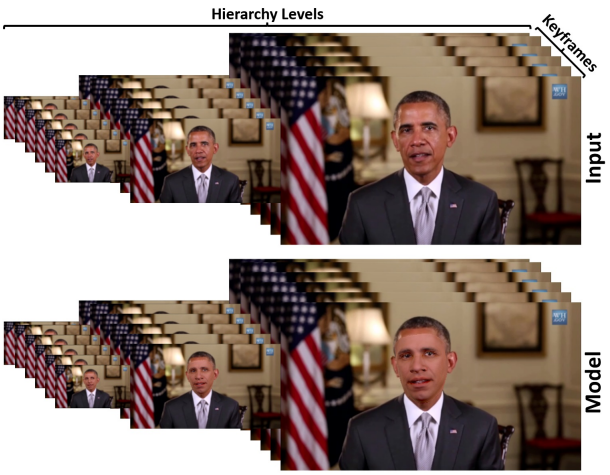

Figure 3: Non-rigid model-based bundling hierarchy: The top row shows the hierarchy of the input video and the second row the overlaid face model.

9. EXPRESSION TRANSFER

To transfer the expression changes from the source to the target actor while preserving person-specificness in each actor's expressions, we propose a sub-space deformation transfer technique. We are inspired by the deformation transfer energy of Sumner et al. [18], but operate directly in the space spanned by the expression blendshapes. This not only allows for the precomputation of the pseudo-inverse of the system matrix, but also drastically reduces the dimensionality of the optimization problem allowing for fast real-time transfer rates. Assuming source identity $\boldsymbol{\alpha}^S$ and target identity $\boldsymbol{\alpha}^T$ fixed, transfer takes as input the neutral $\boldsymbol{\delta}_N^S$, deformed source $\boldsymbol{\delta}^S$, and the neutral target $\boldsymbol{\delta}_N^T$ expression. Output is the transferred facial expression $\boldsymbol{\delta}^T$ directly in the reduced sub-space of the parametric prior.

As proposed by [18], we first compute the source deformation gradients $\mathbf{A}_i \in \mathbb{R}^{3\times3}$ that transform the source triangles from neutral to deformed. The deformed target $\hat{\boldsymbol{v}}_i = \boldsymbol{M}_i(\boldsymbol{\alpha}^T, \boldsymbol{\delta}^T)$ is then found based on the undeformed state $\boldsymbol{v}_i = \boldsymbol{M}_i(\boldsymbol{\alpha}^T, \boldsymbol{\delta}_N^T)$ by solving a linear least-squares problem. Let (i_0, i_1, i_2) be the vertex indices of the i-th triangle, $\mathbf{V} = [\boldsymbol{v}_{i_1} - \boldsymbol{v}_{i_0}, \boldsymbol{v}_{i_2} - \boldsymbol{v}_{i_0}]$ and $\hat{\mathbf{V}} = [\hat{\boldsymbol{v}}_{i_1} - \hat{\boldsymbol{v}}_{i_0}, \hat{\boldsymbol{v}}_{i_2} - \hat{\boldsymbol{v}}_{i_0}]$, then the optimal unknown target deformation $\boldsymbol{\delta}^T$ is the minimizer of:

$$E(\boldsymbol{\delta}^T) = \sum_{i=1}^{|\boldsymbol{F}|} \left|\left| \mathbf{A}_i\mathbf{V} - \hat{\mathbf{V}} \right|\right|_F^2 . \tag{7}$$

This problem can be rewritten in the canonical least-squares form by substitution:

$$E(\boldsymbol{\delta}^T) = \left|\left| \mathbf{A}\boldsymbol{\delta}^T - \boldsymbol{b} \right|\right|_2^2 . \tag{8}$$

