Deep Incremental Learning for Efficient High-Fidelity Face Tracking

CHENGLEI WU, Facebook Reality Labs
TAKAAKI SHIRATORI, Facebook Reality Labs
YASER SHEIKH, Facebook Reality Labs

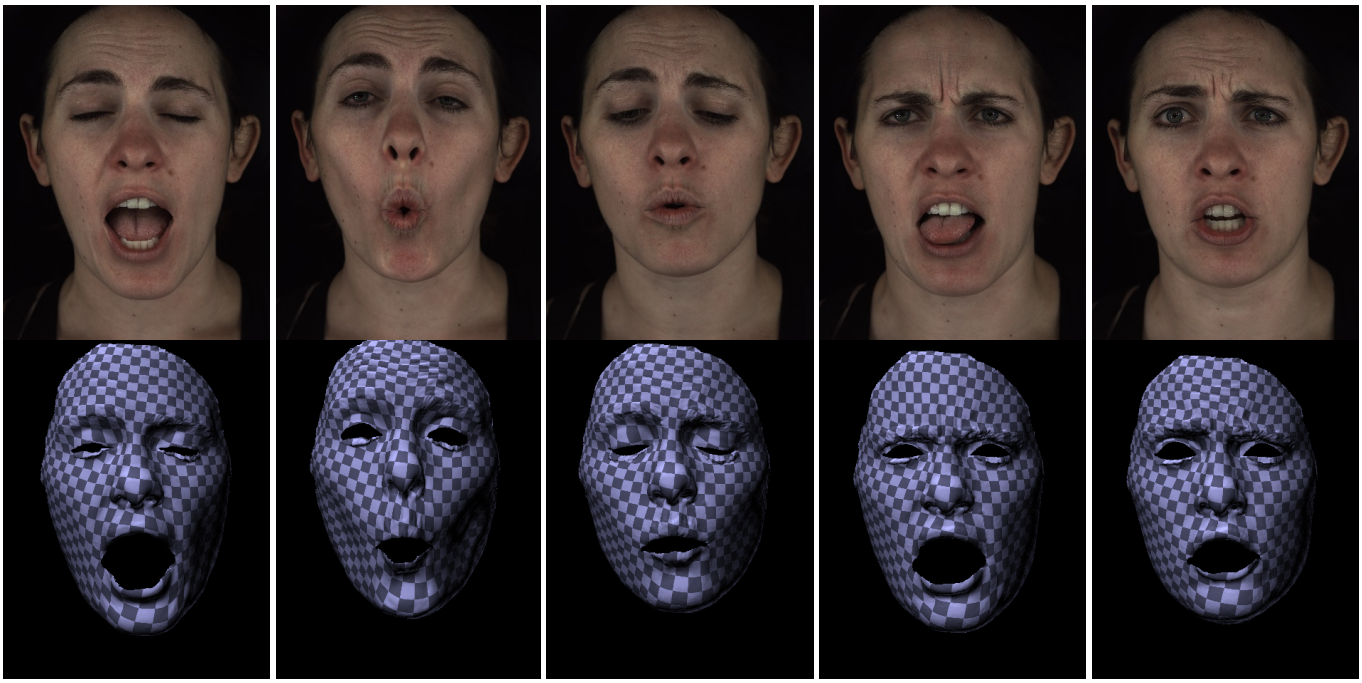


Fig. 1. Given multi-view images (top), our framework alternates the modeling step for our statistical model using tracked meshes and texture maps, and the tracking step from prediction provided by the model, and achieves resistance to drift, significant robustness to fast motion and massive parallelization with high-fidelity (bottom).

In this paper, we present an incremental learning framework for efficient and accurate facial performance tracking. Our approach is to alternate the modeling step, which takes tracked meshes and texture maps to train our deep learning-based statistical model, and the tracking step, which takes predictions of geometry and texture our model infers from measured images and optimize the predicted geometry by minimizing image, geometry and facial landmark errors. Our *Geo-Tex VAE* model extends the convolutional variational autoencoder for face tracking, and jointly learns and represents deformations and variations in geometry and texture from tracked meshes and texture maps. To accurately model variations in facial geometry and texture, we introduce the *decomposition* layer in the Geo-Tex VAE architecture which decomposes the facial deformation into global and local components.

We train the global deformation with a fully-connected network and the local deformations with convolutional layers. Despite running this model on each frame independently – thereby enabling a high amount of parallelization – we validate that our framework achieves sub-millimeter accuracy on synthetic data and outperforms existing methods. We also qualitatively demonstrate high-fidelity, long-duration facial performance tracking on several actors.

CCS Concepts: • **Computing methodologies** → **Motion capture**; Learning latent representations;

Additional Key Words and Phrases: facial performance tracking, variational autoencoder

ACM Reference Format:
Chenglei Wu, Takaaki Shiratori, and Yaser Sheikh. 2018. Deep Incremental Learning for Efficient High-Fidelity Face Tracking. *ACM Trans. Graph.* 37, 6, Article 234 (November 2018), 12 pages. https://doi.org/10.1145/3272127.3275101

Authors' addresses: Chenglei Wu, Facebook Reality Labs, Pittsburgh, PA, chenglei@fb.com; Takaaki Shiratori, Facebook Reality Labs, Pittsburgh, PA, tshiratori@fb.com; Yaser Sheikh, Facebook Reality Labs, Pittsburgh, PA, yaser@fb.com.

0730-0301/2018/11-ART234
https://doi.org/10.1145/3272127.3275101

1 INTRODUCTION

The human face communicates surprisingly various emotions through a subtle range of facial expressions. Many non-verbal signals through

facial expressions are recognized independently of cultures, languages and society. Thus, capturing such dynamic details rich in subtle motion is crucial for developing engaging experiences and interactions in movies, video games, and virtual/augmented reality (VR/AR), and enables to not only add lifelike realism to characters but also tell an appealing and immersive story via verbal and non-verbal cues.

Tracking with such high fidelity still remains challenging despite significant recent progress [Beeler et al. 2011; Fyffe et al. 2014, 2017; Klaudiny and Hilton 2012]. Complex subtle facial expressions and large deformations requires tracking to be very robust to occlusions and appearance changes. This is challenging even when capturing an actor with a multi-view camera setup. It is common to add temporal information or constraints in attempts to mitigate these problems, however, this does not completely resolve the drift issue (*i.e.,* error accumulation over a sequence). Moreover, temporal constraints require tracking methods to be processed in a sequential manner, typically resulting in huge computational cost.

In this paper, we present an incremental learning framework for efficient and accurate facial performance tracking from multi-view images. We alternate the modeling step, which takes tracked meshes and baked-in texture maps to train our deep learning-based statistical model, and the tracking step, which takes predictions of geometry and texture our model infers from measured images as initialization and optimizes the predicted geometry by minimizing image and geometric errors. Key to our approach is *Geo-Tex VAE*, a convolutional variational autoencoder (VAE) [Kingma and Welling 2014] designed for face tracking. Geo-Tex VAE jointly learns and represents deformation and variations in geometry and texture from tracked meshes and texture maps. Trained Geo-Tex VAE can predict mesh and texture map that are close to target images to be tracked enought to satisfy the brightness constancy assumption, and enables the tracking step to be free from any temporal dependency. This achieves high resistance to drift, significant robustness to very fast motion, occlusions and appearance changes, and massive parallelization on a compute server, enabling long-duration facial performance tracking. Additionally, the tracking step does not consider any regularization based on the model, and thus potentially produces mesh and texture that cannot be expressed by the model. The incremental learning framework could further improve the models, and accordingly tracking quality.

Geo-Tex VAE has important properties over conventional VAE for the purpose of high-fidelity face tracking. Following the observation that geometry and baked-in texture show strong correlation such as wrinkles and creases, Geo-Tex VAE learns geometric deformation and texture variations jointly. This allows us to predict geometry and texture at runtime and makes the tracking optimization much more accurate. Inspired by the conventional graphics pipeline – which combined traditional blendshapes with corrective blendshapes – we introduce a *Decomposition Layer* in our VAE that decomposes facial deformation in geometry and texture into global and local deformations. The global deformation is modeled with a fully-connected network and local deformations are modeled with non-linear convolutional layers. Taking the predicted geometry and texture as initialization, the tracking step directly optimizes vertex locations and surface orientations by minimizing differences of images, geometry, and facial landmarks between measurements and predictions, together with conventional geometric regularization. We demonstrate that our framework achieves sub-millimeter accuracy on average over more than one thousand frames of a synthetic sequence, and also demonstrate the effectiveness and efficiency of our method with real data. Our method also qualitatively outperforms state-of-the-art tracking methods.

The technical contributions of this paper are summarized as:

- an incremental learning framework for high-fidelity facial performance tracking;
- joint representation and learning of geometry and texture to learn their correlation;
- effective convergence in Geo-Tex VAE training by decomposing geometry and texture deformations into global and local ones;
- highly precise and accurate mesh tracking with capabilities of drift resistance and parallelization.

These capabilities enable long-duration facial performance tracking (Fig. 1), and potentially open up a range of interesting research directions on speech animation [Karras et al. 2017; Taylor et al. 2017] and social interaction modeling for social VR/AR.

2 RELATED WORK

Our method consists of modeling and tracking steps. Here, we discuss existing work on face tracking and modeling, and clarify the novelty of our approach.

