3D Face Tracking from 2D Video through Iterative Dense UV to Image Flow

Felix Taubner Prashant Raina Mathieu Tuli Eu Wern Teh Chul Lee Jinmiao Huang
LG Electronics
{prashant.raina, mathieu.tuli, euwern.teh, clee.lee}@lge.com

Abstract

*When working with 3D facial data, improving fidelity and avoiding the uncanny valley effect is critically dependent on accurate 3D facial performance capture. Because such methods are expensive and due to the widespread availability of 2D videos, recent methods have focused on how to perform monocular 3D face tracking. However, these methods often fall short in capturing precise facial movements due to limitations in their network architecture, training, and evaluation processes. Addressing these challenges, we propose a novel face tracker, **FlowFace**, that introduces an innovative 2D alignment network for dense per-vertex alignment. Unlike prior work, FlowFace is trained on high-quality 3D scan annotations rather than weak supervision or synthetic data. Our 3D model fitting module jointly fits a 3D face model from one or many observations, integrating existing neutral shape priors for enhanced identity and expression disentanglement and per-vertex deformations for detailed facial feature reconstruction. Additionally, we propose a novel metric and benchmark for assessing tracking accuracy. Our method exhibits superior performance on both custom and publicly available benchmarks. We further validate the effectiveness of our tracker by generating high-quality 3D data from 2D videos, which leads to performance gains on downstream tasks.*

1. Introduction

Access to 3D face tracking data lays the foundation for many computer graphics tasks such as 3D facial animation, 3D human avatar reconstruction, and expression transfer. Obtaining high visual fidelity, portraying subtle emotional cues, and preventing the uncanny valley effect in these downstream tasks is reliant on high motion capture accuracy. As a result, a common approach to generating 3D face tracking data is to use 3D scans and visual markers however, this process is cost-intensive. To alleviate this burden, building computational models to obtain 3D faces from monocular 2D videos and images has cemented its importance in recent years and seen great progress [10, 14, 19, 24, 37, 42, 57]. Nevertheless, three issues persist: First, current methods rely heavily on sparse landmarks and photometric similarity, which is computationally expensive and ineffective in ensuring accurate face motion. Second, the monocular face tracking problem is both ill-posed and contains a large solution space dependent on camera intrinsics, pose, head shape, and expression [58]. Third, current benchmarks for this task neglect the temporal aspect of face tracking and do not adequately evaluate facial motion capture accuracy.

To address the aforementioned issues, we introduce a novel 3D face tracking model called **FlowFace**, consisting of a versatile two-stage pipeline: A 2D alignment network that predicts the screen-space positions of each vertex of a 3D morphable model [2] (3DMM) and an optimization module that jointly fits this model across multiple views by minimizing an alignment energy function. Unlike traditional methods that rely on sparse landmarks and photometric consistency, FlowFace uses only 2D alignment as input signal, similar to recent work [42]. This alleviates the computational burden of inverse rendering and allows joint reconstruction using a very large number of observations. We enhance previous work in four ways: (1) The 2D alignment network features a novel architecture with a vision-transformer backbone and an iterative, recurrent refinement block. (2) In contrast to previous methods that use weak supervision or synthetic data, the alignment network is trained using high-quality annotations from 3D scans. (3) The alignment network predicts dense, per-vertex alignment instead of key-points, which enables the reconstruction of finer details. (4) We integrate an off-the-shelf neutral shape prediction model to improve identity and expression disentanglement.

In addition, we present the screen-space motion error (SSME) as a novel face tracking metric. Based on optical flow, SSME computes and contrasts screen-space motion, aiming to resolve the limitation observed in existing evaluation methods. These often rely on sparse key points, synthetic annotations, or RGB/3D reconstruction errors, and lack a thorough and comprehensive measurement of temporal consistency. Using the Multiface [44] dataset, we develop a 3D face tracking benchmark around this metric.

Finally, through extensive experiments on available benchmarks, we show that our method significantly outperforms the state-of-the-art on various tasks. To round off our work, we demonstrate how our face tracker can positively affect the performance of downstream tasks, including speech-driven 3D facial animation and 3D head avatar synthesis. Specifically, we demonstrate how our method can be used to generate high-quality data — comparable to studio-captured data — for both these tasks by using it to augment existing models to achieve state-of-the-art results.

2. Related Work

Uncalibrated 3D Face Reconstruction. Previous work reconstructing 3D face shapes from uncalibrated 2D images or video fall into two broad categories:

Optimization-based methods recover face shape and motion by jointly optimizing 3D model parameters to fit the 2D observations. They traditionally treat this optimization as an inverse rendering problem [15, 16, 37, 43, 48, 52, 57], using sparse key-points as guidance. Typically, they employ geometric priors such as 3DMMs [2, 6, 22, 26, 47], texture models, simplified illumination models, and temporal priors. Some methods use additional constraints such as depth [37] or optical flow [5]. [58] and [28] present detailed surveys of such methods. Most methods use 3DMMs to disentangle shape and expression components. MPT [57] is the first method to integrate metrical head shape priors predicted by a deep neural network (DNN). However, photometric and sparse landmark supervision is not sufficient to obtain consistent and accurate face alignment, especially in areas not covered by landmarks and or of low visual saliency. More recently, [42] proposes to use only 2D face alignment (dense landmarks) as supervision, avoiding the computationally expensive inverse rendering process. Our method extends this idea with an improved 2D alignment module, better shape priors, and per-vertex deformation.

Regression-based methods train DNNs to directly predict face reconstructions from single images [7, 10, 12, 19, 24, 31, 32, 34, 35]. This reconstruction includes information such as pose, 3DMM components, and sometimes texture. Typically, convolutional networks like image classification networks [21, 33] or encoder-decoder networks [41] are used. Due to the lack of large-scale 2D to 3D annotations, these methods typically rely on photometric supervision for their training. Some methods propose complex multi-step network architectures [24, 32] to improve reconstruction. [24] use additional handcrafted losses to improve alignment, whereas [7] use synthetic data and numerous of landmarks. More recently, [38] proposes to use vision-transformers to improve face reconstruction.

