Self-Supervised Adaptation of High-Fidelity Face Models for Monocular Performance Tracking

Jae Shin Yoon[†] Takaaki Shiratori[‡] Shoou-I Yu[‡] Hyun Soo Park[†]
[†]University of Minnesota [‡]Facebook Reality Labs
{jsyoon, hspark}@umn.edu {tshiratori, shoou-i.yu}@fb.com

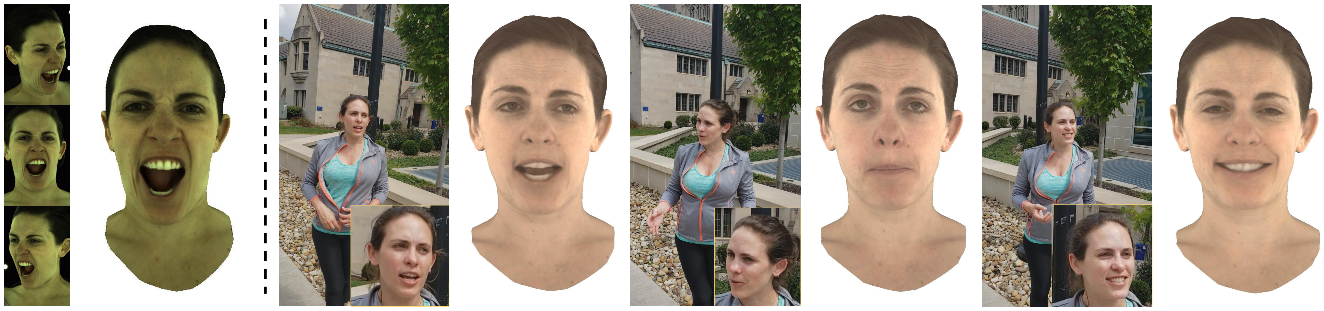

Figure 1: Results of high-fidelity 3D facial performance tracking from our method, which automatically adapts a high-quality face model [13] captured in a controlled lab environment (left) to in-the-wild imagery (right) through our proposed self-supervised domain adaptation method. Note the fine details we are able to recover from cellphone quality video.

Abstract

Improvements in data-capture and face modeling techniques have enabled us to create high-fidelity realistic face models. However, driving these realistic face models requires special input data, e.g. *3D meshes and unwrapped textures. Also, these face models expect clean input data taken under controlled lab environments, which is very different from data collected in the wild. All these constraints make it challenging to use the high-fidelity models in tracking for commodity cameras. In this paper, we propose a self-supervised domain adaptation approach to enable the animation of high-fidelity face models from a commodity camera. Our approach first circumvents the requirement for special input data by training a new network that can directly drive a face model just from a single 2D image. Then, we overcome the domain mismatch between lab and uncontrolled environments by performing self-supervised domain adaptation based on "consecutive frame texture consistency" based on the assumption that the appearance of the face is consistent over consecutive frames, avoiding the necessity of modeling the new environment such as lighting or background. Experiments show that we are able to drive a high-fidelity face model to perform complex facial motion from a cellphone camera without requiring any labeled data from the new domain.*

1. Introduction

High-fidelity face models enable the building of realistic avatars, which play a key role in communicating ideas, thoughts and emotions. Thanks to the uprising of data-driven approaches, highly realistic and detailed face models can be created with active appearance models (AAMs) [6, 5], 3D morphable models (3DMMs) [1], or deep appearance models (DAMs) [13]. These data-driven approaches jointly model facial geometry and appearance, thus empowering the model to learn the correlation between the two and synthesize high-quality facial images. Particularly, the recently proposed DAMs can model and generate realistic animation and view-dependent textures with pore-level details by leveraging the high capacity of deep neural networks.

Unfortunately, barriers exist when applying these high-quality models to monocular in-the-wild imagery due to *modality mismatch* and *domain mismatch*. Modality mismatch refers to the fact that high-fidelity face modeling and tracking requires specialized input data, (*e.g.* DAMs require tracked 3D meshes and unwrapped textures) which is not easily accessible on consumer-grade mobile capture devices. Domain mismatch refers to the fact that the visual statistics of in-the-wild imagery are considerably different from that of a controlled lab environment used to build the high-fidelity face model. In-the-wild imagery includes various background clutter, low resolution, and complex ambient lighting. Such domain gap breaks the correlation between appearance and geometry learned by the data-driven model and the model may no longer work well in the new domain. The existence of these two challenges greatly inhibits the wide-spread use of the high-fidelity face models.

In this paper, we present a method to perform high-fidelity face tracking for monocular in-the-wild imagery based on DAMs face model learned from lab-controlled imagery. Our method bridges the controlled lab domain and in-the-wild domain such that we can perform high-fidelity face tracking with DAM face models on in-the-wild video camera sequences. To tackle the modality mismatch, we train I2ZNet, a deep neural network that takes a monocular image as input and directly regresses to the intermediate representation of the DAM, thus circumventing the need for 3D meshes and unwrapped textures required in DAMs. As I2ZNet relies on data captured in a lab environment and cannot handle the domain mismatch, we present a self-supervised domain adaptation technique that can adapt I2ZNet to new environments without requiring any labeled data from the new domain. Our approach leverages the assumption that the textures (appearance) of a face between consecutive frames should be consistent and incorporates this source of supervision to adapt the domain of I2ZNet such that final tracking results preserve consistent texture over consecutive frames on target-domain imagery, as shown in Figure 1. The resulting face tracker outperforms state-of-the-art face tracking methods in terms of geometric accuracy, temporal stability, and visual plausibility.

The key strength of this approach is that we do not make any other assumptions on the scene or lighting of in-the-wild imagery, enabling our method to be applicable to a wide variety of scenes. Furthermore, our method computes consistency for all visible portions of the texture, thus providing significantly more supervision and useful gradients than per-vertex based methods [22, 8]. Finally, we emphasize that the consecutive frame texture consistency assumption is not simply a regularizer to avoid overfitting. This assumption provides an additional source of supervision which enables our model to adapt to new environments and achieve considerable improvement of accuracy and stability.

In summary, the contributions of this paper are as follows:

1. I2ZNet, a deep neural network that can directly predict the intermediate representation of a DAM from a single image.
2. A self-supervised domain adaptation method based on consecutive frame texture consistency to enhance face tracking. No labeled data is required for images from the target domain.
3. High-fidelity 3D face tracking on in-the-wild videos captured with a commodity camera.

2. Related Work

Humans are evolved to decode, understand and convey non-verbal information from facial motion, *e.g.*, a subtle unnatural eye blink, symmetry, and reciprocal response can be easily detected. Therefore, the realistic rendering of facial motion is key to enable telepresence technology [13]. This paper lies in the intersection between high fidelity face modeling and 3D face reconstruction from a monocular camera, which will be briefly reviewed here.

3D Face Modeling Faces have underlying spatial structural patterns where low dimensional embedding can efficiently and compactly represent diverse facial configurations, shapes, and textures. Active Shape Models (ASMs) [6] have shown strong expressibility and flexibility to describe a variety of facial configurations by leveraging a set of facial landmarks. However, the nature of the sparse landmark dependency limits the reconstruction accuracy that is fundamentally bounded by the landmark localization. AAMs [5] address the limitation by exploiting the photometric measure using both shape and texture, resulting in compelling face tracking. Individual faces are combined into a single 3DMM [1] by computing dense correspondences based on optical flow in conjunction with the shape and texture priors in a linear subspace. Large-scale face scans (more than 10,000 people) from diverse population enables modeling of accurate distributions of faces [3, 2]. With the aid of multi-camera systems and deep neural networks, the limitation of the linear models can be overcome using DAMs [13] that predicts high quality geometry and texture. Its latent representation is learned by a conditional variational autoencoder [12] that encodes view-dependent appearance from different viewpoints. Our approach eliminates the multi-camera requirement of the DAMs by adapting the networks to a video from a monocular camera.