3D Face Tracking. Most face tracking methods can be categorized into two types based on a camera configuration: Multi-view approaches fully utilize strong geometric cues from calibrated images and thus does not rely on a face model, while single-view approaches (*e.g.,* a single color/depth/RGB-D camera) rely on a face model because geometric cues are noisy or unavailable.

For a multi-view setup, *scene flow*, 3D motion estimation techniques for every pixel or vertex, is widely used to update the 3D position of each vertex from one frame to another [Basha et al. 2013; Valgaerts et al. 2010; Vedula et al. 2005]. However naive sequential tracking based on scene flow is susceptible to drift because of occlusions and appearance changes. Drift correction is required after scene flow computation, such as texture alignment in a UV space [Bradley et al. 2010], or minimizing differences between measured images and images rendered with a tracked mesh [Valgaerts et al. 2012]. More recent works are focused on *non-sequential* tracking by dividing a sequence into short segments based on image and/or geometry similarity. This enables tracking to run on each segment individually, thus limiting drift effects and reducing computational time via parallelization. Beeler and colleagues [2011] detect anchor frames based on image similarity to a neutral expression, and apply sequential tracking between consecutive anchor frames. Klaudiny and Hilton [2012] compute a minimum spanning tree based on geometric similarity among each pair of frames, and run tracking from the root node to each leaf node. Fyffe and colleagues [2014] introduce a *performance flow graph*, a similarity graph among static scans and every frame in a sequence. Most similar to our approach is Fyffe and colleagues' method [2017] that achieves

registration from a template face mesh to each frame independently by optimizing 3D landmark positions and then optical flow-based appearance correspondences. This approach yields most minimal drift and massive parallelization among non-sequential methods, regardless of length of a sequence. Our method qualitatively achieves higher precision and accuracy than their methods, thanks to our VAE model that can predict both geometry and texture for tracking initialization.

For a single-view setup, a 3D face model plays an important role to compensate for noisy or absent geometric cues as mentioned earlier. Holistic linear models such as linear blendshape [Lewis et al. 2014] and principal component analysis (PCA) models are empirically known to have a reasonable capacity to represent various facial deformation. Many existing approaches use holistic linear models for real-time facial tracking, which optimize rigid head pose and model coefficients by minimizing facial landmarks, depth and/or image differences [Cao et al. 2014a, 2013; Thies et al. 2015, 2016; Weise et al. 2011]. Region-based linear models [Tena et al. 2011; Weise et al. 2009; Wu et al. 2016] has a higher model capacity, and therefore allows more accurate tracking. Other recent work has captured wrinkle-level details jointly with holistic deformations based on a regression model from an image to wrinkle map [Cao et al. 2015] or from holistic model parameters to detail maps [Ichim et al. 2015], or by using image-based shading cues [Garrido et al. 2013, 2016; Shi et al. 2014; Suwajanakorn et al. 2014]. Huang and colleagues [2011] leverage high-resolution face scans for high-fidelity facial performance tracking from sparse motion capture markers. One major issue of the above approaches is that a model is trained during a preprocessing step and fixed during tracking: If a facial expression lies outside of a model space, tracking will fail. This issue was mitigated by alternating blendshape model refinement and tracking [Li et al. 2010], or incrementally learning correctives [Bouaziz et al. 2013; Li et al. 2013]. Our method also employs incremental learning for the Geo-Tex VAE model, and progressively learns correlation of geometry and texture as more data are tracked.

Parametric Face Models. While many existing methods focused on parametric representations of face geometry only, such as blendshapes [Lewis et al. 2014], PCA models [Weise et al. 2009], multilinear model [Vlasic et al. 2005] or a facial rig consisting blendshapes, joints and correctives [Li et al. 2017], modeling correlation between geometry and appearance is gathering more attention recently, such as regression from an image to 2D/3D facial landmark positions [Cao et al. 2013, 2014b; Saragih et al. 2011], to low-resolution face mesh [Booth et al. 2017], to a wrinkle map [Cao et al. 2015], or inferring a high-frequency albedo map from facial geometry and a low-frequency albedo map [Saito et al. 2017]. To the best of our knowledge, joint representation and modeling of geometry and texture, which is the core of our modeling method, has not been explored sufficiently. Blanz and colleagues introduced a linear morphable model [Blanz and Vetter 1999] for 3D face synthesis, and Matthews introduced an active appearance model [Matthews and Baker 2004] for 2D tracking. Tewari and colleagues used a deep autoencoder network to model geometry and texture morphable model to infer shape [2018; 2017], which relies on linear models for geometry and appearance to create training data of codes of the network. Luan and colleagues [2018] learned a nonlinear 3D face morphable model from a set of in-the-wild face images without 3D scans. Huynh and colleagues [2018] used a deep learning-based model to infer mesoscopic structure of a face from a low-resolution mesh and flat-lit texture. While many of the abovementioned approaches model identity differences of faces as well as expression-related deformation, our model, which is person-specific and does not consider uncontrolled lighting, yet has a strong capacity to jointly represent, model and predict global and non-linear local deformations in geometry and texture at high resolution.

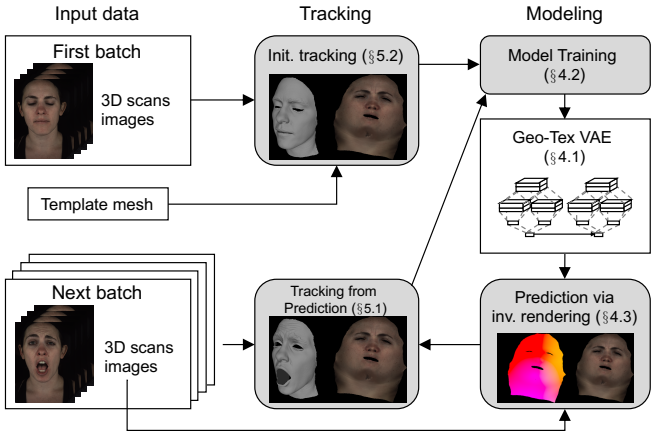


Fig. 2. Overview of our incremental learning framework. Given tracked meshes and texture maps from the initial tracking step, we alternate the model training, prediction and tracking steps until training convergence or the end of a sequence.

3 OVERVIEW

The input data to our tracking method is a sequence of multi-view images captured by calibrated cameras. Additionally, we run the PatchMatch-based multi-view stereo method [S. Galliani and Schindler 2015] and obtain a 3D scan for every frame.

Our approach uses a template mesh that is tracked over the input sequence. The three requirements for the template mesh are 1) the template mesh needs to be tightly registered to the first frame of the sequence, 2) a UV space is defined, and 3) upper and lower eyelids and upper and lower lips are topologically disconnected. In our implementation, we use the 3D scan of the first frame as a template, and create a UV space using commercial graphics software called ZBrush[1].

Fig. 2 illustrates the pipeline of our framework to incremental learning and tracking of facial performance. Our tracking method starts with initial tracking based on local linear models, followed by model-free optimization, for the first batch, *i.e.,* a few hundred frames in the beginning of the input sequence (Section 5.2). Our current implementation assumes that the first batch contains facial deformation. This step could be easily automated by checking variances of vertex displacements in the tracked meshes. The tracked mesh and corresponding baked-in texture maps in the batch are

[1]http://pixologic.com/

passed to train the Geo-Tex VAE model. The architecture of Geo-Tex VAE is designed to model global and local deformations in geometry and texture jointly (Section 4.1). Geo-Tex VAE is trained with the batch by minimizing differences between tracked and predicted meshes and texture maps (Section 4.2). Once training is completed, Geo-Tex VAE is used to predict meshes and texture maps from images in the next batch (Section 4.3). Taking the predictions as initialization, we apply our tracking method to tightly register predicted meshes to measured images and 3D scans (Section 5.1). As this tracking step is independent of the Geo-Tex VAE model except for initialization, it potentially yields data that cannot be expressed by the current model and further improves the Geo-Tex VAE model through incremental learning. Besides, the tracking does not rely on any temporal information, and can process each frame independently, which yields minimal drift and benefits from massive parallelization. The incremental learning and tracking framework is applied until it reaches the end of a sequence or convergence of Geo-Tex VAE training. Once the model is trained with sufficient facial expressions, Geo-Tex VAE prediction enables to run only the tracking step for new sequences of the actor without training the model, achieving further reduction of computational cost.

The first batch size is set to 350 frames. Geo-Tex VAE from the first batch may not be able to provide good predictions because possibly insufficient training data. Therefore, we set the second batch size to 32 frames, and increase the size by 8 incrementally until it reaches 128 frames.

4 MODELING AND PREDICTING FACIAL GEOMETRY AND TEXTURE

A VAE is an unsupervised deep learning technique for feature learning, and has proved its strong capacity to encode training data [Kingma and Welling 2014]. We present *Geo-Tex VAE*, a convolutional VAE that jointly encodes and decodes geometry and texture deformation of a face. We incrementally train Geo-Tex VAE: Once the tracking step processes a certain amount of frames, the tracked meshes together with corresponding baked-in texture maps are used to train and update Geo-Tex VAE.

4.1 Geo-Tex VAE Architecture

Based on the observation that geometry and baked-in texture are correlated, we jointly parameterize and model geometry and texture. We utilize a UV space of the template mesh (512×512 pixels in our implementation), and project mesh geometry and texture into a 6-channel UV image, 3 channels for a 3D position map S and 3 channels for a texture map T.