2D Face Alignment. Traditional 2D face alignment methods predict a sparse set of manually defined landmarks. These methods typically involve convolutional DNNs to predict heat maps for each landmark [4, 30, 54]. Sparse key-points are not sufficient to describe full face motion, and heat maps make it computationally infeasible to predict a larger number of key-points. [42] and [18] achieve pseudo-dense alignment by using classifier networks to directly predict a very large number of landmarks. [20] predict the UV coordinates in image space and then map the vertices onto the image. Just like [41] and [32], our method predicts a per-pixel dense mapping between the UV space of a face model and the image space. However, we set our method apart by using better network architectures with vision-transformers and real instead of synthetic data.

Evaluation of Face Trackers. Prior work evaluates face tracking and reconstruction using key-point accuracy [19, 32, 41, 42, 55], depth [37, 57], photometric [37, 57] or 3D reconstruction [5, 6, 47] errors. Sparse key-points are usually manually-annotated, difficult to define without ambiguities [54], and insufficient to describe the full motion of the face. Dense key-points [55] are difficult to compare between models using different mesh topologies. Photometric errors [37, 38, 57] are unsuitable since a perfect solution already exists within the input data, and areas with low visual saliency are neglected. A fair comparison of depth errors [37, 57] is only possible for methods using a pre-calibrated, perspective camera model. Methods that evaluate 3D reconstruction errors have to rigidly align the target and predicted mesh to fairly evaluate results [6, 34, 47], which causes valuable tracking information such as pose and intrinsics to be lost. Most importantly, depth and 3D reconstruction metrics neglect motion tangential to the surface normal. In contrast, our proposed metric measures the dense face motion in screen space, which is topology-independent and eliminates the need for rigid alignment.

3. Method

Our 3D face tracking pipeline consists of two stages: The first stage is predicting a dense 2D alignment of the face model, and the second stage is fitting a parametric 3D model to this alignment.

3.1. Dense 2D Face Alignment Network

3.1.1 Network Architecture

The 2D alignment module is responsible for predicting the probabilistic location — in image space — of each vertex of our face model. As in [42], the 2D alignment of each vertex is represented as a random variable $A_i = \{\mu_i, \sigma_i\}$. $\mu_i = [x_i, y_i] \in \mathcal{I}$ is the expected vertex position in image space $\mathcal{I} \in [0, D_{img}]^2$, and $\sigma_i \in \mathbb{R}_{>0}$ is its uncertainty, modeled as the standard deviation of a circular 2D Gaussian density function. As an intermediate step, for each iteration

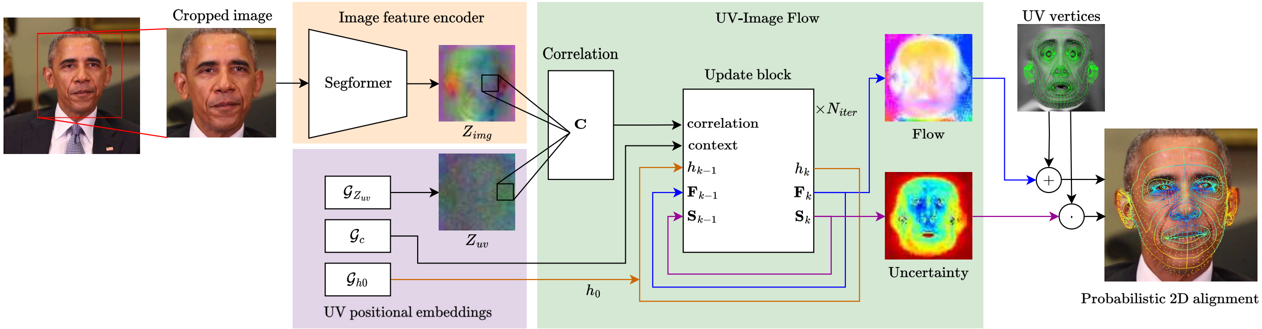

Figure 1. An overview of the proposed 2D alignment network architecture. A feature encoder transforms the image into a latent feature map that is then iteratively aligned with a learned UV positional embedding map by the recurrent update block.

k, the alignment network predicts a dense UV to image correspondence map $\mathbf{F}_k : \mathcal{U} \to \mathcal{I}$ and uncertainty map $\mathbf{S}_k$. $\mathbf{F}_k$ maps any point in UV space $\mathcal{U} \in [0, D_{uv}]^2$ to a position in image space through a pixel-wise offset, which we call *UV-image flow*. This network consists of three parts (Fig. 1):

1. An image feature encoder producing a latent feature map of the target image.
2. A positional encoding module that produces learned positional embeddings in UV space.
3. An iterative, recurrent optical flow module that predicts the probabilistic UV-image flow.

The image space position and uncertainty of each vertex is then bi-linearly sampled from the intermediate correspondence and uncertainty map for each iteration:

$$\mu_{i,k} = \nu_i + \mathbf{F}_k(\nu_i) \quad \text{and} \quad \sigma_{i,k} = \mathbf{S}_k(\nu_i) \tag{1}$$

where $\nu_i \in \mathcal{U}$ denotes the pre-defined UV coordinate of each vertex. These are manually defined by a 3D artist.