Single View Face Reconstruction The main benefit of the compact representation of 3D face modeling is that it allows estimating the face shape, appearance, and illumination parameters from a single view image. For instance, the latent representation of the 3DMMs can be recovered by jointly optimizing pixel intensity, edges and illumination (approximated by spherical harmonics) [17]. The recovered 3DMMs can be further refined to fit to a target face using a collection of photos [18] or depth based camera [4]. [23] leveraged expert designed rendering layers which model face shape, expression, and illumination and utilized inverse rendering to estimate a set of compact parameters which renders a face that best fits the input. This is often an simplification and cannot model all situations. In contrast, our method does not make any explicit assumptions on the lighting of the scene, and thus achieves more flexibility to different environments.

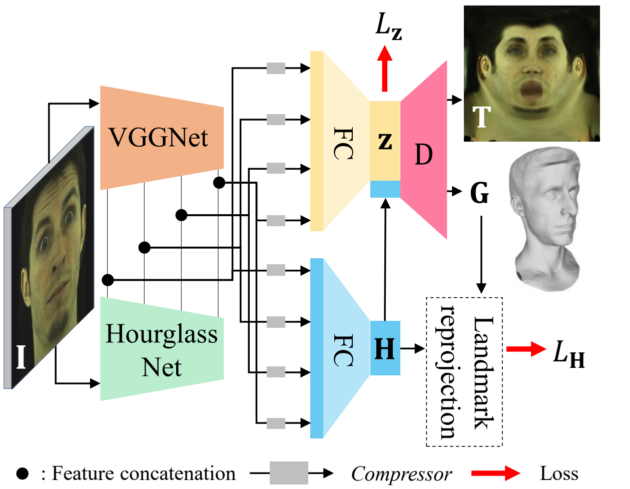

Figure 2: Illustration of the I2ZNet architecture. I2ZNet extracts the domain-invariant perceptual features and facial image features using the pre-trained VGGNet [20] and HourglassNet [14], respectively, from an input image $\mathbf{I}$. The combined multiple depth-level features are then regressed to the latent code $\mathbf{z}$ of the pre-trained DAMs (D) [13] and the head pose $\mathbf{H}$ via fully connected layers. I2ZNet is trained with the losses defined for $\mathbf{z}$ and $\mathbf{H}$, namely $L_{\mathbf{z}}$ and $L_{\mathbf{H}}$, as well as the view consistency loss in Eq. (4).

Other methods include [10, 29], which used cascaded CNNs which densely align the 3DMM with a 2D face in an iterative way based on facial landmarks. The geometry of a 3D face is regressed in a coarse-to-fine manner [16], and asymmetric loss enforces the network to regress the identity consistent 3D face [24]. [22] utilizes jointly learned geometry and reflectance correctives to fit in-the-wild faces. [9] trained UV regression maps which jointly align with the 3DMM to directly reconstruct a 3D face.

Tackling Domain Mismatch A key challenge is the often-times significant gap between the distribution of training and testing data. To this end, [15, 24] utilized synthetic data to boost 3D face reconstruction performance. A challenge here is to generate synthetic data that is representative of the testing distribution. [8] utilized domain invariant motion cues to perform unsupervised domain adaptation for facial landmark tracking. While their method was tested on *sparse* landmarks and benefited from a limited source of supervision, our method performs *dense* per-pixel matching of textures, providing more supervision for domain adaptation.

3. Methodology

When applying existing face models such as AAMs and DAMs to monocular video recordings, we face two challenges: modality mismatch and domain mismatch. Modality mismatch occurs when the existing face model requires input data to be represented in a face centric representation such as 3D meshes with unwrapped texture in a pre-defined topology. This representation does not comply with an image centric representation, thus preventing us from using these face models. Domain mismatch occurs when the visual statistics of in-the-wild images are different from that of the scenes used to construct the models. In the following sections, we first present I2ZNet for the modality mismatch, and then describe how to adapt I2ZNet in a self-supervised fashion for the domain mismatch.

3.1. Handling Modality Mismatch

Many face models including DAMs can be viewed as an encoder and decoder framework. The encoder E_X takes an input $X = (\mathbf{G}, \mathbf{T})$, which corresponds to the geometry and unwrapped texture, respectively. $\mathbf{G} \in \mathbb{R}^{G\times 3}$ represents the 3D locations of G vertices which form a 3D mesh of the face. Note that rigid head motion has already been removed from the vertex locations, *i.e.* $\mathbf{G}$ represents only local deformations of the face. The unwrapped texture $\mathbf{T} \in \mathbb{R}^{T\times T\times 3}$ is a 2D image that represents the appearance at different locations on $\mathbf{G}$ in the UV space. The output of E_X is the intermediate code $\mathbf{z}$. The decoder D then takes $\mathbf{z}$ and computes a reconstructed output $\widetilde{X} = \mathrm{D}(\mathbf{z}) = \mathrm{D}(\mathrm{E}_X(X))$. The encoder and decoder are learned by minimizing the difference between X and $\widetilde{X}$ for a large number of training samples.

The challenge is that $X = (\mathbf{G}, \mathbf{T})$, *i.e.*, the 3D geometry and unwrapped texture, is not readily available in a monocular image $\mathbf{I}$. Therefore, we learn a separate deep encoder called *I2ZNet* (Image-to-$\mathbf{z}$ network): $(\mathbf{z}, \mathbf{H}) \leftarrow \mathrm{E}_{\mathbf{I}}(\mathbf{I})$, which takes a monocular image $\mathbf{I}$ as input and directly outputs $\mathbf{z}$ and the rigid head pose $\mathbf{H}$. I2ZNet first extracts the domain independent two-stream features using the pre-trained VGGNet [20] and HourglassNet [14], which provides perceptual information and facial landmarks, respectively. The multiple depth-level two-stream features are combined with skip connections, and are regressed respectively to the intermediate representation $\mathbf{z} \in \mathbb{R}^{128}$ and the head pose $\mathbf{H} \in \mathbb{R}^{6}$ using several fully connected layers [26]. This architecture allows to directly predicts the parameters $(\mathbf{z}, \mathbf{H})$ based on the category-level semantic information from the deep layers and local geometric/appearance details from the shallow layers at the same time. $\mathbf{z}$ can be given to the existing decoder D to decode the 3D mesh and texture, while $\mathbf{H}$ allows to reproject the decoded 3D mesh onto the 2D image. Figure 2 illustrates the overall architecture of I2ZNet, and more details are described in the supplementary manuscript.

$\mathrm{E}_{\mathbf{I}}$ is trained in a supervised way with multiview image sequences used for training E_X and D of DAMs. The by-product of learning E_X and D are the latent code $\mathbf{z}_{gt}$ and the head pose $\mathbf{H}_{gt}$ at each time. As a result of DAM training, we acquire as many tuples of $\{\mathbf{I}_v, \mathbf{z}_{gt}, \mathbf{H}_{gt}\}$ as the camera views $\{v\}$ at every time t as training data for $\mathrm{E}_\mathbf{I}$.

The total loss to train $\mathrm{E}_\mathbf{I}$ is defined as

$$L_{\mathrm{E_I}} = \lambda_{\mathrm{z}} L_{\mathrm{z}} + \lambda_{\mathrm{H}} L_{\mathrm{H}} + \lambda_{\mathrm{view}} L_{\mathrm{view}}, \tag{1}$$

where L_{z} and L_{H} are the losses for $\mathbf{z}$ and $\mathbf{H}$, respectively, and L_{view} is the view-consistency loss. λ_{z}, λ_{H} and λ_{view} are weights for L_{z}, L_{H} and L_{view}, respectively.