The convolution operation in a typical convolutional VAE exhibits a stationarity assumption [Taigman et al. 2014] that does not hold for facial texture and motion data due to a surprising amount of variability in expressions. Our preliminary experiments share a similar observation that a typical convolutional VAE architecture caused poor training convergence (see Section 6.3). Inspired by existing graphics pipelines that combine a linear global model (*e.g.,* blendshapes) and correctives, we mitigate this problem through the *decomposition layer* in the Geo-Tex VAE architecture that decomposes entire face deformations in geometry and texture into global and local deformations, and encodes/decodes these deformations jointly in the end (Fig. 3a). The decomposition layer is a multilayer perceptron (MLP) to encode the original deformation and decode it as global deformations, S_G and T_G[2], and the reconstruction error with the MLP is then computed as local deformations, S_L and T_L (Fig. 3b). As it is reasonable to assume that the local deformations satisfy the stationarity assumption of the convolution operation, the local deformations are encoded/decoded with the convolutional layers. The convolutional layers consist of four residual blocks [He et al. 2016], each of which applies two convolutions with rectified linear units (ReLU) for non-linearity (Fig. 3c). Finally, latent variables of global and local deformations, $\mathbf{z}_G$ and $\mathbf{z}_L$ respectively, are merged and jointly modeled with another MLP for $\mathbf{z}$ whose posterior is parameterized with mean μ and variance σ.

4.2 Training Geo-Tex VAE

After tracking a sequence of frames, tracked meshes and corresponding images serve as training data for the Geo-Tex VAE. The position map S is simply obtained by projecting each vertex into the UV space and the texture map T is obtained by projecting all captured images into the UV space and merging them with Poisson blending [Pérez et al. 2003].

Every position map contains not only facial deformation but also rigid head motion. Because Geo-Tex VAE is designed to learn a deformation map, such rigid motion needs to be removed from S. To rigidly align all S to a reference position map S_{ref} (*e.g., S* at the first frame), we design and train a rigid stabilization network (Fig. 4). The rigid stabilization network applies a learned rigidity confidence map W to S and S_{ref} and then the rigid registration layer that solves head rotation $\mathbf{R}_{\text{H}}$ and translation $\mathbf{t}_{\text{H}}$ based on W in a closed-form manner via singular value decomposition (SVD) [Arun et al. 1987]. This network is analogous to rigid stabilization based on a skull model for blendshape generation [Beeler and Bradley 2014] but ours does not require any manual intervention.

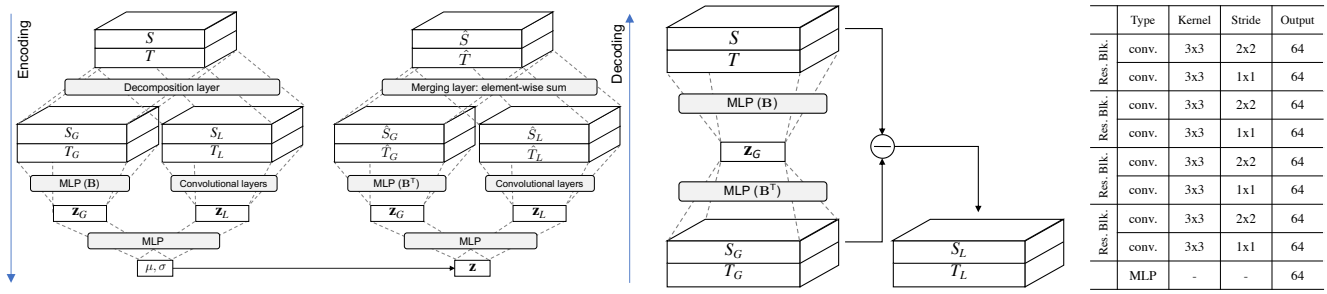
For training the rigid stabilization network, we minimize the loss L_{rigid} that includes a regularization term to avoid a trivial solution (*i.e.,* all confidences in W are zero) as well as $\mathbf{R}_{\text{H}}$ and $\mathbf{t}_{\text{H}}$ with the Adam optimizer [Kingma and Ba 2015]

$$\begin{aligned} L_{\text{rigid}} = &\left\| (W \odot S) \otimes (\mathbf{R}_{\text{H}}, \mathbf{t}_{\text{H}}) - W \odot S_{\text{ref}} \right\|_F^2 \\ &+ \lambda_{\text{rigid}} \min\left(\rho N - \|W\|_F^2, 0 \right), \end{aligned} \tag{1}$$

where $\odot$ is element-wise multiplication, $\otimes$ is element-wise rigid transformation, and $\|\cdot\|_F$ is the Frobenius norm. N is the total number of pixels in the UV space, ρ is a proportion of the number of pixels with nonzero confidence to N (set to 0.4), and λ_{rigid} is a weight for this regularization loss (set to 10). To constrain W to be always positive, an intermediate variable G is defined so that W is the sigmoid function on G and ranges from 0 to 1. Note that there is no parameter in the rigid registration layer because of the SVD-based closed-form solution.

Now that we obtain rigidly registered position maps S_{aligned} for all training data, the next step is to train Geo-Tex VAE. Unfortunately, the large number of parameters in MLPs easily causes overfitting

[2]The output of the MLP is reshaped into image form as shown in Fig. 3b



(a) Geo-Tex VAE. (b) Decomposition layer. (c) Convolutional layers.

Fig. 3. Geo-Tex VAE architecture. The decomposition layer decomposes deformations of facial geometry and texture into global and local ones. Geo-Tex VAE models global and local deformationss with MLP and convolutional layers, respectively, followed by joint encoding of those two.

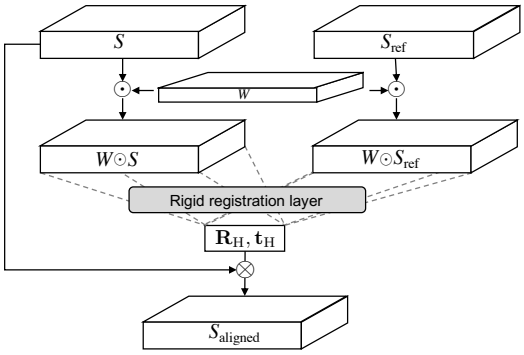


Fig. 4. Rigid stabilization network.

and very poor generalizability. Therefore, instead of optimizing all parameters, we compute PCA bases $\mathbf{B}$ from a training data set $\{S_{\text{aligned}}, T\}$. $\mathbf{B}$ is used for MLP related to global deformation (*i.e.,* MLPs associated with $\mathbf{B}$ or $\mathbf{B}^{\mathsf{T}}$ in Fig. 3), and fixed in the following Geo-Tex VAE training. The rest parameters are optimized by minimizing the data loss L_{data} as well as the Kullback-Leibler divergence loss with the weight of 10^{-4} with the Adam optimizer:

$$L_{\text{data}} = \lambda_{\text{S}}\|\hat{S}-S_{\text{aligned}}\|_F^2+\lambda_{\text{L}}\|L(\hat{S})-L(S_{\text{aligned}})\|_F^2+\lambda_{\text{T}}\|\hat{T}-T\|_F^2, \quad (2)$$

where $L(\cdot)$ computes surface Laplacian, $\lambda_{\text{S}}, \lambda_{\text{L}}, \lambda_{\text{T}}$ are weights for reconstruction error in 3D position map S, surface Laplacian map $L(S)$, and texture map T (set to 10, 10^5 and 10^{-3}), respectively. We found the surface Laplacian loss useful to reduce block artifacts caused by convolutions.

4.3 Predicting Geometry and Texture with Inverse Rendering

At runtime we use the decoder from the trained Geo-Tex VAE to predict mesh and texture from a set of multivew images to perform tracking. This could be considered as an inverse rendering problem, as we want to infer latent variables $\mathbf{z}$ so that images synthesized from $\hat{S}$ and $\hat{T}$ predicted with $\mathbf{z}$ and camera parameters match to the actual images. Note that we infer baked-in texture maps, rather than albedo maps [Saito et al. 2017; Tewari et al. 2017].

Because the Geo-Tex VAE is trained only for deformation and does not handle rigid transformation, we need to optimize $\mathbf{z}$ along with rigid head rotation $\mathbf{R}_{\text{H}}$ and translation $\mathbf{t}_{\text{H}}$. We formulate three types of losses: the image loss L_{img}, the geometry loss L_{geo}, the landmark loss L_{land} and a regularization term on $\mathbf{z}$. We optimize the parameters by minimizing the total inverse rendering loss L_{IR}:

$$L_{\text{IR}} = \lambda_{\text{img}}L_{\text{img}} + \lambda_{\text{geo}}L_{\text{geo}} + \lambda_{\text{land}}L_{\text{land}} + \lambda_{\text{reg}}\|\mathbf{z}\|^2, \quad (3)$$

where λ_{img}, λ_{geo}, λ_{land} and λ_{reg} are weights for L_{img}, L_{geo}, L_{land} and L_{reg} (set to 1, 10, 0.1 and 1), respectively.

The image loss L_{img} considers the sum of pixel-wise intensity differences between captured images I and synthesized images $\hat{I}$, defined as

$$L_{\text{img}} = \sum_c \|I_c - \hat{I}_c(\mathbf{R}_{\text{H}}, \mathbf{t}_{\text{H}}, \mathbf{z})\|_1, \quad (4)$$

where c represents a camera. We use the L1 norm for robustness to outliers such as specularity and appearance changes.