Image feature encoder. To obtain the input to the image encoder $\mathcal{F}$, we use SFD [51] to detect a square face bounding box from the target image and enlarge it by 20%. We then crop the image to the bounding box and resize it to D_{img}. We use Segformer [45] as the backbone, and replace the final classification layer with a linear layer to produce a 128-dimensional feature encoding. We further down-sample it to attain a final image feature map $Z_{img} \in \mathbb{R}^{D_{uv} \times D_{uv} \times 128}$ through average pooling. With image $\mathbf{I}$ and network parameters $\theta_{\mathcal{F}}$, this is defined as:

$$Z_{img} = \mathcal{F}(\mathbf{I}, \theta_{\mathcal{F}}) \tag{2}$$

UV positional encoding module. We use a set of modules $\mathcal{G}$ with identical architecture to generate learned positional embeddings in UV-space. Each module is comprised of a multi-scale texture pyramid and a pixel-wise linear layer. This pyramid consists of four trainable textures with 32 channels and squared resolutions of D_{uv}, $\frac{D_{uv}}{2}$, $\frac{D_{uv}}{4}$, and $\frac{D_{uv}}{8}$ respectively. Each texture is upsampled to D_{uv} through bi-linear interpolation before concatenating them along the channel dimension. The concatenated textures are then passed through a pixel-wise linear layer to produce the UV positional embeddings. The multi-scale setup ensures structural consistency in UV space (closer pixels in UV should have similar features). We use 3 of these modules: $\mathcal{G}_{Z_{uv}}$ to generate a UV feature map Z_{uv}, $\mathcal{G}_c$ to generator a context map c, and $\mathcal{G}_{h_0}$ to generate an initial hidden state h_0. With corresponding network parameters $\theta_{\mathcal{G}_{Z_{uv}}}$, $\theta_{\mathcal{G}_c}$ and $\theta_{\mathcal{G}_{h_0}}$, this is described as:

$$Z_{uv} = \mathcal{G}(\theta_{\mathcal{G}_{Z_{uv}}}); \quad c = \mathcal{G}(\theta_{\mathcal{G}_c}); \quad h_0 = \mathcal{G}(\theta_{\mathcal{G}_{h_0}}) \tag{3}$$

UV-image flow. The RAFT [36] network is designed to predict the optical flow between two images. It consists of a correlation block that maps the latent features encoded from each image into a 4D correlation volume. A context encoder initializes the hidden state of a recurrent update block and provides it with additional context information. The update block then iteratively refines a flow estimate while sampling the correlation volume.

We adapt this network to predict the UV-image flow $\mathbf{F} \in \mathbb{R}^{D_{uv} \times D_{uv} \times 2}$. We directly pass Z_{uv} and Z_{img} to the correlation block $\mathbf{C}$. We use the context map c and initial hidden state h_0 from the positional encoding modules for the update module $\mathbf{U}$. We modify the update module to also predict a per-iteration uncertainty in addition to the flow estimate, by duplicating the flow prediction head to predict a 1-channel uncertainty map $\mathbf{S} \in \mathbb{R}_{>0}^{D_{uv} \times D_{uv}}$. An exponential operation is applied to ensure positive values. The motion encoder head is adjusted to accept the uncertainty as an input. The modified RAFT network then works as follows: For each iteration k, the recurrent update module performs a look-up in the correlation volume, context map c, previous hidden state h_{k-1}, previous flow $\mathbf{F}_{k-1}$ and previous uncertainty $\mathbf{S}_{k-1}$. It outputs the refined flow estimate $\mathbf{F}_k$ and uncertainty $\mathbf{S}_k$ and the subsequent hidden state h_k. Formally,

$$\mathbf{F}_k, \mathbf{S}_k, h_k = \mathbf{U}(\mathbf{C}(Z_{uv}, Z_{img}), c, \mathbf{F}_{k-1}, \mathbf{S}_{k-1}, h_{k-1}, \theta_{\mathbf{U}}) \tag{4}$$

with update module weights $\theta_{\mathbf{U}}$. For a detailed explanation of our modified RAFT, we defer to [36] and Appendix B.

3.1.2 Loss Functions

We supervise our network with Gaussian negative log-likelihood (GNLL) both on the probabilistic per-vertex positions and the dense UV-image flow. For each iteration k of the update module, we apply the per-vertex loss function:

$$L_k^{vertex} = \sum_{i=1}^{N_v} \lambda_i (\log(\sigma_{i,k}^2) + \frac{\| \mu_{i,k} - \mu_i' \|^2}{2\sigma_{i,k}^2}) \tag{5}$$

where λ_i is a pre-defined vertex weight and μ_i' is the ground truth vertex position. We encourage our network to predict coherent flow and uncertainty maps in areas with no vertices by applying the GNLL loss for each pixel p in UV space:

$$L_k^{dense} = \sum_{p \in |\mathcal{U}|} \lambda_p (\log(\mathbf{S}_{k,p}^2) + \frac{\| \mathbf{F}_{k,p} - \mathbf{F}_p' \|^2}{2\mathbf{S}_{k,p}^2}) \tag{6}$$

where λ_p is a pre-defined per-pixel weight and $\mathbf{F}'$ is the ground truth UV-image flow. The final loss is a weighted sum of these losses, with a decay factor for each iteration of $\alpha = 0.8$ and a dense weight of $\lambda_{dense} = 0.01$:

$$\text{Loss} = \sum_{k=1}^{N_{iter}} \alpha^{N_{iter}-k} (L_k^{vertex} + \lambda_{dense} L_k^{dense}) \tag{7}$$

3.2. 3D Model Fitting

As in [42], the 3D reconstruction is obtained by jointly fitting a 3D head model and camera parameters to the predicted 2D alignment observations for the entire sequence. This is done by optimizing the energy function $E(\Phi; A)$ w.r.t to the model parameters Φ and alignment A (see Fig. 2). These parameters and the energy terms are defined below.

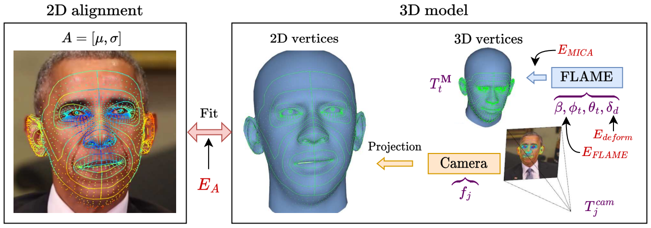

Figure 2. An illustration of the 3D model fitting process.