The matrix $\mathbf{A} \in \mathbb{R}^{6|\boldsymbol{F}|\times76}$ is constant and contains the edge information of the template mesh projected to the expression subspace. Edge information of the target in neutral expression is included in the right-hand side $\boldsymbol{b} \in \mathbb{R}^{6|\boldsymbol{F}|}$. $\boldsymbol{b}$ varies with $\boldsymbol{\delta}^S$ and is computed on the GPU for each new input frame. The minimizer of the quadratic energy can be computed by solving the corresponding normal equations. Since the system matrix is constant, we can precompute its *Pseudo Inverse* using a Singular Value Decomposition (SVD). Later, the small 76×76 linear system is solved in real-time. No additional smoothness term as in [2, 18] is needed, since the blendshape model implicitly restricts the result to plausible shapes and guarantees smoothness.

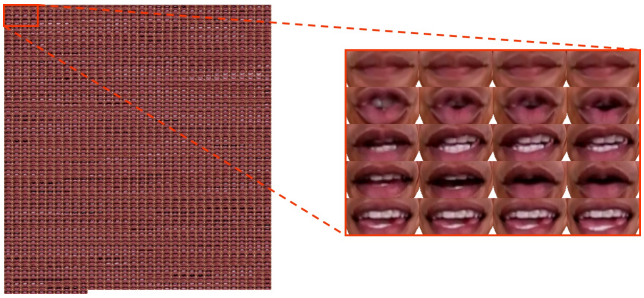

Figure 4: Mouth Database: We use the appearance of the mouth of a person that has been captured in the target video sequence.

10. MOUTH RETRIEVAL

For a given transferred facial expression, we need to synthesize a realistic target mouth region. To this end, we retrieve and warp the best matching mouth image from the target actor sequence. We assume that sufficient mouth variation is available in the target video, i.e., we assume that the entire target video is known or at least a short part of it. It is also important to note that we maintain the appearance of the target mouth. This leads to much more realistic results than either copying the source mouth region [23] or using a generic 3D teeth proxy [8, 19]. For detailed information on the mouth retrieval process, we refer to the original paper.

11. RESULTS

Live Reenactment Setup.

Our live reenactment setup consists of standard consumer-level hardware. We capture a live video with a commodity webcam (source), and download monocular video clips from Youtube (target). In our experiments, we use a *Logitech HD Pro C920* camera running at 30Hz in a resolution of 640×480; although our approach is applicable to any consumer RGB camera. Overall, we show highly-realistic reenactment examples of our algorithm on a variety of target Youtube videos at a resolution of 1280×720. The videos show different subjects in different scenes filmed from varying camera angles; each video is reenacted by several volunteers as source actors. Reenactment results are generated at a resolution of 1280×720. We show real-time reenactment results in Fig. 5 and in the accompanying video.

Runtime.

For all experiments, we use three hierarchy levels for tracking (source and target). In pose optimization, we only consider the second and third level, where we run one and seven Gauss-Newton steps, respectively. Within a Gauss-Newton step, we always run four PCG steps. In addition to tracking, our reenactment pipeline has additional stages whose timings are listed in Table 1. Our method runs in real-time on a commodity desktop computer with an NVIDIA Titan X and an Intel Core i7-4770.

Tracking Comparison to Previous Work.

Face tracking alone is not the main focus of our work, but the following comparisons show that our tracking is on par with or exceeds the state of the art. Here we show some of the comparisons that we conducted in the original paper.

Figure 5: Results of our reenactment system. Corresponding run times are listed in Table 1. The length of the source and resulting output sequences is 965, 1436, and 1791 frames, respectively; the length of the input target sequences is 431, 286, and 392 frames, respectively.

CPU		GPU			FPS
SparseFT	MouthRT	DenseFT	DefTF	Synth	(Hz)
5.97ms	1.90ms	22.06ms	3.98ms	10.19ms	**27.6**
4.85ms	1.50ms	21.27ms	4.01ms	10.31ms	**28.1**
5.57ms	1.78ms	20.97ms	3.95ms	10.32ms	**28.4**

Table 1: Avg. run times for the three sequences of Fig. 5, from top to bottom. Standard deviations w.r.t. the final frame rate are 0.51, 0.56, and 0.59 fps, respectively. Note that CPU and GPU stages run in parallel.