The geometry loss L_{geo} computes differences between predicted geometry and a 3D scan. Similar to the image loss, we render a 3D position map for each view using the 3D scan and the predicted geometry, D and $\hat{D}$, respectively, and compute pixel-wise position differences, as

$$L_{\text{geo}} = \sum_c \|D_c - \hat{D}_c(\mathbf{R}_{\text{H}}, \mathbf{t}_{\text{H}}, \mathbf{z})\|_1. \quad (5)$$

The landmark loss computes reprojection errors between measured and predicted landmarks. We use the convolutional pose machine method (CPM) [Wei et al. 2016] trained with annotated facial landmarks:

$$L_{\text{land}} = \sum_f \left\| \mathbf{R}_{\text{H}}\hat{S}_{\mathbf{z}}(\mathbf{u}_f^{\text{ref}}) + \mathbf{t}_{\text{H}} - \mathbf{f}_f \right\|^2, \quad (6)$$

where $\mathbf{u}^{\text{ref}}$ is a landmark location in the UV space, $\mathbf{f}$ is the 3D position of a landmark detected in an input image (*i.e.*, projecting 2D landmarks in the input image to the corresponding 3D scan). $\mathbf{u}^{\text{ref}}$ is obtained by projecting landmarks detected in the reference image into S_{ref}. Fig. 5 shows example input images for prediction, meshes and texture maps predicted with the trained model. Subtle wrinkles as well as holistic expressions are predicted in both meshes and texture reasonably close to the input images.

Fig. 5. Example meshes and texture predicted with the Geo-Tex VAE. Subtle details such as wrinkles highlighted with red arrows as well as holistic expressions are predicted.

Once the inverse rendering is completed by minimizing Eq. (3) with the Adam optimizer, we decode $\mathbf{z}$ with the decoder of Geo-Tex VAE to obtain $\hat{S}$ and $\hat{T}$, and apply the head rigid pose $\mathbf{R}_{\mathrm{H}}$ and $\mathbf{t}_{\mathrm{H}}$ to $\hat{S}$. Then these predictions are used to initialize the tracking step. Note that there is no temporal dependency in the prediction step, and therefore we can run this prediction step for each frame in parallel.

5 FACIAL SURFACE TRACKING

Now that Geo-Tex VAE is trained with tracked meshes and corresponding textures in previous batches, the next step is to run face tracking for a next batch. We use the inverse rendering-based prediction as initialization, and optimize the vertex positions of the template mesh based on images and 3D scan for each frame in the next batch individually. Because our tracking formulation does not include any model-based regularization, tracked meshes could lie outside of the space spanned by Geo-Tex VAE, and thus could further improve the Geo-Tex VAE in next batch training.

5.1 Facial Surface Tracking from Prediction

Similar to the prediction step, we synthesize image $\hat{I}$ and 3D position map $\hat{D}$ for each view based on the prediction for a target frame, and minimize differences between those and the images I and the position maps D measured at the target frame. Unlike previous methods that updates vertex positions via optical flow and triangulation computed in a sequential manner [Beeler et al. 2011; Bradley et al. 2010; Fyffe et al. 2014], we directly optimize the 3D positions $\mathbf{X}$ and surface orientation $\mathbf{R}$ of all vertices by minimizing the total objective E_{track}:

$$E_{\mathrm{track}} = w_{\mathrm{pho}}E_{\mathrm{pho}} + w^0_{\mathrm{pho}}E^0_{\mathrm{pho}} + w_{\mathrm{geo}}E_{\mathrm{geo}} + w_{\mathrm{feat}}E_{\mathrm{feat}} + E_{\mathrm{reg}}, \quad (7)$$

where E_{pho}, E^0_{pho}, E_{geo}, E_{feat} and E_{reg} are the photo-consistency term w.r.t. synthesized image, the photo-consistency term w.r.t. the images at the first frame of a sequence, the geometric consistency term, the 3D feature distance term and the regularization term, which are weighted with w_{pho}, w^0_{pho}, w_{geo}, and w_{feat} (set to 10, 10, 10 and 50), respectively.

Photo-consistency Terms E_{pho} and E^0_{pho}. Typical approaches in optical flow and scene flow computation compares rectangular patches in image domain. This assumption holds only if the target surface is frontal parallel. For more accurate photo-consistency, we consider a 3D local tangent plane for each $\mathbf{X}$, and transform a patch around 2D projection of $\mathbf{X}$ between I and $\hat{I}$ via a homography $\mathbf{H}$ parameterized with $\mathbf{X}$ and $\mathbf{R}$ [Hartley and Zisserman 2004]. For robustness to local brightness changes, we consider enhanced correlation coefficients [Evangelidis and Psarakis 2008]. Accordingly, E_{pho} is formulated as

$$E_{\mathrm{pho}} = \sum_v \sum_{c \in C(\mathbf{X}_v)} \psi\left(\left\| \frac{\bar{\hat{I}}_c(\mathbf{P}_c\hat{\mathbf{X}}_v)}{\|\bar{\hat{I}}_c(\mathbf{P}_c\hat{\mathbf{X}}_v)\|} - \frac{\bar{I}_c(\mathbf{H}^c_v(\mathbf{P}_c\mathbf{X}_v))}{\|\bar{I}_c(\mathbf{H}^c_v(\mathbf{P}_c\mathbf{X}_v))\|} \right\| \right), \quad (8)$$

where where $\bar{I}$ is an image patch with mean intensity subtracted, $C(\mathbf{X})$ is a set of cameras where $\mathbf{X}$ is visible, $\hat{\mathbf{X}}$ is the 3D position of a mesh vertex from the Geo-Tex VAE prediction and $\mathbf{P}_c$ is the camera matrix of camera c. $\psi(\cdot)$ is a robust kernel to handle outliers [Zollhöfer et al. 2014], formulated as

$$\psi(e) = \min_{\omega}(2\omega^2 e^2/\gamma^2 + (1-\omega^2)^2), \quad (9)$$

where γ for E_{pho} is set to 0.1

E^0_{pho} is the photo-consistency term w.r.t. the images at the first frame of a sequence. Therefore, this is computed by simply replacing synthesized images $\hat{I}$ and predicted positions $\hat{\mathbf{X}}$ in Eq. (8) to those at the first frame. We set the patch size to 15×15 pixels for these photo-consistency terms.

Geometric Consistency Term E_{geo}. We assume the same local planarity around each vertex for geometry, and compute Euclidean distance for each 3D position in a patch, as

$$E_{\mathrm{geo}} = \sum_v \sum_{c \in C(\mathbf{X}_v)} \psi\left(\|\mathbf{R}_v(\hat{D}(\mathbf{P}_c\hat{\mathbf{X}}_v) - \hat{\mathbf{X}}_v) + \mathbf{X}_v - D(\mathbf{H}^c_v(\mathbf{P}_c\mathbf{X}_v))\| \right). \quad (10)$$

We set the patch size for E_{geo} to 7×7 pixels, and γ in the robust kernel for E_{geo} is set to 1.

3D Feature Distance Term E_{feat}. We also consider image-based features and minimize their 3D distances. In addition to landmarks detected by CPM, we also detect SIFT features [Lowe 2004]. For the eyelids, we use eyelid curve fitting [Wen et al. 2017] to improve and densify eyelid landmarks. For SIFT, we run descriptor-based matching between images to prune outlier correspondences.

We apply these feature detectors to the measured frontal image, and get their 3D positions $\mathbf{p}$ from the corresponding 3D scan. For the synthesized frontal image, we similarly detect the same features, and parameterize their 3D positions $\mathbf{q}$ with vertex positions $\mathbf{X}$ via barycentric coordinates in the template mesh. The optimization minimizes the Euclidean distance between $\mathbf{p}$ and $\mathbf{q}$ with the robust

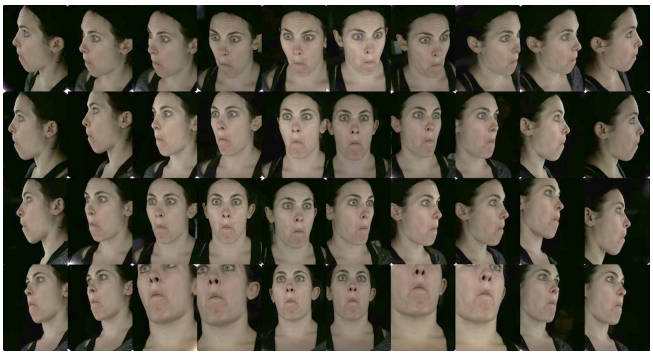


Fig. 6. Example image set captured by our multi-view camera setup.

kernel ψ:

$$E_{\text{feat}} = \sum_f \psi(\|\mathbf{p}_f - \mathbf{q}_f(\mathbf{X})\|), \tag{11}$$

where γ in the robust kernel for E_{feat} is set to 10.

Regularization Term E_{reg}. We also consider the regularization term, mainly for textureless regions and regions that are sometimes occluded such as regions around eye creases and inner lips. We consider conventional Laplacian and as-rigid-as-possible (ARAP) [Sorkine and Alexa 2007] regularization terms:

$$\begin{aligned} E_{\text{reg}} = w_{\text{L}} \sum_v \|L(\mathbf{X}_v) - L(\hat{\mathbf{X}}_v)\|^2 \\ + w_{\text{A}} \sum_v \sum_{i \in \mathcal{N}(v)} \|(\mathbf{X}_v - \mathbf{X}_i) - R_i(\hat{\mathbf{X}}_v - \hat{\mathbf{X}}_i)\|^2, \end{aligned} \tag{12}$$

where w_{L} and w_{A} are weights for the Laplacian and ARAP regularization terms (set to 100 and 0.5), respectively, and $\mathcal{N}(v)$ is a set of neighbors of vertex v.