3.2.1 Tracking Model and Parameters

The tracking model consists of a 3D head model and a camera model. A tracking sequence contains C cameras, F frames with a total of $C \times F$ images.

3D head model. We use FLAME [26] as our 3D head model $\mathbf{M}$. This model consists of $N_v = 5023$ vertices, which are controlled by identity shape parameters $\boldsymbol{\beta} \in \mathbb{R}^{300}$, expression shape parameters $\boldsymbol{\phi} \in \mathbb{R}^{100}$ and $K = 5$ skeletal joint poses $\boldsymbol{\theta} \in \mathbb{R}^{3K+3}$ (including the root translation) through linear blend skinning [25]. We ignore root, neck and jaw pose and use the FLAME2023 model, which includes deformations due to jaw rotation within the expression blend-shapes. We also introduce additional static per-vertex deformations $\delta_d \in \mathbb{R}^{N_v \times 3}$ to enhance identity shape detail. The local head model vertices can be expressed using its parameters as follows:

$$\mathbf{M}(\boldsymbol{\beta}, \boldsymbol{\delta}_d, \boldsymbol{\phi}, \boldsymbol{\theta}) = \mathit{FLAME}(\boldsymbol{\beta}, \boldsymbol{\phi}, \boldsymbol{\theta}) + \boldsymbol{\delta}_\mathrm{d} \tag{8}$$

The rigid transform $\mathbf{T}^{\mathbf{M}} \in \mathbb{R}^{3\times4}$ represents the head pose, which transforms head model vertices i into world space for each frame t:

$$\mathbf{x}_{i,t}^{\mathrm{3D}} = \mathbf{T}_t^{\mathbf{M}} \mathbf{M}_i \tag{9}$$

Camera model. The cameras are described by the world-to-camera rigid transform $\mathbf{T}_{cam} \in \mathbb{R}^{3\times4}$ and the pinhole camera projection matrix $\mathbf{K} \in \mathbb{R}^{3\times3}$ defined by a single focal length $\boldsymbol{f} \in \mathbb{R}$ parameter. The camera model defines the image-space projection of the 3D vertices in camera j:

$$\mathbf{x}_{i,j,t}^{\mathrm{2D}} = \mathbf{K}_j \mathbf{T}_j^{cam} \mathbf{x}_{i,t}^{\mathrm{3D}} \tag{10}$$

Parameters. The parameters Ψ consist of the head model and camera parameters, which are optimized to minimize $E(\Phi; A)$. The camera parameters can be fixed to known values, if the calibration is available. Expression and poses vary for each frame t, whereas camera, identity shape, and deformation parameters are shared over the sequence.

$$\boldsymbol{\Psi} = \{\boldsymbol{\beta}, \Phi_{F\times|\boldsymbol{\phi}|}, \boldsymbol{\Theta}_{F\times|\boldsymbol{\theta}|}, \boldsymbol{\delta}_\mathrm{d}; \mathbf{T}_{F\times3\times4}^{\mathbf{M}}; \mathbf{T}_{C\times3\times4}^{cam}, \boldsymbol{f}_C\} \tag{11}$$

3.2.2 Energy Terms

The energy function is defined as:

$$E(\Phi; A) = E_A + E_{FLAME} + E_{temp} + E_{MICA} + E_\mathrm{deform} \tag{12}$$

$\boldsymbol{E_A}$ encourages 2D alignment:

$$E_A = \sum_{i,j,t}^{N_v,C,F} \lambda_i \frac{\| \mathbf{x}_{i,j,t}^{\mathrm{2D}} - \mu_{i,j,t} \|^2}{2\sigma_{i,j,t}^2} \tag{13}$$

where for vertex i seen by camera j in frame t. $\mu_{i,j,t}$ and $\sigma_{i,j,t}$ is the 2D location and uncertainty predicted by the final iteration of our 2D alignment network, and $\mathbf{x}_{i,j,t}^{\mathrm{2D}}$ (Eq. (10)) is the 2D camera projection of that vertex.

$\boldsymbol{E_{FLAME}} = \lambda_{FLAME}(\| \ \beta \ \|^2 + \| \ \Phi \ \|^2)$ encourages the optimizer to explain the data with smaller identity and expression parameters. This leads to face shapes that are statistically more likely [10, 14, 26, 57] and a more accurate 3D reconstruction. We do not penalize joint rotation, face translation or rotation.

$\boldsymbol{E_{temp}}$ applies a loss on the acceleration of the 3D position $\mathbf{x}_{i,t}^{\mathrm{3D}}$ of every vertex of the 3D model to prevent jitter and encourage a smoother, more natural face motion:

$$E_{temp} = \lambda_{temp} \sum_{i,j,t=2}^{N_v,C,F-1} \| \ \mathbf{x}_{j,t-1}^{\mathrm{3D}} - 2\mathbf{x}_{j,t}^{\mathrm{3D}} + \mathbf{x}_{j,t+1}^{\mathrm{3D}} \ \|^2 \tag{14}$$

$\boldsymbol{E_{MICA}} = \lambda_{MICA} \parallel \mathbf{M}_{\mathbf{\Phi}=0,\boldsymbol{\theta}=0} - \mathbf{M}_{MICA} \parallel^2$ provides a 3D neutral geometry prior for the optimizer to enable a better disentanglement between identity and expression components. It consists of the L2 distance of the neutral head model vertices to the MICA [57] template $\mathbf{M}_{MICA}$. This template is computed by predicting the average neutral head vertices using the MICA model [57] for all frames of the sequence. The term also enables a more accurate 3D reconstruction since the model can rely on MICA predictions where the alignment is uncertain, such as in the depth direction or for occluded vertices. In areas of confident alignment, the MICA prediction can be refined.

$\boldsymbol{E_{deform}} = \lambda_{deform} \parallel \boldsymbol{\delta}_{\mathrm{d}} \parallel^2$ encourages per-vertex deformations to be small w.r.t. the FLAME model.