L_{z} is the direct supervision term for $\mathbf{z}$ defined as

$$L_{\mathrm{z}} = \sum_{v,t} \left\| \mathbf{z}_{\mathbf{I}_v^t} - \mathbf{z}_{gt}^t \right\|_2^2, \tag{2}$$

where $\mathbf{z}_\mathbf{I}$ is a DAM latent code regressed from $\mathbf{I}$ via $\mathrm{E}_\mathbf{I}$.

Inspired by [22, 11], we formulate $L_\mathbf{H}$ as the reprojection error of the 3D landmarks predicted via $\mathrm{E}_\mathbf{I}$ w.r.t. the 2D ground-truth landmarks $\mathbf{K}_{gt} \in \mathbb{R}^{K\times 2}$ for the head pose prediction:

$$L_{\mathrm{H}} = \frac{1}{K} \sum_{k,v,t} \left\| \mathbf{\Pi} \mathbf{H}_{\mathbf{I}_v^t} \mathbf{K}^k (\mathbf{G}_{\mathbf{I}_v^t}) - \mathbf{K}_{gt}^k \right\|_2^2, \tag{3}$$

where K is the number of landmarks, $\mathbf{\Pi} = [1\ 0\ 0; 0\ 1\ 0]$ is a weak perspective projection matrix, and $\mathbf{H}_\mathbf{I}$ is the head pose regressed from $\mathbf{I}$ via I2ZNet. $\mathbf{G}_\mathbf{I}$ is the set of vertex locations decoded from $\mathbf{z}_\mathbf{I}$ via D, and $\mathbf{K}^k(\cdot)$ computes the 3D location of k-th landmark from $\mathbf{G}_\mathbf{I}$.

Because the training image data is captured with synchronized cameras, we want to ensure that the regressed $\mathbf{z}$ is the same for images from different views captured at the same time. Therefore, we incorporate the view-consistency loss L_{view}, defined as

$$L_{\mathrm{view}} = \sum_{v,w,t} \left\| \mathbf{z}_{\mathbf{I}_v^t} - \mathbf{z}_{\mathbf{I}_w^t} \right\|_2^2. \tag{4}$$

We randomly select two views at every training iteration.

3.2. Handling Domain Mismatch

To handle the domain mismatch, we adapt I2ZNet to a new domain using a set of unlabeled images in a self-supervised manner. The overview of the proposed domain adaptation is illustrated in Figure 3. Given a monocular video, we refine the encoder $\mathrm{E}_\mathbf{I}$ by minimizing the domain adaptation loss L_{DA} (Eq. (5)), which consists of (1) consecutive frame texture consistency L_{CFTC}, (2) model-to-observation texture consistency L_{MOTC}, and (3) facial landmark reprojection consistency L_{FLRC}:

$$L_{\mathrm{DA}}^t = \lambda_{\mathrm{CFTC}} L_{\mathrm{CFTC}}^t + \lambda_{\mathrm{MOTC}} L_{\mathrm{MOTC}}^t + \lambda_{\mathrm{FLRC}} L_{\mathrm{FLRC}}^t, \tag{5}$$

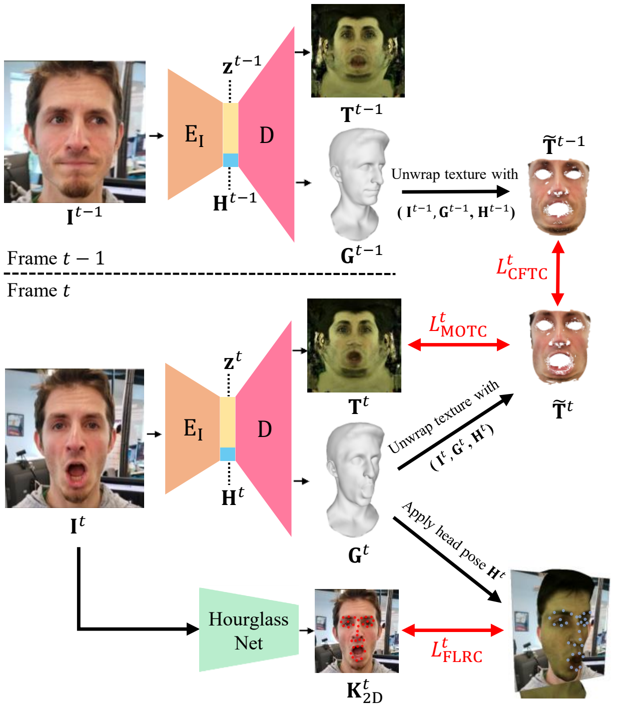

Figure 3: Overview of our self-supervised domain adaptation process. Given two consecutive frames ($\mathbf{I}^{t-1}$, $\mathbf{I}^t$), we run $\mathrm{E}_\mathbf{I}$ followed by D to acquire the geometry ($\mathbf{G}^{t-1}$, $\mathbf{G}^t$), textures ($\mathbf{T}^{t-1}$, $\mathbf{T}^t$) and head poses ($\mathbf{H}^{t-1}$, $\mathbf{H}^t$). Then, $\mathbf{I}$, $\mathbf{G}$ and $\mathbf{H}$ are used to compute observed textures ($\widetilde{\mathbf{T}}^{t-1}$, $\widetilde{\mathbf{T}}^t$). These enable us to compute L_{CFTC} and L_{MOTC}. For frame t, we run a hourglass facial landmark detector to get 2D landmark locations $\mathbf{K}_{2\mathrm{D}}^t$, which is then used to compute L_{FLRC}. These losses can back-propagate gradients back to $\mathrm{E}_\mathbf{I}$ to perform self-supervised domain adaptation.

where λ_{CFTC}, λ_{MOTC} and λ_{FLRC} correspond to the weights for each loss term. L_{CFTC} is our key contribution. It adapts $\mathrm{E}_\mathbf{I}$ such that textures computed from predicted geometry are temporally coherent. L_{MOTC} enforces the consistent color of DAM generated texture with the observed texture via pixel-wise matching. L_{FLRC} anchors the tracked 3D face by minimizing the reprojection error of the 3D model landmarks with the detected facial landmarks.

3.2.1 Consecutive Frame Texture Consistency

Inspired by the brightness constancy assumption employed in many optical flow algorithms, we can reasonably assume that 3D face tracking for two consecutive frames is accurate only if unwrapped textures for the two frames are nearly identical. Inversely, if we see large changes in unwrapped texture across consecutive frames, it is highly likely due to inaccurate 3D geometry predictions. We make the assumption that environmental lighting and the appearance of the face does not change significantly between consecutive

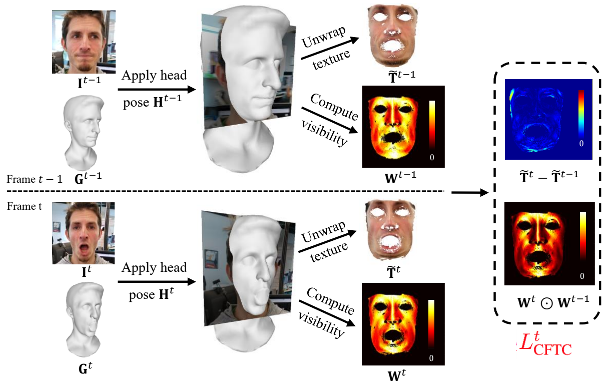
$\mathbf{I}^{t-1}$
Apply head pose $\mathbf{H}^{t-1}$
Frame $t-1$ $\mathbf{G}^{t-1}$
Unwrap texture
$\widetilde{\mathbf{T}}^{t-1}$
Compute visibility
$\mathbf{W}^{t-1}$
0
$\widetilde{\mathbf{T}}^{t} - \widetilde{\mathbf{T}}^{t-1}$
Frame t
$\mathbf{I}^{t}$
Apply head pose $\mathbf{H}^{t}$
$\mathbf{G}^{t}$
Unwrap texture
$\widetilde{\mathbf{T}}^{t}$
Compute visibility
$\mathbf{W}^{t}$
$\mathbf{W}^{t} \odot \mathbf{W}^{t-1}$
L^{t}_{CFTC}

Figure 4: Illustration of how to compute L_{CFTC}.

frames, which is satisfied in most scenarios. Otherwise, we do not make any assumptions on the lighting environment of a new scene, which makes our method more generalizable than existing methods which, for example, approximates lighting with spherical harmonics [22].