Optimization. We solve Eq. (7) via the Gauss-Newton method. To efficiently optimize the large number of parameters, we use GPU-based implementation that computes the Jacobian matrix of each term in parallel and solve parameter updates with preconditioned conjugate gradient [Zollhöfer et al. 2014]. We run the optimization in a coarse-to-fine manner with 5 layers for effective convergence.

5.2 Initial Tracking

Our approach, as described thus far, assumes that the Geo-Tex VAE has already been trained. However, we must run an initial tracking pass to the first batch for initial training of the Geo-Tex VAE. Therefore, we first run sequential tracking based on the region-based blendshape model [Wu et al. 2016], followed by model-free optimization, for the first batch. We added three minor modifications to the existing method. First, we exclude the anatomical constraint that parameterizes rigid head pose based on a skull, because 3D scans from multi-view images resolves the geometric ambiguity. Second, we consider only the 3D feature distance term E_{feat} for this model-based tracking step. We found that thousands of SIFT features are detectable for high-resolution images, making the optimization over-constrained. Lastly, we start without any region blendshapes for the very first frame, and progressively update the region blendshapes

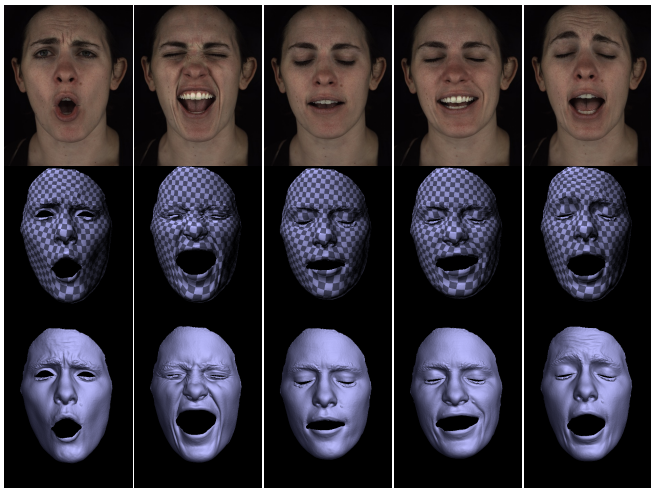


Fig. 7. Result of tracking for the A1-2 sequence with the Geo-Tex VAE model trained from the A1-1 sequence. Top: images, middle: rendering with a grid pattern, and bottom: rendering with Phong shading.

by selecting several meshes (10 at most) that are not expressed with each other based on shape similarity analysis.

The model-free optimization minimizes the objective defined in Eq. (7) in a sequential manner, and thus all variables denoted with $\hat{\cdot}$ in Eqs. (8), (10), (12) are replaced with those at frame $t-1$, and the variables without $\hat{\cdot}$ are replaced with those at frame t. Tracked meshes from this initial tracking step serve as the first batch for training the Geo-Tex VAE.

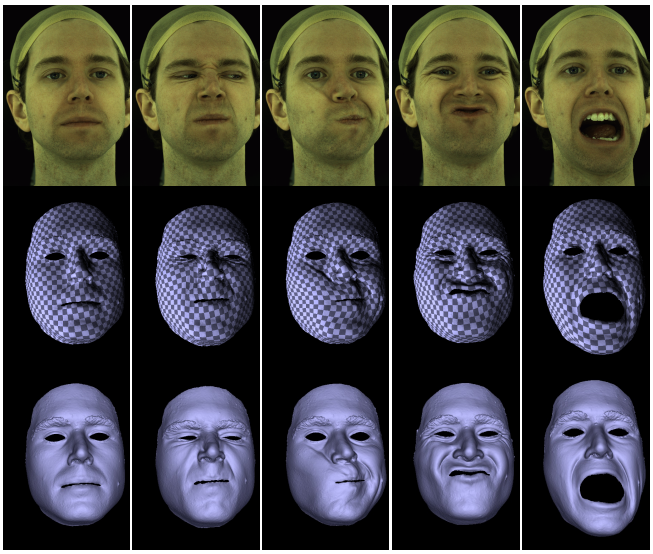
6 EXPERIMENTS

We conducted experiments with facial performance data captured with our multi-view camera setup. Our setup consists of 40 hardware-synchronized machine vision cameras capturing 2560×1920 pixels resolution at 30 fps with uniformly distributed LED lights around the capture area. Fig. 6 shows an example image set from this camera setup. We evaluate our framework with synthetic data and real data captured with this setup.

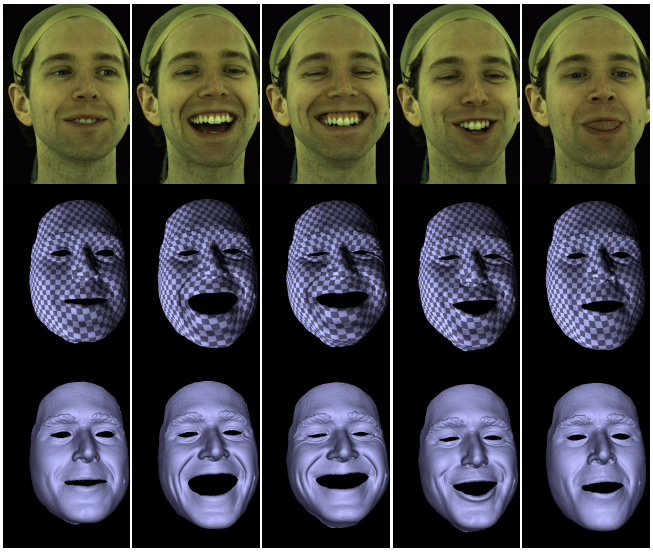
We implemented the modeling step with PyTorch and the tracking step with C++ and CUDA. For the modeling step, we set the learning rate to 10^{-3} and minibatch size to 16. For Adam optimization, the number of epochs for the first batch of our modeling step is 300, and the number of epochs for the following batches is 100, with 500 iterations of parameter update. All the template meshes we used have approximately 3×10^5 vertices.

6.1 Results with Our Datasets

We recorded facial performances of three actors, namely A1, A2 and A3. A1 performed some speech performance in an exaggerated tone for ~2 minutes (3600 frames), and we divided the sequence into two, namely A1-1 and A1-2. The A1-1 and A1-2 sequences consist of 2500 frames and 1100 frames, respectively. A2 performed range-of-motion (ROM) for ~53 seconds (1600 frames) and performed conversational speech performance for ~10 minutes, out of which we picked up an expressive sub-sequence for 28 seconds (840 frames), referred to as

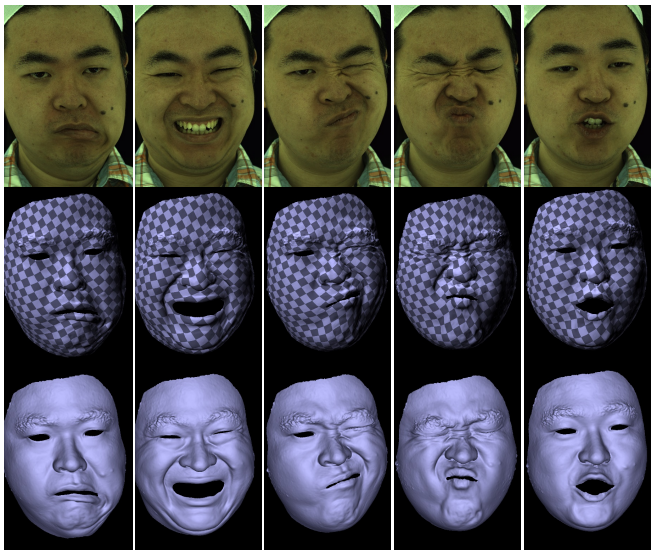


(a) Tracking result for A2-1 with the complete framework.

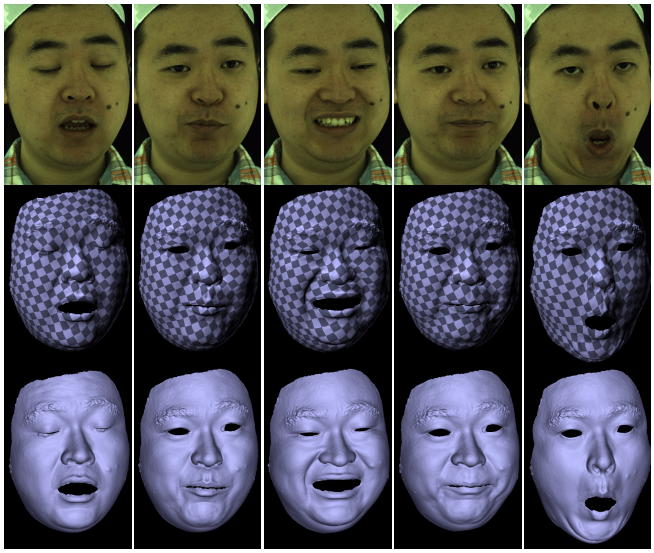


(b) Tracking result for A2-2 with the model trained from A2-1.

Fig. 8. Results of tracking for A2 Top: images, middle: rendering with a grid pattern, and bottom: rendering with Phong shading.