3.3. Multiface Face Tracking Benchmark

Our monocular 3D face tracking benchmark focuses on 3D reconstruction and motion capture accuracy. To evaluate these, we use our proposed screen space motion error (SSME) and the scan-to-mesh chamfer distance (CD).

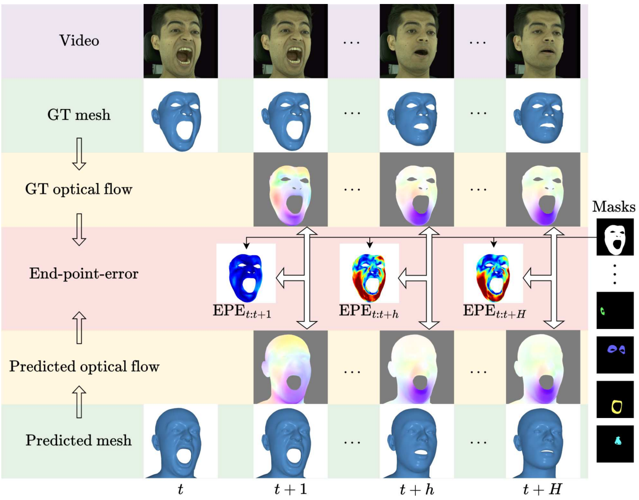

Figure 3. An illustration of the EPE computation for each frame.

Screen Space Motion Error. To define the **S**creen **S**pace **M**otion **E**rror (SSME), we reformulate face tracking as an optical flow prediction problem over a set of time windows. First, we project the ground truth mesh and predicted mesh into screen space using the respective camera model. Then, we use the screen space coordinates to compute the ground truth optical flow $\mathbf{f}'_{t:t+h}$ and predicted optical flow $\mathbf{f}_{t:t+h}$ from frame t to frame $t+h$ for each frame $t \in [1, \dots, F]$ and a sequence of frame windows $h = [1, ..., N_H]$. For each frame and frame window, the average end-point-error $EPE_{t:t+h}$ is computed by averaging the L2-distance between ground truth and predicted optical flow for each pixel (see Fig. 3).

$$EPE_{t:t+h} = \parallel V \odot (\mathbf{f}_{t:t+h} - \mathbf{f}'_{t:t+h}) \parallel^2 \tag{15}$$

where V is a mask to separate different face regions and $\odot$ is the Hadamard product. See Fig. 3 for a visual reference.

The screen space motion error $SSME_h$ for frame window h is then defined as the mean of all EPEs over all frames t where frame $t+h$ exists:

$$SSME_h = \frac{1}{F-h} \sum_{t=1}^{t+h \leq F} EPE_{t:t+h} \tag{16}$$

Finally, to summarize tracking performance in one value, we compute the average screen space motion error $\overline{SSME}$ over all frame windows as

$$\overline{SSME} = \sum_{h=1}^{N_H} SSME_h \tag{17}$$

In other words, $\overline{SSME}$ measures the average trajectory accuracy of each pixel over a time horizon of N_H frames. We choose a maximum frame window of $N_H = 30$ (1 second) since most human expressions are performed within this time frame. Because the screen space motion is directly affected by most face-tracking parameters such as intrinsics, pose, and face shape, it also measures their precision in a holistic manner. In contrast to prior works and benchmarks that use sparse key-points, SSME covers the motion of all visible face regions and is invariant to mesh topology. As it operates in screen space, it does not require additional alignment and works with all camera models, unlike 3D reconstruction or depth errors. In our benchmark, we evaluate SSME over a set of masks for semantically meaningful face regions (face, eyes, nose, mouth, and ears) (Fig. 3), permitting a more nuanced analysis of the tracking performance.

3D Reconstruction. To complete our benchmark, we additionally measure the chamfer distance (CD) to account for the depth dimension. Similar to [34], the tracked mesh is rigidly aligned to the ground truth mesh using 7 key-points and ICP. Then, the distance of each ground truth vertex with respect to the predicted mesh is computed and averaged. For a detailed explanation, we defer to the NoW benchmark [34]. Just like the SSME, we evaluate the CD for the same set of face regions to provide a more detailed analysis of reconstruction accuracy, similar to [6].

Multiface Dataset. We build our benchmark around the Multiface dataset [44]. Multiface consists of multi-view videos with high quality topologically consistent 3D registrations. High-resolution videos are captured at 30 FPS from a large variety of calibrated views. We limit the evaluation data to a manageable size by carefully selecting a subset of 86 sequences with a diverse set of view directions and facial performances (see Appendix C).

4. Experiments

Training data. To train the 2D alignment network, we use a combined dataset made up of FaceScape [47], Stirling [1], and FaMoS [3]. Where a FLAME [26] registration is not available, we fit the FLAME template mesh to the 3D scan through semi-automatic key-point annotation and commercial topology fitting software. For an accurate capture of face motion, we auto-annotate expression scans with additional key-points propagated with optical flow (more information in Appendix D). The ground truth image space vertex positions μ' are obtained by projecting the vertices of the fitted FLAME mesh into screen space using the available camera calibrations.

Training strategy for 2D alignment network. We use Segformer-b5 (pre-trained on ImageNet [11]) as our backbone, with $D_{img} = 512$, $D_{uv} = 64$ and $N_{iter} = 3$. We use the RAFT-L configuration for the update module and keep its hyperparameters when possible [36]. We optimize the model for 6 epochs using the AdamW optimizer [27], an initial learning rate of 1×10^{-4} and a decay of 0.1 every 2 epochs. We use image augmentation such as random scaling, rotation, and color corruption [42], synthetic occlusions [39] and synthetic backgrounds (see Appendix D).

3D model fitting. To minimize the energy function and obtain tracking parameters, we use the AdamW optimizer with an initial learning rate of 1×10^{-2} and a automatic learning rate scheduler with a decay factor of 0.5 and patience of 30 steps, until convergence. We enable $\boldsymbol{\delta}_d$ only for multi-view reconstruction, and only for the nose region.