The consecutive frame texture consistency loss L_{CFTC} is:

$$L^{t}_{\text{CFTC}} = \frac{1}{W^{t,t-1}} \left\| (\mathbf{W}^{t} \odot \mathbf{W}^{t-1}) \odot (\widetilde{\mathbf{T}}^{t} - \widetilde{\mathbf{T}}^{t-1}) \right\|_{F}^{2}, \tag{6}$$

where $\mathbf{W} \in \mathbb{R}^{T \times T}$ is a confidence matrix, $\widetilde{\mathbf{T}}$ is a texture obtained by projecting $\mathbf{G}_{\mathbf{I}}$ onto $\mathbf{I}$ with $\mathbf{H}_{\mathbf{I}}$, and $\odot$ is element-wise multiplication. We use the cosine of incident angle of the ray from the camera center to each texel as the confidence to reduce the effect of texture distortion caused at grazing angles. Elements smaller than a threshold in $\mathbf{W}^{t} \odot \mathbf{W}^{t-1}$ are set to 0. $W^{t,t-1}$ is the number of non-zero elements in $\mathbf{W}^{t} \odot \mathbf{W}^{t-1}$. Figure 4 shows example confidence matrices and textures as well as computation of L_{CFTC}.

$\widetilde{\mathbf{T}}$ is obtained by projecting the 3D location of each texel decoded from $\mathbf{z}$ to an observed image $\mathbf{I}$.

$$\widetilde{\mathbf{T}}_{ij} = \mathbf{I}(\mathbf{\Pi}\mathbf{H}_{\mathbf{I}}\mathbf{X}(\mathbf{G}_{\mathbf{I}}, i, j)), \tag{7}$$

where (i, j) is texel coordinates. Unlike existing methods that compute per-vertex texture loss [22, 8], L_{CFTC} considers all visible texels, providing significantly richer source of supervision and gradients than per-vertex-based methods. The aforementioned steps are all differentiable, thus the entire model can be updated in an end-to-end fashion.

3.2.2 Model-to-Observation Texture Consistency

This loss enforces the predicted textures $\mathbf{T}$ to match the texture observed in the image $\widetilde{\mathbf{T}}$. Although this is similar to the photometric loss used in [22], a challenge in our technique is the aforementioned domain mismatch: $\mathbf{T}$ could be significantly different from $\widetilde{\mathbf{T}}$ mainly due to lighting condition changes. Therefore, we incorporate an additional network $\mathbf{T} \leftarrow \mathrm{C}(\mathbf{T})$ to convert the color of the predicted texture to the one of the currently observed texture. $\mathrm{C}(\mathbf{T})$ is also learned, and since training data is limited, we learn a single 1-by-1 convolutional filter which can be viewed as the color correction matrix and corrects the white-balance between the two textures. The model-to-observation texture consistency (MOTC) is formulated as

$$L^{t}_{\text{MOTC}} = \frac{1}{W^{t}} \left\| \mathbf{W}^{t} \odot \left(\widetilde{\mathbf{T}}^{t} - \mathrm{C}\left(\mathbf{T}^{t}\right) \right) \right\|_{F}^{2}. \tag{8}$$

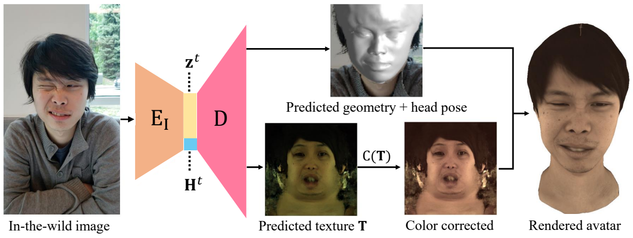

Figure 5: Proposed method during testing phase.

3.2.3 Facial Landmark Reprojection Consistency

This loss enforces a sparse set of vertices on the 3D mesh corresponding to the landmark locations to be consistent with 2D landmark predictions. Given K facial landmarks, the facial landmark reprojection consistency (FLRC) loss is formulated as:

$$L^{t}_{\text{FLRC}} = \frac{1}{K} \sum_{k} \left\| \mathbf{K}^{k,t}_{\text{2D}} - \mathbf{\Pi}\mathbf{H}^{t}_{\mathbf{I}}\mathbf{K}^{k}(\mathbf{G}^{t}_{\mathbf{I}}) \right\|_{2}^{2}, \tag{9}$$

where $\mathbf{K}^{k,t}_{\text{2D}}$ is the location of the k-th detected 2D landmark.

3.3. Testing Phase

Figure 5 depicts the steps required during the testing phase of our network, which is simply a feed-forward pass through the adapted $\mathrm{E}_{\mathbf{I}}$ and the estimated color correction function C. Note that $\widetilde{\mathbf{T}}$ and the landmark detection are no longer required. Therefore, the timing of the network is still exactly the same as the original network except for the additional color correction, which itself is simple and fast.

4. Experiments

To demonstrate the effectiveness of our proposed self-supervised domain adaptation method for high-fidelity 3D face tracking, we perform both quantitative and qualitative analysis. Though qualitative analysis is relatively straight forward, quantitative analysis for evaluating the accuracy and stability of tracking results requires a high-resolution in-the-wild video dataset with ground-truth 3D meshes, which unfortunately is difficult to collect because scanning high quality 3D facial scans usually requires being in a special lab environment with controlled settings. Thus quantitative analysis of recent 3D face tracking methods such as [22, 23] are limited to static image datasets [4], or video sequences shot in a controlled environment [25]. Therefore, in light of the aforementioned limitations, we collected a new dataset and devised two metrics for quantitatively evaluating 3D face tracking performance.

Evaluation metrics: We employ two metrics, accuracy and temporal stability, which are denoted as "Reprojection" and "Temporal" in Table 1, respectively. For accuracy, since we do not have ground truth 3D meshes for in-the-wild data, we utilize average 2D landmark reprojection error as a proxy for the accuracy of the predicted 3D geometry. First, a 3D point corresponding to a 2D landmark is projected into 2D, and then the Euclidean distance between the reprojected point and ground truth 2D point is computed. For temporal stability, we propose a smoothness metric as

$$\frac{1}{G}\sum_{i=1}^{G}\frac{\left\|\mathbf{G}_i^{t+1}-\mathbf{G}_i^{t}\right\|_2+\left\|\mathbf{G}_i^{t}-\mathbf{G}_i^{t-1}\right\|_2}{\left\|\mathbf{G}_i^{t+1}-\mathbf{G}_i^{t-1}\right\|_2}, \tag{10}$$

where $\mathbf{G}_i^t$ corresponds to the 3D location of vertex i at time t. This metric assumes that the vertices of the 3D mesh should move on a straight line over the course of three frames, thus unstable or jittering predictions will lead to higher (worse) score. The lowest (best) metric score is 1.

Dataset collection and annotation: We recorded 1920×1080 resolution facial performance data in the wild for four different identities. Recording environments include indoor, outdoor, plain background and cluttered background under various lighting conditions.