(a) Tracking result for A3-1 with the complete framework.



(b) Tracking result for A3-2 with the model trained from A3-1.

Fig. 9. Results of tracking for A3. Top: images, middle: rendering with a grid pattern, and bottom: rendering with Phong shading.

A2-1 and A2-2, respectively. A3 performed ROM for ~1.6 minutes (2900 frames) and performed conversational speech performance for approximately one minutes (1900 frames), referred to as A3-1 and A3-2, respectively.

Figs. 1, 8(a) and 9(a) show several frames of the tracking results with our complete learning and tracking framework for A1-1, A2-1, and A3-1, respectively. We render the results with checkerboard pattern texture to easily check drift. We also refer the readers to the supplemental video for the tracking results. Despite the fact that models in early stages are not expected to be trained well, and the fact that these sequences contain large and quick motion, the results show reasonable holistic facial motion and subtle deformation such as creases and wrinkles without postprocessing such as high-resolution detail transfer.

Figs. 7, 8(b) and 9(b) are the tracking results for A1-2, A2-2, and A3-2, respectively. The models used in these results were trained from the A*-1 sequences and fixed for these sequence. It is validated that the tracking results are qualitatively quite similar to the input images, despite the differences of the content between A*-1 and A*-2 and noise in images such as dust on the lens as shown in the supplemental video.

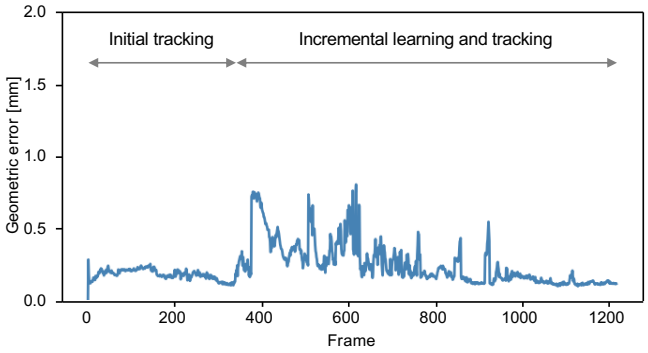


Fig. 10. Quantitative evaluation on tracking accuracy with synthetic data. The first 350 frames are tracked by initial tracking, and the rest is tracked with our incremental learning and tracking framework.

In all the results, minor jitters around eyelids and eyebrows are observed. These are caused by occlusion and appearance changes around eye crease, eyelashes and eyebrows. However, it is noted that it does not cause drift thanks to temporal independency in our tracking method. Please see Section 7 for more detailed discussions.

6.2 Quantitative Evaluation with Synthetic Data

We created a synthetic data from 1260 frames out of the tracking result of the A1-1 sequence for quantitative evaluation. We used the static texture map reconstructed from the first frame, and rendered the tracked mesh with the texture map under a constant illumination. Then, we ran the complete framework of our method for this sequence.

Fig. 10 shows root mean square errors (RMSE) over ~300k vertices along frames. The average RMSE over frames is 0.23 mm with standard deviation of 0.11 mm. As expected, the model produces higher errors at early stages of our pipeline due to an insufficient amount of data. These errors become smaller as the model is trained on more data. Besides, it is observed that errors are not accumulated or propagated to other frames because of the nature of individual frame registration.

To confirm the effectiveness of predction with our Geo-Tex VAE model, we compare the geometric errors of prediction and tracking from the Geo-Tex VAE model with those from a traditional PCA model with the same number of embedding dimensions as the VAE model. Fig. 12 shows the comparisons of the errors based on the synthetic data. Note that the first 350 frames are excluded in Fig. 12, because the same sequential tracking was performed for these frames to get the first batch of training data. The prediction and tracking errors up to around 750th frame from our method were similar to those from the PCA model, mainly due to the insufficient amount of training data for our model. However, with more data tracked and used for incremental model training (*i.e.*, after around 750th frame), our Geo-Tex VAE model significantly outperformed the PCA model for prediction. The smaller prediction errors from our model also achieved the consistently smaller surface tracking errors, as the surface tracking step requires predictions to be close to input images and geometry for the brightness constancy assumption in Eq. (8). In contrast, the PCA model has a lower capacity to encode complex deformations even with more training data and therefore produces larger prediction errors. These large prediction errors from the PCA model made the surface tracking more difficult to converge to good minima, leading to larger tracking errors compared with our model.

6.3 Evaluation on Geo-Tex VAE Training and Prediction

One of the key components in our framework is the Geo-Tex VAE architecture with the decomposition layer. To assess the performance of this architechure, we compared training errors with alternative models: one using only the four convolutional layers of Geo-Tex VAE (cov_O) and the other with 13 convolutional layers (cov_L), which follow typical convolutional VAE architecture. We used 3000 frames from the tracked meshes of the A1's sequence as a training dataset for each model and the remaining frames as a testing dataset. Fig. 11 shows how training errors of geometry and texture decrease for all the three models, and demonstrates that Geo-Tex VAE converges quickly and achieves the lowest training error. The average testing errors for geometry and texture after the training convergence are 0.23 and 0.047 for our model, 0.32 and 0.078 for cov_O, and 0.41 and 0.082 for cov_L, respectively, indicating that our model has the lowest testing error.

As it is challenging to obtain ground truth on real data, we compared only the prediction accuracy of our model with that of cov_L, cov_O and PCA on the real data same as above. In detail, we trained these four models with the same 3000 frame training dataset and computed predictions using the inverse rendering optimization Eq. (3) for each model on the testing dataset. Fig. 13 shows RMSE of vertex positions for each model over frames and validates that the Geo-Tex VAE-based prediction achieved the highest prediction accuracy.

6.4 Comparisons with Existing Methods

We applied our method to the data from Fyffe and colleagues [2017], and performed comparison with their result. We used the half of their data to run the complete learning and tracking framework, and used the other half, for which their tracking results are available, to run our tracking method with the Geo-Tex VAE model fixed. To assess the quality of the tracking results, we reconstructed a texture map for each frame based on the tracked meshes, normalized each texture map to remove the effect of brightness changes, and computed temporal standard deviation for intensity of each pixel in the UV space. Fig. 14 shows comparison of the standard deviation between our and their methods. Our method achieves substantially lower standard deviation, specifically around the forehead, cheeks and mouth, than the existing method. This can be also observed in the accompanying video where per-frame texture maps are visualized. While the texture maps from the existing method has noticeable sliding artifacts around the mouth, those from our method has better alignment in the UV space.

We also compare our method with the method of Beeler and colleagues [2011]. As the data they release is too short (~340 frames) to run our complete framework, we applied the initial tracking step

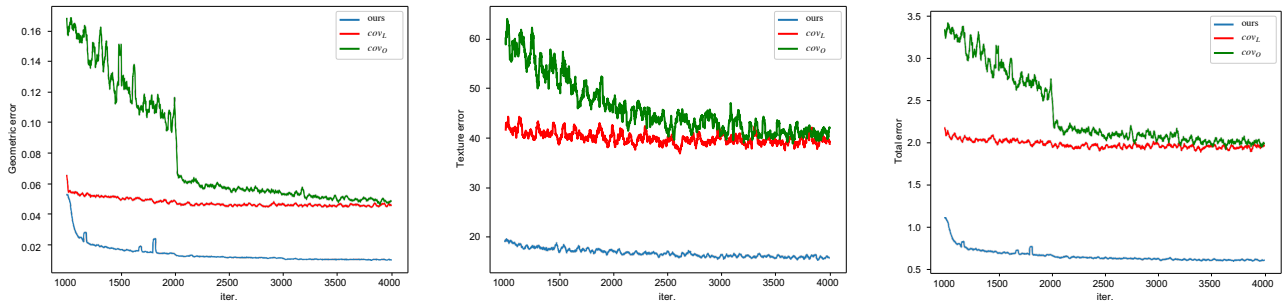


Fig. 11. Comparisons of training errors among Geo-Tex VAE, cov_L and cov_O. Our model achieves quickest convergence and lowest training errors in geometry, texture and total.

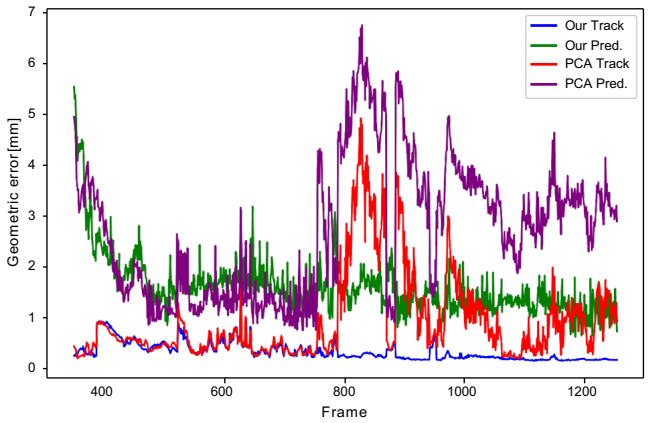


Fig. 12. Comparison of prediction and tracking errors of our method (green and blue, respectively) with those of the PCA model (purple and red, respectively) on synthetic data.