Baselines. We implement and test against the most recent publicly available methods for single image regression-based approaches 3DDFAv2 [19], SADRNet [32], PRNet [41], DECA (coarse) [14], EMOCA (coarse) [10], and HRN [24]. We extend the ability of these methods to use temporal priors by applying a simple temporal Gaussian filter to the screen-space vertices. We also include the popular photometric optimization-based approach MPT [57]. Lastly, we compare against the key-point-only optimization-based method *Dense* proposed by [42] on public benchmarks.

4.1. Multiface Benchmark

We divide our Multiface benchmark into two categories: Without temporal information sharing, where each method is restricted to operate on single images, and with (both forward and backward) temporal information sharing, where each method is allowed to use the entire sequence as observations. Our method significantly outperforms the best publicly available method by 54% w.r.t. face-region $\overline{SSME}$ on both on single-image and by 46% on sequence prediction. This confirms the superior 2D alignment accuracy of our method. Despite using only 2D alignment as supervision, our method performs 8% better in terms of 3D reconstruction (CD) than the photometric optimization approach MPT [57] (see Tab. 2. To our surprise, MPT performs inferior w.r.t. motion error than some regression-based models — this is likely due to uniform lighting and texture in the Multiface dataset. Qualitative results Fig. 5 confirm that methods using photometric errors (DECA, HRN, MPT) perform inferior w.r.t. screen space motion in areas without key-point supervision such as cheeks and forehead. Plotting the $SSME_h$ over different time windows h (see Fig. 4) gives a previously unseen overview of temporal stability. Regression-based methods suffer from high short-term error ($SSME_1$) which is due to temporal instability and jitter. As expected, introducing temporal smoothing improves this issue and the overall $\overline{SSME}$ for these methods. Our method achieves very low short-term SSME even with single image prediction, which indicates the high robustness and accuracy of the alignment network. As expected, introducing temporal priors reduces $\overline{SSME}$.

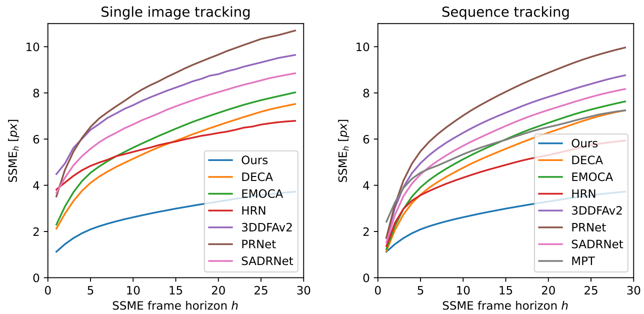

Figure 4. $SSME_h$ plotted over all frame horizons for each evaluated tracker for single-image and full sequence tracking (right). Lower $SSME_h$ in smaller frame horizons h (left in the graph) means short-term temporal stability while lower $SSME_h$ in larger frame horizons (right in the graph) means better long-term tracking consistency. Our tracker performs significantly better over every time horizon.

4.2. FaceScape Benchmark

Method	CD ↓ (mm)	NME ↓ (rad)
MGCNet [35]	4.00	0.093
PRNet [41]	3.56	0.126
SADRNet [32]	6.75	0.133
DECA [14]	4.69	0.108
3DDFAv2 [19]	3.60	0.096
HRN [24]	3.67	0.087
Ours	**2.21**	**0.083**

Table 1. Results on the FaceScape benchmark [47].

We also compare our method on the FaceScape benchmark [47], which measures 3D reconstruction accuracy from 2D images under large view (up to 90°) and expression variations. On this benchmark, we outperform the best previous regression-based methods by 38% in terms of CD and 4.6% in terms of mean normal error (NME) Tab. 1. This shows that our method can accurately reconstruct faces even

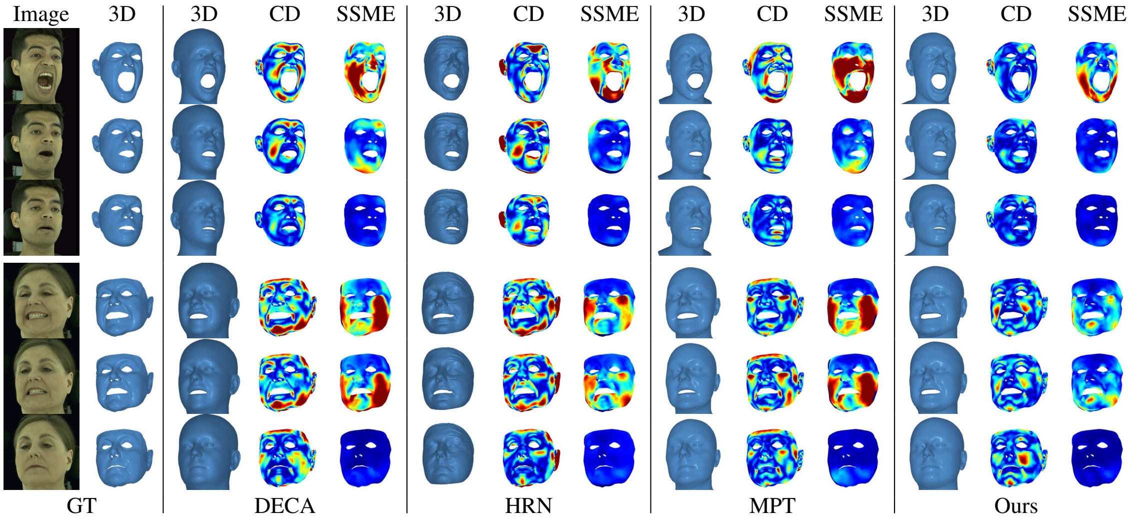

Figure 5. Qualitative results on two sequences (top and bottom 3 rows) of our Multiface benchmark. Warmer colors represent high error, while colder colors represent low error. DECA [14], HRN [24], and MPT [57] struggle with motion in the cheek and forehead region, which is visible in the SSME error plot (right columns). Despite using only 2D alignment as supervision, our method achieves a better 3D reconstruction (CD) (center columns).