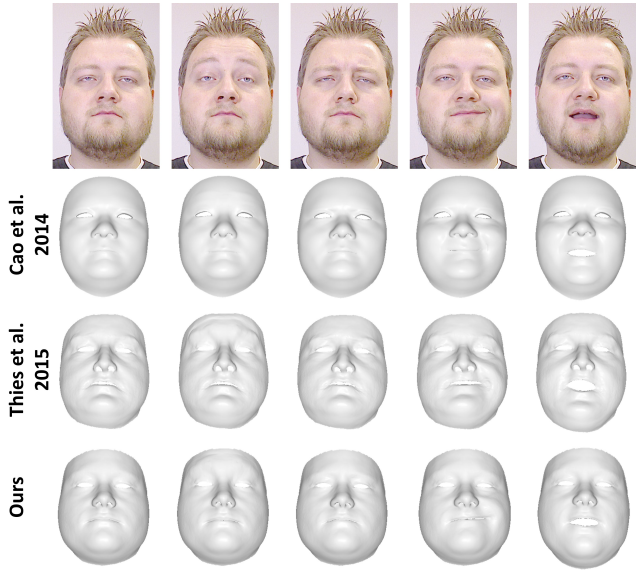

Figure 6: Comparison of our RGB tracking to Cao et al. [5], and to RGB-D tracking by Thies et al. [19].

Figure 7: Dubbing: Comparison to Garrido et al. [8].

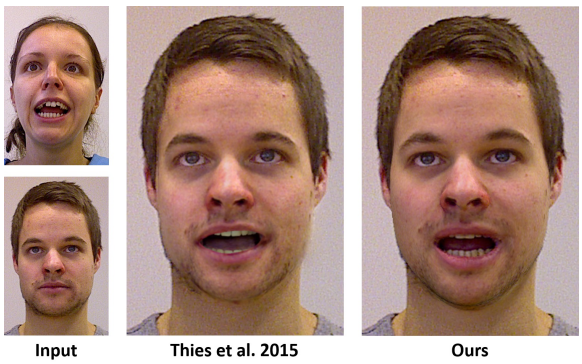

Figure 8: Comparison of the proposed RGB reenactment to the RGB-D reenactment of Thies et al. [19].

Cao et al. 2014 [5]: They capture face performance from monocular RGB in real-time. In most cases, our and their method produce similar high-quality results (see Fig. 6); our identity and expression estimates are slightly more accurate though.

Thies et al. 2015 [19]: Their approach captures face performance in real-time from RGB-D, Fig. 6. While we do not require depth data, results of both approaches are similarly accurate.

Reenactment Evaluation.

In Fig. 7, we compare our approach against state-of-the art reenactment by Garrido et al. [8]. Both methods provide highly-realistic reenactment results; however, their method is fundamentally offline, as they require all frames of a sequence to be present at any time. In addition, they rely on a generic geometric teeth proxy which in some frames makes reenactment less convincing. In Fig. 8, we compare against the work by Thies et al. [19]. Runtime and visual quality are similar for both approaches; however, their geometric teeth proxy leads to an undesired appearance of the reenacted mouth. Thies et al. use an RGB-D camera, which limits the application range; they cannot reenact Youtube videos.

12. LIMITATIONS

The assumption of Lambertian surfaces and smooth illumination is limiting, and may lead to artifacts in the presence of hard shadows or specular highlights; a limitation shared by most state-of-the-art methods. Scenes with face occlusions by long hair and a beard are challenging. Furthermore, we only reconstruct and track a low-dimensional blendshape model (76 coefficients), which omits fine-scale static and transient surface details. Our retrieval-based mouth synthesis assumes sufficient visible expression variation in the target sequence. On a too short sequence, or when the target remains static, we cannot learn the person-specific mouth behavior. In this case, temporal aliasing can be observed, as the target space of the retrieved mouth samples is too sparse. Another limitation is caused by our commodity hardware setup (webcam, USB, and PCI), which introduces a small delay of ≈ 3 frames.