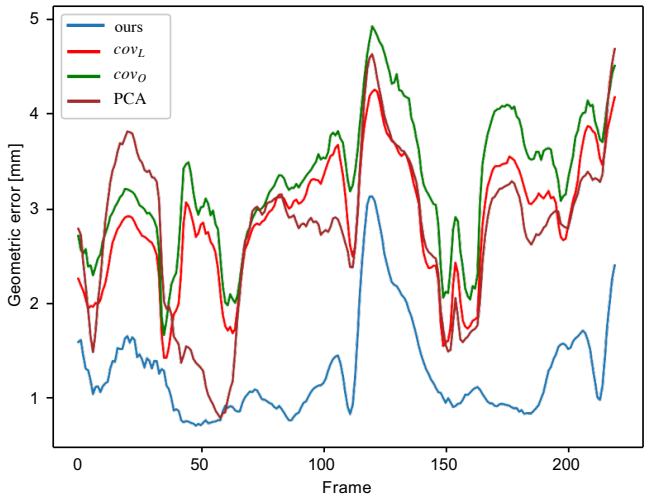


Fig. 13. Comparisons of prediction accuracy among our model, cov_L, cov_O and PCA. Our model achieves the highest prediction accuracy.

(Section 5.2) for this comparison. Fig. 15 shows the comparison of temporal standard deviation of intensities in the per-frame texture maps reconstructed with tracked meshes, Our method achieves lower standard deviation particularly around the mouth and the cheeks than the existing method, indicating that the initial tracking step can provide reasonable tracked meshes and corresponding texture maps as the first batch for training a Geo-Tex VAE model.

6.5 Performance Timings

We ran all the above experiments on an NVIDIA DGX compute server with eight Tesla V100 graphics cards. Each Tesla V100 card has 16 GB memory.

To train the rigid stabilization network and Geo-Tex VAE with the size of 1024 meshes and texture maps, it took ~20 seconds and ~1 minutes per epoch, respectively. The Geo-Tex VAE prediction took ~2 minutes, and the tracking using the prediction took ~1 minute. Note that the prediction and tracking steps are parallelizable, and we can process eight frames simultaneously with the DGX compute server. The major bottleneck in our framework is the model training step. However, as demonstrated in Section 6.1, we do not need the training step once the model is trained with facial performances containing various expressions. This indicates that, if we want to process many sequences, perhaps we will need to apply the complete framework for a few, very expressive sequences, and the rest will be processed only with the prediction and tracking steps, taking much less time than existing methods.

7 DISCUSSION

We presented a deep incremental learning framework that alternates between training a model with tracked meshes and texture maps and then initializing the tracking step using predicted meshes. To effectively model expressive facial performance, we introduced Geo-Tex VAE, which jointly models geometry and texture and consists of MLPs for global deformation and convolutional layers for local deformation. The tracking step directly optimizes mesh vertex positions and surface orientations by minimizing image, geometry and feature error together with conventional geometric regularization terms. We performed qualitative evaluation with real data captured with our setup, and quantitative evaluation with synthetic data that showed that our framework achieves sub-millimeter accuracy. We

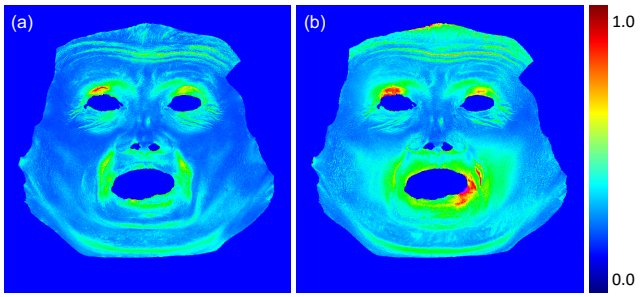


Fig. 14. Temporal standard deviation of texture maps reconstructed from the meshes by (a) our method and (b) Fyffe and colleagues' method [2017].

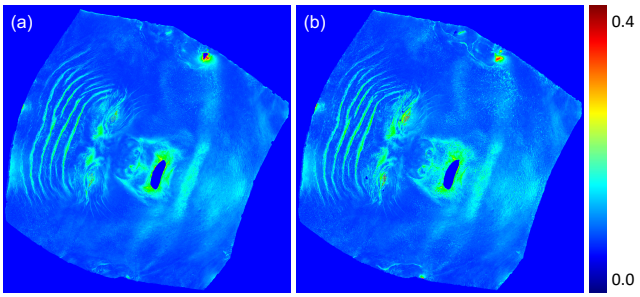


Fig. 15. Temporal standard deviation of texture maps reconstructed from the meshes by (a) our initial tracking method and (b) Beeler and colleagues' method [2011].

also did convergence analysis and assessed effectiveness of Geo-Tex VAE. Our method enables long-duration facial performance tracking as demonstrated in this paper, and perhaps introduces interesting research directions to statistically learn how people communicate each other through facial expressions.

While we showed high-fidelity results, there are a few limitations. We need a long, expressive sequence(s) such as range-of-motion to train Geo-Tex VAE that achieves sufficiently high performance. Because the major bottleneck in terms of runtime performance is the training step, as described in Section 6.5, one might consider this as a major limitation. With the data parallelism feature recently introduced in PyTorch, multiple GPUs can be utilized to accelerate the training. Our initial test shows that one epoch training time for the Geo-Tex VAE can be reduced from ~1 minutes to ~16 seconds with 8 Tesla V100 GPUs. Besides, as demonstrated in Section 6.1, we only need to perform the prediction and tracking steps once the model is trained with a sufficient amount of data. These steps are highly parallelizable, and accordingly the runtime performance becomes reasonably fast.

Another limitation is the limited amount of GPU memory. The current GPU memory size restricts us to use low-resolution UV maps (512×512 in our current implementation). The more memory on GPU, the higher UV map resolution we can utilize and possibly the better prediction from the Geo-Tex VAE.

Concave geometry under a chin and rolled-out lower lip causes significantly less visibility, and accordingly some sliding artifacts around those regions. More cameras observing from further lower viewpoints would help mitigate this issue.

One might be concerned about jitters around eyelids and sparse hair regions such as eyelashes and eyebrows. It is well known that eyelids and hair are difficult to track because of occlusion and appearance changes related to eye creases and hair reflectance property. Some tracking methods dedicated to eyelids [Bermano et al. 2015] and hair reconstruction [Beeler et al. 2012], or Geo-Tex VAE specialized for those regions, would be necessary to mitigate this issue. On the other hand, it is important to note that the capability of our method to register a template mesh to each frame independently successfully prevents such errors from being accumulated/propogated over other frames.

Currently, we do not consider a sophisticated appearance model such as albedo or subsurface scattering. We designed our framework for tracking purposes, so more sophisticated appearance models are out of the scope of our tracking framework: Rendering a mesh with baked-in texture map can synthesize an image pretty close to a real image captured under the same lighting condition and thus produce reasonable predictions. It would be an interesting research direction to estimate, for example, albedo maps in the tracking step, and model them together with tracked meshes in the modeling step. This would open up various applications such as modeling dynamic appearance changes, relighting and high-fidelity facial performance tracking in the wild.

ACKNOWLEDGMENTS

We would like to thank the actors for allowing us to use their data in Section 6.1, the authors of Beeler and colleagues [2011] and Fyffe and colleagues [2017] for providing their data for the comparisons in Section 6.4, and Colin Lea and all reviewers for their constructive discussions and feedback.

REFERENCES

K. S. Arun, T. S. Huang, and S. D. Blostein. 1987. Least-Squares Fitting of Two 3-D Point Sets. *IEEE TPAMI* 9, 5 (1987), 698–700.
Tali Basha, Yael Moses, and Nahum Kiryati. 2013. Multi-view Scene Flow Estimation: A View Centered Variational Approach. *IJCV* 101, 1 (2013), 6–21.
Thabo Beeler, Bernd Bickel, Gioacchino Noris, Paul Beardsley, Steve Marschner, Robert W. Sumner, and Markus Gross. 2012. Coupled 3D Reconstruction of Sparse Facial Hair and Skin. *ACM TOG* 31, 4 (2012).
Thabo Beeler and Derek Bradley. 2014. Rigid Stabilization of Facial Expressions. *ACM TOG* 33, 4 (2014).
Thabo Beeler, Fabian Hahn, Derek Bradley, Bernd Bickel, Paul Beardsley, Craig Gotsman, Robert W. Sumner, and Markus Gross. 2011. High-quality Passive Facial Performance Capture Using Anchor Frames. *ACM TOG* 30, 4 (2011).
Amit Bermano, Thabo Beeler, Yeara Kozlov, Derek Bradley, Bernd Bickel, and Markus Gross. 2015. Detailed Spatio-temporal Reconstruction of Eyelids. *ACM TOG* 34, 4 (2015).
Volker Blanz and Thomas Vetter. 1999. A Morphable Model for the Synthesis of 3D Faces. In *Proc. SIGGRAPH*.
James Booth, Epameinondas Antonakos, Stylianos Ploumpis, George Trigeorgis, Yannis Panagakis, and Stefanos Zafeiriou. 2017. 3D Face Morphable Models "In-The-Wild". In *Proc. CVPR*.
Sofien Bouaziz, Yangang Wang, and Mark Pauly. 2013. Online Modeling for Realtime Facial Animation. *ACM TOG* 32, 4 (2013).
Derek Bradley, Wolfgang Heidrich, Tiberiu Popa, and Alla Sheffer. 2010. High Resolution Passive Facial Performance Capture. *ACM TOG* 29, 4 (2010).
Chen Cao, Derek Bradley, Kun Zhou, and Thabo Beeler. 2015. Real-time High-fidelity Facial Performance Capture. *ACM TOG* 34, 4 (2015).
Chen Cao, Qiming Hou, and Kun Zhou. 2014a. Displaced Dynamic Expression Regression for Real-time Facial Tracking and Animation. *ACM TOG* 33, 4 (2014).