150 frames of facial performance data were annotated for each of the 4 identities. For each frame, we annotate on the person's face 5 salient landmarks that do **not** correspond to any typical facial landmark such as eye corners and mouth corners that can be detected by our landmark detector. These points are selected because our domain adaptation method already optimizes for facial landmark reprojection consistency, so our evaluation metric should use a separate set of landmarks for evaluation. Therefore, we focus on annotating salient personalized landmarks, such as pimples or moles on a person's face, which can be easily identified and accurately annotated by a human. In this way, our annotations enable us to measure performance of tracking in regions where there are no generic facial landmarks and provide a more accurate measure of tracking performance.

Implementation Details: DAMs [13] are first created for all four identities from multi-view images captured in a lighting-controlled environment, and our I2ZNet is newly trained for each identity. Our proposed self-supervised domain adaptation method is then applied to videos of the four identities in a different lighting and background environment. For DAM, the unwrapped texture resolution is $T = 1024$, and the geometry had $G = 7306$ vertices. We train the I2ZNet with Stochastic Gradient Decent (SGD). The face is cropped and resized to 256×256 image and given to $\mathrm{E}_{\mathbf{I}}$. During the self-supervised domain adaptation, the related parameters are set to $\lambda_{\text{CFTC}} = 100$, $\lambda_{\text{MOTC}} = 100$, $\lambda_{\text{FLRC}} = 1$.

Table 1: Evaluation on in-the-wild dataset. "Ours w/o DA" represents $\mathrm{E}_{\mathbf{I}}$ before doing any domain adaptation.

		Subject1	Subject2	Subject3	Subject4	Average
HPEN	Temporal	1.5197	1.2951	1.8206	1.3559	1.4978
	Reprojection	8.8075	**5.5475**	13.3823	10.4688	9.5515
3DDFA	Temporal	1.5503	1.4500	1.8608	1.5139	1.5938
	Reprojection	14.1171	10.2568	21.5077	18.1647	16.011
PRNet	Temporal	1.5551	1.3701	1.5700	1.4973	1.4981
	Reprojection	8.4867	7.2522	14.052	9.6586	9.8624
Ours w/o DA	Temporal	1.4106	1.2476	1.8322	1.4169	1.4768
	Reprojection	6.2171	7.4914	10.9225	9.5953	8.5566
Ours w/ L_{FLRC}	Temporal	1.3624	1.3274	1.6583	1.132	1.3700
	Reprojection	5.7558	6.982	10.1258	7.5230	7.5960
Ours	Temporal	**1.1299**	**1.0498**	**1.2934**	**1.0915**	**1.1412**
	Reprojection	**5.5689**	6.7281	**9.6015**	**7.1368**	**7.2588**

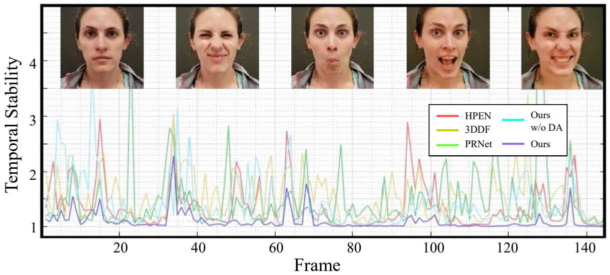

Figure 6: Temporal stability graph for subject 4. Note that smaller stability score means more stable results.

4.1. Results on In-the-wild Dataset

We compare our method against three state-of-the-art baselines: **HPEN** [28]: 3DMM fitting based on landmarks, **3DDFA** [27]: 3DMM fitting based on landmarks and dense correspondence, and **PRNet** [9]: 3DMM fitting based on the direct depth regression map. The system input image size is 256×256 except for **3DDFA** (100×100). We also add our method without domain adaptation (**Ours w/o DA**) and only with facial landmark reprojection consistency (**Ours w/ L_{FLRC}**). As shown in Table 1, the proposed domain adaptation consistently increases the performance of the our model without domain adaptation for all 4 sub-

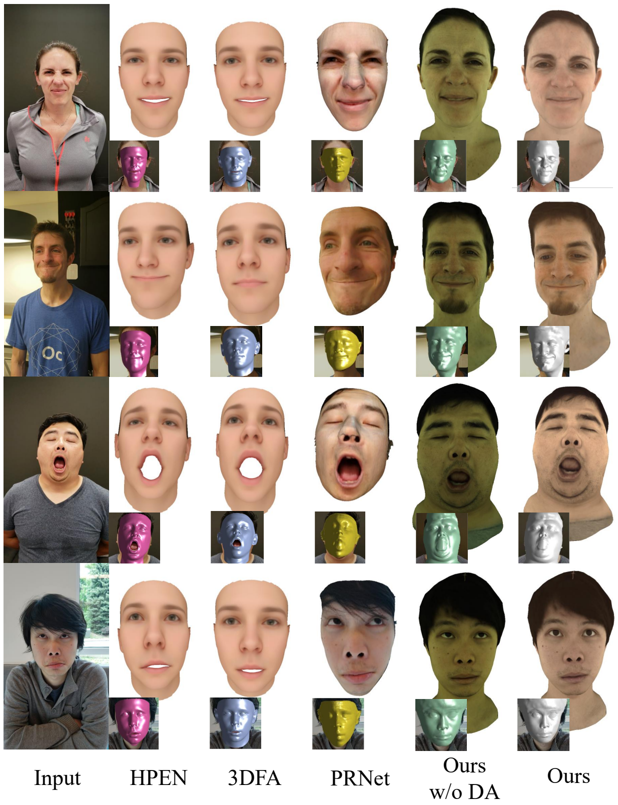

Figure 7: Qualitative comparisons with baseline methods.

jects. In terms of stability, the proposed domain adaptation method improves our model by 22% relative. Particularly, we are able to achieve 1.05 stability score for subject 2, which is close to the lowest possible stability score (1.0). This demonstrates the effectiveness of our proposed method. For the other baselines, our model without the domain adaptation already outperforms them in terms of geometry. This may be because our model is pre-trained with many pairs of $(\mathbf{I}, \mathbf{H}, \mathbf{z})$ training data, while the baselines were used out of the box. But on the other hand, all baselines including **Ours w/o DA** perform similarly in terms of stability (between 1.45-1.60), but our domain adaptation method is able to improve it to 1.14, which clearly demonstrates the effectiveness of our method.

Figure 6 visualizes the temporal stability metric for all the different methods for a single sequence. Our method has a consistently better (*i.e.*, smaller) stability score than all the other methods for nearly all the frames, and demonstrates not only the effectiveness, but also the reliability and robustness of our method for in-the-wild sequences.

Figure 7 shows qualitative comparisons with baselines. Overall, our face tracking results most closely resemble the input facial configuration, especially for the eyes and the mouth. For example, in the second row, the baselines erroneously predicted that the person's mouth is opened, while our method correctly predicted that the person's mouth is closed. We can also clearly see the effectiveness of our color correction approach, which is able to correct the relatively green-looking face to better match to the appearance in the input.

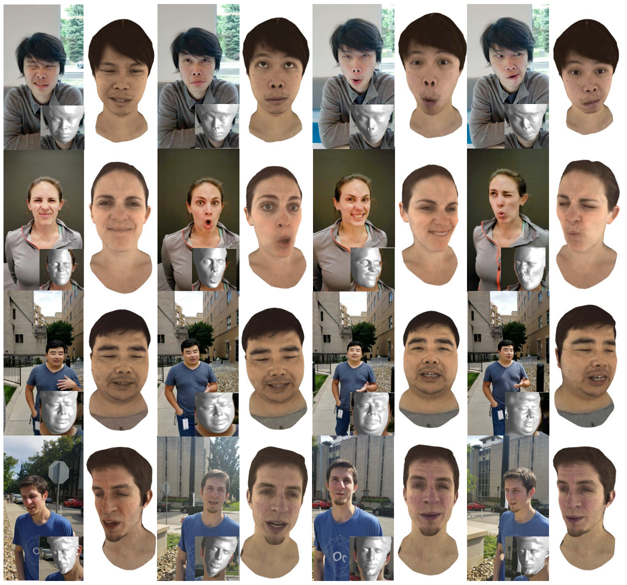

Figure 8: Visualization of 3D face tracking for in-the-wild video. For each input image, we show in the bottom right corner the predicted geometry overlaid on top of the face, and the predicted color corrected face.