13. DISCUSSION

Our face reconstruction and photo-realistic re-rendering approach enables the manipulation of videos at real-time frame rates. In addition, the combination of the proposed approach with a voice impersonator or a voice synthesis system, would enable the generation of made-up video content that could potentially be used to defame people or to spread so-called 'fake-news'. We want to emphasize that computer-generated content has been a big part of feature-film movies for over 30 years. Virtually every high-end movie production contains a significant percentage of synthetically-generated content (from Lord of the Rings to Benjamin Button). These results are already hard to distinguish from reality and it often goes unnoticed that the content is not real. Thus, the synthetic modification of video clips was already possible for a long time, but it was a time consuming process and required domain experts. Our approach is a game changer, since it enables editing of videos in real-time on a commodity PC, which makes this technology accessible to non-experts. We hope that the numerous demonstrations of our reenactment systems will teach people to think more critical about the video content they consume every day, especially if there is no proof of origin. The presented system also demonstrates the need for sophisticated fraud detection and watermarking algorithms. We believe that the field of digital forensics will receive a lot of attention in the future.

14. CONCLUSION

The presented approach is the first real-time facial reenactment system that requires just monocular RGB input. Our live setup enables the animation of legacy video footage – e.g., from Youtube – in real time. Overall, we believe our system will pave the way for many new and exciting applications in the fields of VR/AR, teleconferencing, or on-the-fly dubbing of videos with translated audio. One direction for future work is to provide full control over the target head. A properly rigged mouth and tongue model reconstructed from monocular input data will provide control over the mouth cavity, a wrinkle formation model will provide more realistic results by adding fine-scale surface detail and eye-tracking will enable control over the target's eye movement.

15. ACKNOWLEDGMENTS

We would like to thank Chen Cao and Kun Zhou for the blendshape models and comparison data, as well as Volker Blanz, Thomas Vetter, and Oleg Alexander for the provided face data. The facial landmark tracker was kindly provided by TrueVisionSolution. We thank Angela Dai for the video voice over and Daniel Ritchie for video reenactment. This research is funded by the German Research Foundation (DFG), grant GRK-1773 Heterogeneous Image Systems, the ERC Starting Grant 335545 CapReal, and the Max Planck Center for Visual Computing and Communications (MPC-VCC). We also gratefully acknowledge the support from NVIDIA Corporation for hardware donations.

16. REFERENCES

[1] V. Blanz and T. Vetter. A morphable model for the synthesis of 3d faces. In *Proc. SIGGRAPH*, pages 187–194. ACM Press/Addison-Wesley Publishing Co., 1999.

[2] S. Bouaziz, Y. Wang, and M. Pauly. Online modeling for realtime facial animation. *ACM TOG*, 32(4):40, 2013.

[3] C. Bregler, M. Covell, and M. Slaney. Video rewrite: Driving visual speech with audio. In *Proc. SIGGRAPH*, pages 353–360. ACM Press/Addison-Wesley Publishing Co., 1997.

[4] C. Cao, D. Bradley, K. Zhou, and T. Beeler. Real-time high-fidelity facial performance capture. *ACM TOG*, 34(4):46:1–46:9, 2015.

[5] C. Cao, Q. Hou, and K. Zhou. Displaced dynamic expression regression for real-time facial tracking and animation. *ACM TOG*, 33(4):43, 2014.

[6] Y.-L. Chen, H.-T. Wu, F. Shi, X. Tong, and J. Chai. Accurate and robust 3d facial capture using a single rgbd camera. *Proc. ICCV*, pages 3615–3622, 2013.