	No temporal information sharing (single image)										**With temporal information sharing (sequence)**									
Method	CD (mm) ↓					$\overline{\text{SSME}}$ (px) ↓					CD (mm) ↓					$\overline{\text{SSME}}$ (px) ↓				
	face	mouth	nose	eyes	ears	face	mouth	nose	eyes	ears	face	mouth	nose	eyes	ears	face	mouth	nose	eyes	ears
DECA [14]	1.37	1.29	1.32	1.08	2.68	5.66	6.16	3.60	4.25	8.34	1.37	1.29	1.32	1.08	2.68	5.26	6.12	3.22	3.87	7.10
EMOCA [10]	1.47	1.46	1.49	1.10	2.71	6.14	7.32	3.99	4.26	8.55	1.47	1.46	1.49	1.10	2.71	5.63	6.95	3.56	3.87	7.28
HRN [24]	1.49	1.39	1.24	1.09	-	5.75	6.04	4.20	4.84	-	1.49	1.39	1.24	1.09	-	4.63	5.39	3.02	3.68	-
3DDFAv2 [19]	1.53	1.52	1.59	1.24	-	7.91	9.47	6.65	6.55	-	1.53	1.52	1.59	1.24	-	6.71	8.43	5.43	5.44	-
PRNet [41]	1.55	1.59	1.50	1.28	-	8.45	10.66	5.98	6.03	-	1.55	1.59	1.50	1.28	-	7.54	9.80	5.25	5.35	-
SADRNet [32]	1.49	1.52	1.49	1.22	-	7.11	8.21	5.15	5.53	-	1.49	1.52	1.49	1.22	-	6.18	7.46	4.31	4.72	-
MPT [57]	-	-	-	-	-	-	-	-	-	-	1.30	1.47	1.11	0.96	-	5.74	7.34	4.64	4.01	-
Ours	**1.20**	**1.3**	**1.05**	**0.97**	**2.34**	**2.58**	**3.14**	**1.33**	**2.07**	**1.72**	**1.19**	**1.31**	**1.04**	**0.96**	**2.34**	**2.50**	**3.16**	**1.27**	**2.03**	**1.68**

Table 2. Results on our Multiface tracking benchmark with and without temporal information sharing. Our method consistently outperforms previous methods on every single category, metric and face region.

	Single-view			**Multi-view**		
Method	Error (mm) ↓			Error (mm) ↓		
	Median	Mean	Std	Median	Mean	Std
MGCNet [35]	1.31	1.87	2.63	-	-	-
PRNet [41]	1.50	1.98	1.88	-	-	-
DECA [14]	1.09	1.38	1.18	-	-	-
Deep3D [12]	1.11	1.41	1.21	1.08	1.35	1.15
Dense [42]	1.02	1.28	1.08	0.81	1.01	0.84
MICA [57]	0.90	1.11	0.92	-	-	-
TokenFace [38]	**0.76**	**0.95**	**0.82**	-	-	-
Ours	0.87	1.07	0.88	**0.71**	**0.88**	**0.73**

Table 3. Results on the NoW Challenge [34]. Multi-view evaluation is done as in [42]. Multi-view results for [12] and [42] are reported by [42].

under large view deviations.

4.3. Now Challenge

The NoW benchmark is a public benchmark for evaluating neutral head reconstruction from 2D images captured indoors and outdoors, with different expressions, and under variations in lighting conditions and occlusions. We evaluate our method on the non-metrical challenge (Tab. 3). For single-view reconstruction, our model outperforms our neutral shape predictor MICA [57] by 4% on mean scan-to-mesh distance. For the multi-view case, we outperform the baseline *Dense* [42] by 13%, likely due to our method's high 2D alignment accuracy, better neutral shape priors, and per-vertex deformations. TokenFace [38] performs better for the single-view case, however, their predictions could be integrated into our pipeline since they use the FLAME topology. Importantly, our network is able to generalize to these in-the-wild images despite being trained only on in-the-lab data captured under controlled lighting conditions. An important sub-task for 3D face trackers is to disentangle the identity and expression components of the face shape. The outstanding results on the NoW benchmark indicate the ability of our tracker to accomplish this.

4.4. Downstream Tasks

In the following, we show how we enhance downstream models using our face tracker.

3D Head Avatar Synthesis. Recent head avatar synthesis methods heavily rely on photometric head trackers to generate face alignment priors [17, 53, 56]. INSTA [56], a top-performing model, uses MPT [57]. We modify INSTA by replacing their tracker with ours. We compare our enhanced FlowFace-INSTA to the baseline MPT-INSTA. On their publicly available dataset, we outperform MPT-INSTA by 10.5% on perceptual visual fidelity (LPIPS). On our Multiface benchmark videos, we outperform MPT-INSTA by 20.3% on LPIPS. Detailed results can be viewed in Appendix G. These results demonstrate how better face trackers can directly improve performance on down-stream tasks which highlights the importance of our research.

Speech-driven 3D facial animation. The field of speech-driven facial animation often suffers from data sparsity [9, 13, 46]. To alleviate this issue, we generate 3D face meshes using the multi-view video dataset MEAD [40]. In using this generated dataset to augment the training of the state-of-the-art model CodeTalker [46] (see Appendix H), we are able to improve from a lip vertex error of 3.13×10^{-5} to 2.85×10^{-5} on the VOCASET benchmark [9], an 8.8% improvement. This underlines the benefit of high-accuracy video face trackers for large-scale data generation.