Figure 8 shows the visualization of our in-the-wild face tracking results. Our method is able to track complex motion in many different backgrounds, head pose, and lighting conditions that are difficult to approximate with spherical harmonics such as hard shadow. Our method is also able to adapt to the white-balance of the current scene. Note that the gaze direction is also tracked for most cases.

4.2. Ablation Studies

To gain more insight to our model, we performed the following ablation experiments.

4.2.1 Evaluation of I2ZNet Structure

To validate the performance gain of each component on our regression network, we compare I2ZNet against three baseline networks: **VGG+Skip+Key** denotes I2ZNet, which uses VGGNet, multi-level features (skip connections), and landmarks from HourglassNet. **VGG+Skip**: landmarks guidance is removed. **VGG**: Multi-level features (skip connection) are further removed and only deep features are used for regression. **VGG Scratch** has the same structure with **VGG** but it is trained from scratch. For other settings which use **VGG**, pre-trained VGG-16 features are used, and

Table 2: Ablation test on I2ZNet. The average score with respect to all subjects are reported.

	VGG Scratch	VGG	VGG+Skip	VGG+Skip+Key
Geometry	1.011	1.481	0.411	**0.315**
Texture	0.016	0.027	0.007	**0.004**
Temporal	2.143	3.138	1.499	**1.446**

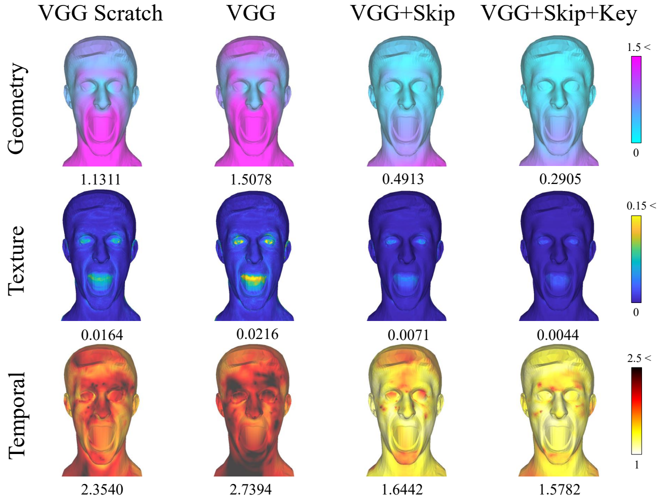

Figure 9: Ablation test on I2ZNet with a representative subject. The vertex-wise error is visualized with the associated average score for subject 1.

the VGG portion of the network is not updated during training. The models are tested on unseen test datasets where the vertex-wise dense ground-truth is available. Three metrics are employed to evaluate performance: (1) accuracy for geometry is computed by Euclidean distance between predicted and ground-truth 3D vertices, (2) accuracy for texture is calculated by pixel intensity difference between predicted and ground-truth texture, and (3) the temporal stability is measured in the same way as Eq. 10.

The average scores with respect to the four test subjects are reported in Table 2, and the representative subject results are visualized in Figure 9. We observe that multi-level features (**VGG+Skip**) significantly improves performance over **VGG**, while adding keypoints (**VGG+Skip+Key**) further improves performance. **VGG** seems to lack of capacity to directly regress the latent parameters with only pretrained deep features which are not updated. More ablation studies (*e.g.*, tests on view consistency and robustness to the synthetic visual perturbation) on I2ZNet are described in the supplementary manuscript.

4.2.2 Effect of Image Resolution

The cropped image resolution plays a key role in the accuracy of face tracking. In this experiment, we quantify the performance degradation according to the resolution using relative reprojection error metric. Relative reprojection error is computed by comparing the 2D reprojected vertices location of the estimated geometry from different resolution images with the one of the gold-standard geometry, which is the geometry acquired when using the highest image resolution 256×256. Figure 10 shows the results. Until 175×175, we achieve average error less than 4 pixel-error, but performance degrades significantly as the resolution becomes further smaller.

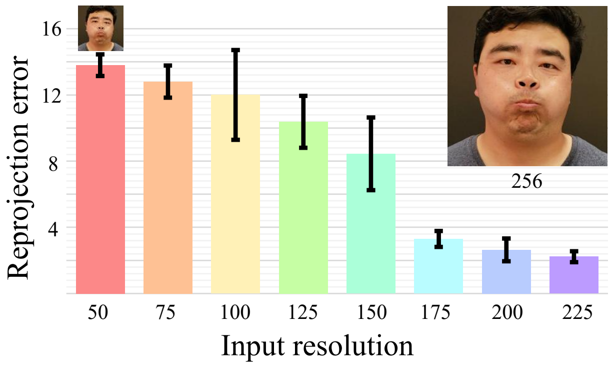

Figure 10: Ablation studies on the performance degradation under various input resolution.

4.3. Limitations

There are two main limitations to the proposed approach. The first limitation is that our method assumes that a person-specific DAM already exists for the person to be tracked, as our method takes the DAM as input. The second limitation is that our MOTC color correction cannot handle complex lighting and specularities. For example, in Figure 8 first row first image, a portion of the face is brighter due to the sun, but since we only have a global color correction matrix for color correction, the sun's effect could not be captured and thus not reflected in the output.

5. Conclusion

We proposed a deep neural network that predicts the intermediate representation and head pose of a high-fidelity 3D face model from a single image and its self-supervised domain adaptation method, thus enabling high-quality facial performance tracking from a monocular video in-the-wild. Our domain adaptation method leverages the assumption that the textures of a face over two consecutive frames should not change drastically, and this assumption enables us to extract supervision from unlabeled in-the-wild video frames to fine-tune the existing face tracker. The results demonstrated that the proposed method not only improves face-tracking accuracy, but also the stability of tracking.

References

[1] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3D faces. In *Proc. ACM SIGGRAPH*, pages 187–194, 1999. 1, 2

[2] James Booth, Epameinondas Antonakos, Stylianos Ploumpis, George Trigeorgis, and Yannis Panagakis andStefanos Zafeiriou. 3D face morphable models "in-the-wild". In *Proc. CVPR*, 2017. 2

[3] James Booth, Anastasios Roussos, Allan Ponniah, David Dunaway, and Stefanos Zafeiriou. Large scale 3D morphable models. *IJCV*, 126(2-4):233–254, 2018. 2

[4] Chen Cao, Yanlin Weng, Shun Zhou, Yiying Tong, and Kun Zhou. FaceWarehouse: A 3D facial expression database for visual computing. *IEEE TVCG*, 20(3):413–425, 2014. 2, 6

[5] Timothy F Cootes, Gareth J Edwards, and Christopher J Taylor. Active appearance models. *IEEE TPAMI*, (6):681–685, 2001. 1, 2

[6] Timothy F Cootes, Christopher J Taylor, David H Cooper, and Jim Graham. Active shape models-their training and application. *CVIU*, 61(1):38–59, 1995. 1, 2

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Proc. CVPR*, 2009. 12

[8] Xuanyi Dong, Shoou-I Yu, Xinshuo Weng, Shih-En Wei, Yi Yang, and Yaser Sheikh. Supervision-by-registration: An unsupervised approach to improve the precision of facial landmark detectors. In *Proc. CVPR*, 2018. 2, 3, 5

[9] Yao Feng, Fan Wu, Xiaohu Shao, Yanfeng Wang, and Xi Zhou. Joint 3D face reconstruction and dense alignment with position map regression network. In *Proc. ECCV*, 2018. 3, 6