[7] P. Garrido, L. Valgaerts, O. Rehmsen, T. Thormaehlen, P. Perez, and C. Theobalt. Automatic face reenactment. In *Proc. CVPR*, 2014.

[8] P. Garrido, L. Valgaerts, H. Sarmadi, I. Steiner, K. Varanasi, P. Perez, and C. Theobalt. Vdub: Modifying face video of actors for plausible visual alignment to a dubbed audio track. In *Computer Graphics Forum*. Wiley-Blackwell, 2015.

[9] L. Hu, S. Saito, L. Wei, K. Nagano, J. Seo, J. Fursund, I. Sadeghi, C. Sun, Y. Chen, and H. Li. Avatar digitization from a single image for real-time rendering. *ACM Trans. Graph.*, 36(6):195:1–195:14, 2017.

[10] I. Kemelmacher-Shlizerman, A. Sankar, E. Shechtman, and S. M. Seitz. Being john malkovich. In *Computer Vision - ECCV 2010, 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part I*, pages 341–353, 2010.

[11] H. Li, J. Yu, Y. Ye, and C. Bregler. Realtime facial animation with on-the-fly correctives. *ACM TOG*, 32(4):42, 2013.

[12] K. Li, F. Xu, J. Wang, Q. Dai, and Y. Liu. A data-driven approach for facial expression synthesis in video. In *Proc. CVPR*, pages 57–64, 2012.

[13] R. Ramamoorthi and P. Hanrahan. A signal-processing framework for inverse rendering. In *Proc. SIGGRAPH*, pages 117–128. ACM, 2001.

[14] J. M. Saragih, S. Lucey, and J. F. Cohn. Deformable model fitting by regularized landmark mean-shift. *IJCV*, 91(2):200–215, 2011.

[15] J. M. Saragih, S. Lucey, and J. F. Cohn. Real-time avatar animation from a single image. In *Automatic Face and Gesture Recognition Workshops*, pages 213–220, 2011.

[16] F. Shi, H.-T. Wu, X. Tong, and J. Chai. Automatic acquisition of high-fidelity facial performances using monocular videos. *ACM TOG*, 33(6):222, 2014.

[17] C. Siegl, V. Lange, M. Stamminger, F. Bauer, and J. Thies. Faceforge: Markerless non-rigid face multi-projection mapping. *IEEE Transactions on Visualization and Computer Graphics*, 2017.

[18] R. W. Sumner and J. Popović. Deformation transfer for triangle meshes. *ACM TOG*, 23(3):399–405, 2004.

[19] 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)*, 34(6), 2015.

[20] J. Thies, M. Zollhöfer, M. Stamminger, C. Theobalt, and M. Nießner. Demo of face2face: Real-time face capture and reenactment of rgb videos. In *ACM SIGGRAPH 2016 Emerging Technologies*, SIGGRAPH '16, pages 5:1–5:2, New York, NY, USA, 2016. ACM.

[21] J. Thies, M. Zollhöfer, M. Stamminger, C. Theobalt, and M. Nießner. Face2Face: Real-time Face Capture and Reenactment of RGB Videos. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2016.

[22] J. Thies, M. Zollhöfer, M. Stamminger, C. Theobalt, and M. Nießner. FaceVR: Real-Time Facial Reenactment and Eye Gaze Control in Virtual Reality. *ArXiv, non-peer-reviewed prepublication by the authors*, abs/1610.03151, 2016.

[23] D. Vlasic, M. Brand, H. Pfister, and J. Popović. Face transfer with multilinear models. *ACM TOG*, 24(3):426–433, 2005.

[24] T. Weise, S. Bouaziz, H. Li, and M. Pauly. Realtime performance-based facial animation. 30(4):77, 2011.

[25] T. Weise, H. Li, L. V. Gool, and M. Pauly. Face/off: Live facial puppetry. In *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer animation (Proc. SCA'09)*, ETH Zurich, August 2009. Eurographics Association.