Chen Cao, Yanlin Weng, Stephen Lin, and Kun Zhou. 2013. 3D Shape Regression for Real-time Facial Animation. *ACM TOG* 32, 4 (2013).
Xudong Cao, Yichen Wei, Fang Wen, and Jian Sun. 2014b. Face alignment by explicit shape regression. *IJCV* (2014).
G.D. Evangelidis and E.Z. Psarakis. 2008. Parametric Image Alignment by Using Enhanced Correlation Coefficient Maximization. *IEEE TPAMI* 30, 10 (2008).
Graham Fyffe, Andrew Jones, Oleg Alexander, Ryosuke Ichikari, and Paul Debevec. 2014. Driving High-Resolution Facial Scans with Video Performance Capture. *ACM TOG* 34, 1 (2014).
G. Fyffe, K. Nagano, L. Huynh, S. Saito, J. Busch, A. Jones, H. Li, and P. Debevec. 2017. Multi-View Stereo on Consistent Face Topology. *Computer Graphics Forum* (2017).
Pablo Garrido, Levi Valgaerts, Chenglei Wu, and Christian Theobalt. 2013. Reconstructing Detailed Dynamic Face Geometry from Monocular Video. *ACM TOG* 32, 6 (2013).
Pablo Garrido, Michael Zollhöfer, Dan Casas, Levi Valgaerts, Kiran Varanasi, Patrick Pérez, and Christian Theobalt. 2016. Reconstruction of Personalized 3D Face Rigs from Monocular Video. *ACM TOG* 35, 3 (2016).
R. I. Hartley and A. Zisserman. 2004. *Multiple View Geometry in Computer Vision* (second ed.). Cambridge University Press.
Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *Proc. CVPR*.
Haoda Huang, Jinxiang Chai, Xin Tong, and Hsiang-Tao Wu. 2011. Leveraging Motion Capture and 3D Scanning for High-fidelity Facial Performance Acquisition. *ACM TOG* 30, 4 (2011).
Loc Huynh, Weikai Chen, Shunsuke Saito, Jun Xing, Koki Nagano, Andrew Jones, Paul Debevec, and Hao Li. 2018. Mesoscopic Facial Geometry Inference Using Deep Neural Networks. In *Proc. CVPR*.
Alexandru Eugen Ichim, Sofien Bouaziz, and Mark Pauly. 2015. Dynamic 3D Avatar Creation from Hand-held Video Input. *ACM TOG* 34, 4 (2015).
Tero Karras, Timo Aila, Samuli Laine, Antti Herva, and Jaakko Lehtinen. 2017. Audio-Driven Facial Animation by Joint End-to-End Learning of Pose and Emotion. *ACM TOG* 36, 4 (2017).
D. P. Kingma and J. Ba. 2015. Adam: A Method for Stochastic Optimization. In *Proc. ICLR*.
D. P. Kingma and M. Welling. 2014. Auto-Encoding Variational Bayes. In *Proc. ICLR*.
M. Klaudiny and A. Hilton. 2012. High-Detail 3D Capture and Non-sequential Alignment of Facial Performance. In *Proc. 3DIMPVT*. 17–24.
John P. Lewis, Ken ichi Anjyo, Taehyun Rhee, Mengjie Zhang, Frédéric H. Pighin, and Zhigang Deng. 2014. Practice and Theory of Blendshape Facial Models. In *Proc. Eurographics State of The Art Report*.
Hao Li, Thibaut Weise, and Mark Pauly. 2010. Example-Based Facial Rigging. *ACM TOG* 29, 3 (2010).
Hao Li, Jihun Yu, Yuting Ye, and Chris Bregler. 2013. Realtime Facial Animation with On-the-fly Correctives. *ACM TOG* 32, 4 (2013).
Tianye Li, Timo Bolkart, Michael. J. Black, Hao Li, and Javier Romero. 2017. Learning a model of facial shape and expression from 4D scans. *ACM TOG* 36, 6 (2017).
David G. Lowe. 2004. Distinctive image features from scale-invariant keypoints. *IJCV* 60, 2 (2004).
Iain Matthews and Simon Baker. 2004. Active Appearance Models Revisited. *IJCV* 60, 2 (2004).
Patrick Pérez, Michel Gangnet, and Andrew Blake. 2003. Poisson Image Editing. *ACM TOG* 22, 3 (2003).
K. Lasinger S. Galliani and K. Schindler. 2015. Massively Parallel Multiview Stereopsis by Surface Normal Diffusion. In *Proc. ICCV*.
Shunsuke Saito, Lingyu Wei, Liwen Hu, Koki Nagano, and Hao Li. 2017. Photorealistic Facial Texture Inference Using Deep Neural Networks. In *Proc. CVPR*.
Jason M. Saragih, Simon Lucey, and Jeffrey F. Cohn. 2011. Deformable Model Fitting by Regularized Landmark Mean-Shift. *IJCV* 91, 2 (2011).
Fuhao Shi, Hsiang-Tao Wu, Xin Tong, and Jinxiang Chai. 2014. Automatic Acquisition of High-fidelity Facial Performances Using Monocular Videos. *ACM TOG* 33, 6 (2014).
Olga Sorkine and Marc Alexa. 2007. As-rigid-as-possible Surface Modeling. In *Proc. SGP*.
Supasorn Suwajanakorn, Ira Kemelmacher-Shlizerman, and Steven M. Seitz. 2014. Total Moving Face Reconstruction. In *Proc. ECCV*. 796–812.
Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. 2014. DeepFace: Closing the Gap to Human-Level Performance in Face Verification. In *Proc. CVPR*.
Sarah Taylor, Taehwan Kim, Yisong Yue, Moshe Mahler, James Krahe, Anastasio Garcia Rodriguez, Jessica Hodgins, and Iain Matthews. 2017. A Deep Learning Approach for Generalized Speech Animation. *ACM TOG* 36, 4 (2017).
J. Rafael Tena, Fernando De la Torre, and Iain Matthews. 2011. Interactive Region-Based Linear 3D Face Models. *ACM TOG* 30, 4 (2011).
A. Tewari, M. Zollhöfer, P. Garrido, F. Bernard, H. Kim, P. Pérez, and C. Theobalt. 2018. Self-supervised Multi-level Face Model Learning for Monocular Reconstruction at over 250 Hz. In *Proc. CVPR*.
Ayush Tewari, Michael Zollöfer, Hyeongwoo Kim, Pablo Garrido, Florian Bernard, Patrick Perez, and Christian Theobalt. 2017. MoFA: Model-based Deep Convolutional Face Autoencoder for Unsupervised Monocular Reconstruction. In *Proc. ICCV*.
Justus Thies, Michael Zollhöfer, Matthias Nießner, Levi Valgaerts, Marc Stamminger, and Christian Theobalt. 2015. Real-time Expression Transfer for Facial Reenactment. *ACM TOG* 34, 6 (2015).
Justus Thies, Michael Zollhöfer, Marc Stamminger, Christian Theobalt, and Matthias Nießner. 2016. Face2Face: Real-Time Face Capture and Reenactment of RGB Videos. In *Proc. CVPR*.
Luan Tran and Xiaoming Liu. 2018. Nonlinear 3D Face Morphable Model. In *Proc. CVPR*.
L. Valgaerts, A. Bruhn, H. Zimmer, J. Weickert, C. Stoll, and C. Theobalt. 2010. Joint estimation of motion, structure and geometry from stereo sequences. In *Proc. ECCV*.
Levi Valgaerts, Chenglei Wu, Andrés Bruhn, Hans-Peter Seidel, and Christian Theobalt. 2012. Lightweight Binocular Facial Performance Capture under Uncontrolled Lighting. *ACM TOG* 31, 6 (2012).
S. Vedula, P. Rander, R. Collins, and T. Kanade. 2005. Three-dimensional scene flow. *IEEE TPAMI* 27, 3 (2005).
D. Vlasic, M. Brand, H. Pfister, and J. Popovic. 2005. Face transfer with multilinear models. *ACM TOG* 24, 3 (2005).
Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. 2016. Convolutional Pose Machines. In *Proc. CVPR*.
Thibaut Weise, Sofien Bouaziz, Hao Li, and Mark Pauly. 2011. Realtime performance-based facial animation. *ACM TOG* (2011).
Thibaut Weise, Hao Li, Luc Van Gool, and Mark Pauly. 2009. Face/Off: Live Facial Puppetry. In *Proc. SCA*.
Quan Wen, Feng Xu, Ming Lu, and Yong Jun-Hai. 2017. Real-time 3D Eyelids Tracking from Semantic Edges. *ACM TOG* (2017).
Chenglei Wu, Derek Bradley, Markus Gross, and Thabo Beeler. 2016. An Anatomically-constrained Local Deformation Model for Monocular Face Capture. *ACM TOG* 35, 4 (2016).
Michael Zollhöfer, Matthias Nießner, Shahram Izadi, Christoph Rehmann, Christopher Zach, Matthew Fisher, Chenglei Wu, Andrew Fitzgibbon, Charles Loop, Christian Theobalt, and Marc Stamminger. 2014. Real-time Non-rigid Reconstruction Using an RGB-D Camera. *ACM TOG* 33, 4 (2014).