[10] László A Jeni, Jeffrey F Cohn, and Takeo Kanade. Dense 3D face alignment from 2D video for real-time use. *Image Vision Comput.*, 58(C):13–24, 2017. 3

[11] Angjoo Kanazawa, Michael J. Black, David W. Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In *Proc. CVPR*, 2018. 4

[12] Diederik P. Kingma and Max Welling. Auto-encoding variational Bayes. In *Proc. ICLR*, 2014. 2

[13] Stephen Lombardi, Tomas Simon, Jason Saragih, and Yaser Sheikh. Deep appearance models for face rendering. *ACM TOG*, 37(4), 2018. 1, 2, 3, 6, 12

[14] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *Proc. ECCV*, 2016. 3, 11

[15] Elad Richardson, Matan Sela, and Ron Kimmel. 3D face reconstruction by learning from synthetic data. In *Proc. 3DV*, 2016. 3

[16] Elad Richardson, Matan Sela, Roy Or-El, and Ron Kimmel. Learning detailed face reconstruction from a single image. In *Proc. CVPR*, 2017. 3

[17] Sami Romdhani and Thomas Vetter. Estimating 3D shape and texture using pixel intensity, edges, specular highlights, texture constraints and a prior. In *Proc. CVPR*, 2005. 2

[18] Joseph Roth, Yiying Tong, and Xiaoming Liu. Adaptive 3D face reconstruction from unconstrained photo collections. In *Proc. CVPR*, 2016. 2

[19] Christos Sagonas, Epameinondas Antonakos, Georgios Tzimiropoulos, Stefanos Zafeiriou, and Maja Pantic. 300 faces in-the-wild challenge: Database and results. *Image Vision Comput.*, 47:3–18, 2016. 12

[20] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proc. ICLR*, 2015. 3, 11

[21] Ran Tao, Efstratios Gavves, and Arnold W. M. Smeulders. Siamese instance search for tracking. In *Proc. CVPR*, 2016. 12

[22] Ayush Tewari, Michael Zollhöfer, Pablo Garrido, Florian Bernard, Hyeongwoo Kim, Patrick Pérez, and Christian Theobalt. Self-supervised multi-level face model learning for monocular reconstruction at over 250 Hz. In *Proc. CVPR*, 2018. 2, 3, 4, 5, 6

[23] Ayush Tewari, Michael Zollhöfer, Hyeongwoo Kim, Pablo Garrido, Florian Bernard, Patrick Pérez, and Christian Theobalt. MoFA: Model-based deep convolutional face autoencoder for unsupervised monocular reconstruction. In *Proc. ICCV*, 2017. 2, 6

[24] Anh Tuan Tran, Tal Hassner, Iacopo Masi, and Gérard Medioni. Regressing robust and discriminative 3D morphable models with a very deep neural network. In *Proc. CVPR*, 2017. 3

[25] Levi Valgaerts, Chenglei Wu, Andrés Bruhn, Hans-Peter Seidel, and Christian Theobalt. Lightweight binocular facial performance capture under uncontrolled lighting. *ACM TOG*, 31(6):187–1, 2012. 6

[26] Jae Shin Yoon, Francois Rameau, Junsik Kim, Seokju Lee, Seunghak Shin, and In So Kweon. Pixel-level matching for video object segmentation using convolutional neural networks. In *Proc. ICCV*, 2017. 3, 12

[27] Xiangyu Zhu, Zhen Lei, Xiaoming Liu, Hailin Shi, and Stan Z. Li. Face alignment across large poses: A 3D solution. In *Proc. CVPR*, 2016. 6

[28] Xiangyu Zhu, Zhen Lei, Junjie Yan, Dong Yi, and Stan Z. Li. High-fidelity pose and expression normalization for face recognition in the wild. In *Proc. CVPR*, 2015. 6

[29] Xiangyu Zhu, Xiaoming Liu, Zhen Lei, and Stan Z. Li. Face alignment in full pose range: A 3D total solution. *IEEE TPAMI*, 2019. 3

Supplementary Material: Self-Supervised Adaptation of High-Fidelity Face Models for Monocular Performance Tracking

Jae Shin Yoon[†] Takaaki Shiratori[‡] Shoou-I Yu[‡] Hyun Soo Park[†]
[†]University of Minnesota [‡]Facebook Reality Labs
{jsyoon, hspark}@umn.edu {tshiratori, shoou-i.yu}@fb.com

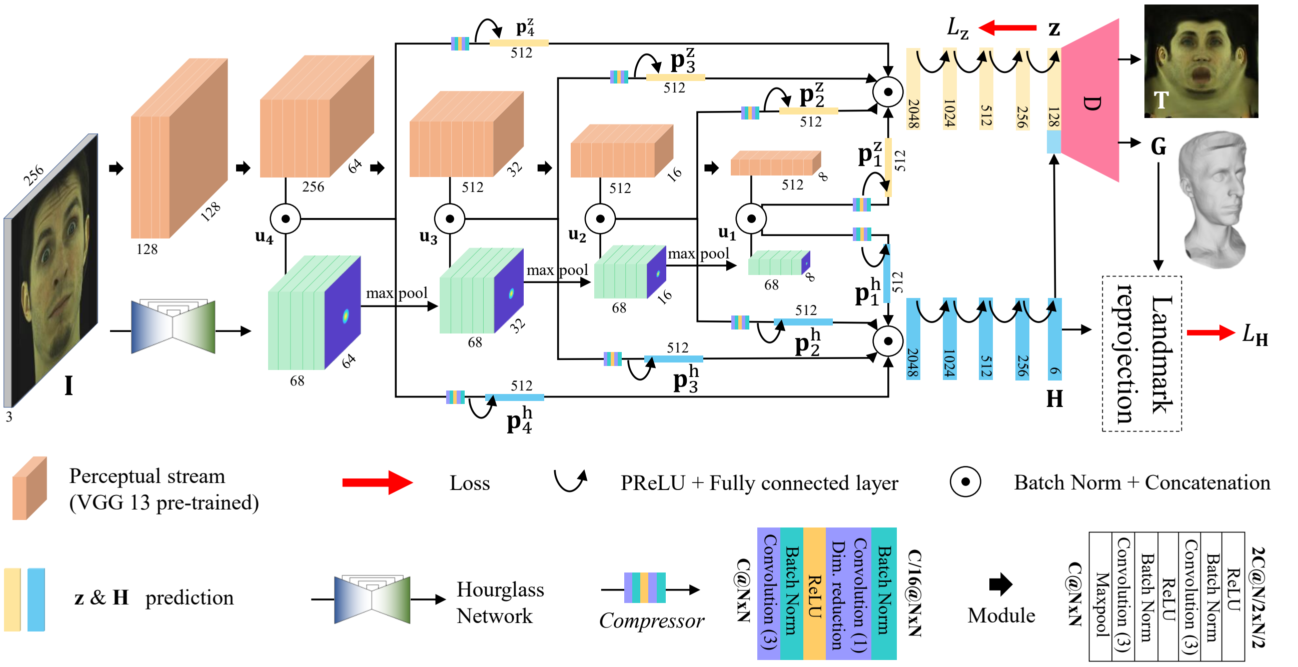

Figure 11: I2ZNet directly regresses the latent facial state codes **z** and headpose **H** from a face image **I**, and the pre-trained decoder D generates full 3D face geometry **G** and high resolution texture **T**.

In the supplementary materials, we provide details on the architecture of I2ZNet in Section A, and the additional ablation studies on I2ZNet will be followed in Section B.

A. I2ZNet

In this section, we detail the architecture of I2ZNet.

Other than only utilizing self-supervised domain adaptation to overcome domain differences, we also explored different networks which could lead to the most domain invariance. Namely, we utilized a combination of a pre-trained VGGNet [20] and HourglassNet [14] to achieve better domain invariance. More details are in Section A.2, and the property of domain invariant features are validated in Section B.

Domain specific layers are still necessary to complete our tasks, but thanks to the domain invariant features already extracted by VGGNet and HourglassNet, the domain specific layers can have less parameters thus they are easier to train. We use a combination of deep and shallow features to achieve better performance. More details are in Section A.3.

A.1. Inputs and Outputs

Given a cropped input face image $\mathbf{I} \in \mathbb{R}^{256\times256\times3}$, the I2ZNet directly predicts the low-dimensional facial state codes $\mathbf{z} \in \mathbb{R}^{128}$, and a set of head pose parameters $\mathbf{H} \in$

$\mathbb{R}^6 = \{f,\ r_x,\ r_y,\ r_z,\ t_x,\ t_y\}$, where $\mathbf{f} = \{f\}$, $\mathbf{r} = \{r_x,\ r_y,\ r_z\}$, $\mathbf{t} = \{t_x,\ t_y\}$ are focal length scale, Euler angle, and 2D translation respectively. The pre-trained decoder D decodes $[\mathbf{z}^\mathsf{T}, \mathbf{H}^\mathsf{T}]$ to generate high fidelity 3D face geometry $\mathbf{G} \in \mathbb{R}^{7306\times 3}$ and view dependent texture map $\mathbf{T} \in \mathbb{R}^{1024\times 1024\times 3}$. Note that, we are using the same decoder with [13], while we replace its encoder network E_X with our I2ZNet.

A.2. Domain Invariant Multi-level Unified Features

Given an input image **I**, I2ZNet extracts the features from two-stream networks: VGGNet and HourglassNet. VGGNet captures perceptual information such as facial details or shape, while HourglassNet guides "where to look" by providing facial geometry features, e.g. facial landmark heatmaps. We complete the multi-level unified features $\mathbf{u}_l \in \mathbb{R}^{(32*2^l)\times(32*2^l)\times ch_l}$ by concatenating the two-stream features, where $l = \{4, 3, 2, 1\}$ denotes the feature depth-level and the associated channel size is $ch_l \in CH = \{324, 580, 580, 580\}$. Here, we simply max-pool the output from HourglassNet to make the feature size equal to each level of VGG feature. The feature scale inconsistency between two different networks (VGGNet and HourglassNet) is resolved by normalization layer before concatenation. Our multi-level unified features are more domain (color, illumination, or head pose) invariant by learning from domain generalized datasets [7, 19]. Note that, the pre-learned weights on the two-stream networks are fixed in the following training steps such that we prevent I2ZNet from being domain specific.

A.3. Latent Parameter Regression

Inspired by many recent papers [21, 26] which have proposed the use of combination of deep and shallow features to capture semantic-level information and local appearance details at the same time, we concatenate feature vectors from each depth level $\mathbf{p}^z_{4..1}$, $\mathbf{p}^h_{4..1} \in \mathbb{R}^{512}$, which are encoded from $\mathbf{u}_{4..1}$, and they are respectively regressed to $\mathbf{z}$ and $\mathbf{H}$ using several fully connected layers. Here, however, it requires very heavy computational costs for converting three-dimensional features $\mathbf{u}_l$ to single dimensional one $\mathbf{p}^{z,h}_l$ in a fully connected way. Similar to [26], we alleviate this bottleneck by channel-wise feature compression of $\mathbf{u}_l$ to one-sixteenth of its original channel size using two convolutional layers as described as *Compressor* layer in Figure 11.

B. Ablation Studies on I2ZNet

In Section A, we introduced the domain and view invariant property of our network. To verify this, we test I2ZNet on four different scenarios, **View**, **Color**, **Light**, and **Jitter**, where the baseline networks are the same with the ones described in Section 4.2.1.

Table 3: Ablation studies on I2ZNet.

		View	Color	Light	Jitter	Background
VGG Scratch	Geometry	0.607	1.485	1.175	0.983	1.285
	Headpose	-	17.48	6.965	-	15.84
	Texture	-	0.021	0.014	0.016	0.015
VGG	Geometry	1.352	1.258	1.510	1.736	1.076
	Headpose	-	16.61	13.98	-	16.42
	Texture	-	0.020	0.021	0.025	0.016
VGG +Skip	Geometry	0.3967	0.622	0.227	1.331	0.669
	Headpose	-	2.579	0.728	-	8.750
	Texture	-	0.009	0.003	0.018	0.009
VGG +Skip +Key	Geometry	**0.255**	**0.505**	**0.151**	**0.896**	**0.417**
	Headpose	-	**1.676**	**0.684**	-	**8.172**
	Texture	-	**0.007**	**0.002**	**0.012**	**0.006**

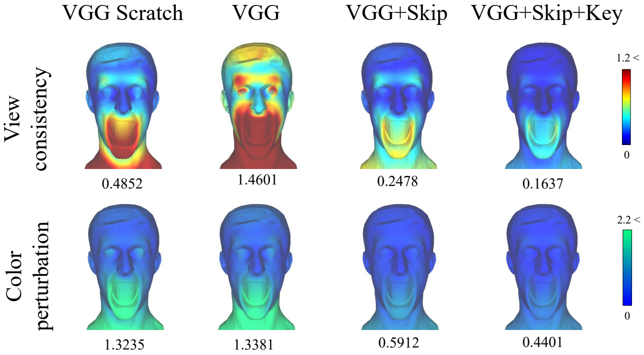

Figure 12: Visualization of the vertex-wise accuracy with a representative subject for ablation studies on view consistency and color sensitivity. The average score is reported for each metric, where the lower score shows the better performance for both scenarios.

View represents the test dataset of multiview videos, where they are accurately synchronized and thus I2ZNet should predict the same facial local deformation to make the facial configuration consistent across the views. To verify this view consistent prediction ability, we pick the most central camera as a ground-truth view and evaluate the performance of other views. We use simple vertex-wise Euclidean distance between the 3D faces predicted from central view and other views meaning that the lower score shows better consistency. The overall performance is summarized in Table 3 and Figure 12, where the proposed network outperforms all other baselines. We can further notice that the combination of skip connection and landmark guidance helps the network to figure out the facial geometry configuration when predicting the facial configuration from different views based on the comparison of **VGG** with **VGG+Skip** and **VGG+Skip+Key**. Note that, when evaluating the view consistency, we remove the texture and head pose from a predicted 3D face because they have view dependent property in our system.

Color, **Light**, **Jitter**, and **Background** represent video sequences which contain synthetic perturbation with random color, gamma, jitters by similarity transformation (scale, rotation, and translation variation), and white dotted background noise. The goal of the test on these sequences is to verify the domain generality. For example, if I2ZNet outputs a completely different 3D facial configuration given a perturbed image comparing to the one before the perturbation, then it implies that the network is overfitted to the training data domain. Therefore, we evaluate the performance of I2ZNet on the sequence after the perturbation in light of the results from the ones before the perturbation. To measure this relative accuracy, we employ three metrics: geometry, texture, and head pose. For geometry and texture, we simply calculate the 3D distance and color difference of the 3D faces. For head pose, we measure the 2D distance between the ground-truth points and the reprojection of the vertices on the 3D face to the input with the predicted head pose. The average scores with respect to the entire test subjects (4 subjects) are reported in Table 3, and the representative subject results are visualized in Figure 12. From the comparison of **VGG Scratch** with **VGG+Skip+Key**, we can notice that the pre-trained nature of the feature extraction parts (VGGNet and HourglassNet) plays a key role to avoid overfitting from a specific domain. Further, the comparison between **VGG+Skip** and **VGG+Skip+Key** implies that the landmark module guides the attention of the network such that it prevents from the network distraction even under the background perturbation.