4.5. 2D Alignment

To show the benefit of our 2D alignment model architecture, we conduct an evaluation on our validation set, which consists of 84 subjects of our dataset. We implement the dense landmark model of [42] (ResNet-101 backbone) and adapt it to output FLAME vertex alignment and uncertainty. We also implement PRNet [41] and modify it in the same way. We retrain each method on our training set. In evaluate the 2D alignment accuracy with respect to normalized mean error (NME) of every vertex in the face area (Fig. 14, green vertices). With an NME of 1.30, our method performs significantly better than the ResNet architecture of *Dense* [42] (NME $= 1.63$), and PRNet (NME $= 2.52$). We note that the accuracy of uncertainty cannot be evaluated with NME. A qualitative comparison can be viewed in Fig. 17.

4.6. Ablation Studies

2D alignment network. To analyze the effect of different feature encoder backbones, we replace our backbone with different variations of the Segformer model and also test the CNN-based backbone BiSeNet-v2 [49] (see Tab. 4). As expected, vision-transformer-based networks show better performance. Experimenting with the number of iterations N_{iter} for the update module, we find that multiple iterations instead of one improves the performance. Finally, we confirm the superior performance of our 2D alignment network compared to the ResNet-101-based network of [42] mentioned in Sec. 4.5.

Backbone	N_{iter}	#Param	latency (ms)	CD↓	$\overline{SSME}$↓
ResNet-101	—	73.4M	9	1.54	3.90
BiSeNet-v2	3	17.6M	23	1.21	3.52
MiT-b1	3	17.3M	29	1.22	3.21
MiT-b2	3	31.0M	46	1.20	2.78
MiT-b5	1	88.2M	66	1.25	2.70
MiT-b5	2	88.2M	71	1.21	2.61
MiT-b5	3	88.2M	75	1.18	2.58
MiT-b5	4	88.2M	80	1.23	2.62

Table 4. Ablations for backbone architectures and hyper-parameters of the 2D alignment network on our Multiface benchmark. Latency is evaluated on a *Quadro RTX 5000* GPU.

3D model fitting. We show in Tab. 5 the benefit of integrating the MICA neutral shape prediction on the NoW Challenge validation set. The significant performance gain on single-image predictions shows that our 3D tracking pipeline can integrate MICA predictions very well, even improving them. We also show the benefit of predicting a dense face alignment in conjunction with per-vertex deformations in multi-view settings. This shows that our 2D alignment is precise enough to predict face shapes that lie outside of the FLAME blend-shape space, which previous optimization-based methods [42, 57] cannot achieve. For a qualitative analysis, see Appendix E.

	Single-view			**Multi-view**		
	Error (mm) ↓			Error (mm) ↓		
Method	Median	Mean	Std	Median	Mean	Std
Ours w/o MICA	0.99	1.23	1.03	0.71	0.88	0.76
MICA only	0.91	1.13	0.94	-	-	-
Ours w/o δ_d	-	-	-	0.68	0.84	0.72
Ours	0.82	1.02	0.85	0.67	0.83	0.71

Table 5. Ablations for the 3D model fitting module on single and multi-view reconstruction on the NoW validation set.

5. Conclusion and Future Work

This paper presents a state-of-the-art face tracking pipeline with a highly robust and accurate 2D alignment module. Its performance is thoroughly validated on a variety of benchmarks and downstream tasks. However, the proposed two-stage pipeline is not fully differentiable, which prevents end-to-end learning. Furthermore, our training data is limited to data captured in-the-lab. In future work, we intend to extend the alignment network to directly predict depth as well, obviating the need for the 3D model fitting step. Synthetic datasets [42] could alleviate the data issue.

We're confident that our tracker will accelerate research in downstream tasks by generating large-scale face capture data using readily available video datasets [8, 29, 50]. We also believe that our novel motion capture evaluation benchmark will focus and align future research efforts to create even more accurate methods.

References

[1] Stirling/esrc 3d face database. https://pics.stir.ac.uk/ESRC/. Accessed: 2023-10-25. 5, 2, 4

[2] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, page 187–194, USA, 1999. ACM Press/Addison-Wesley Publishing Co. 1, 2

[3] Timo Bolkart, Tianye Li, and Michael J. Black. Instant multi-view head capture through learnable registration. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 768–779, 2023. 6, 2

[4] Adrian Bulat and Georgios Tzimiropoulos. How far are we from solving the 2d & 3d face alignment problem? (and a dataset of 230,000 3d facial landmarks). In *International Conference on Computer Vision*, 2017. 2

[5] Chen Cao, Menglei Chai, Oliver Woodford, and Linjie Luo. Stabilized real-time face tracking via a learned dynamic rigidity prior. *ACM Trans. Graph.*, 37(6), 2018. 2

[6] Zenghao Chai, Haoxian Zhang, Jing Ren, Di Kang, Zhengzhuo Xu, Xuefei Zhe, Chun Yuan, and Linchao Bao. Realy: Rethinking the evaluation of 3d face reconstruction, 2022. 2, 5

[7] Zenghao Chai, Tianke Zhang, Tianyu He, Xu Tan, Tadas Baltrušaitis, HsiangTao Wu, Runnan Li, Sheng Zhao, Chun Yuan, and Jiang Bian. Hiface: High-fidelity 3d face reconstruction by learning static and dynamic details, 2023. 2

[8] J. S. Chung, A. Nagrani, and A. Zisserman. Voxceleb2: Deep speaker recognition. In *INTERSPEECH*, 2018. 8

[9] Daniel Cudeiro, Timo Bolkart, Cassidy Laidlaw, Anurag Ranjan, and Michael Black. Capture, learning, and synthesis of 3D speaking styles. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 10101–10111, 2019. 8, 7

[10] Radek Danecek, Michael J. Black, and Timo Bolkart. Emoca: Emotion driven monocular face capture and animation, 2022. 1, 2, 4, 6, 7

[11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 6, 1

[12] Yu Deng, Jiaolong Yang, Sicheng Xu, Dong Chen, Yunde Jia, and Xin Tong. Accurate 3d face reconstruction with weakly-supervised learning: From single image to image set. In *IEEE Computer Vision and Pattern Recognition Workshops*, 2019. 2, 7

[13] Yingruo Fan, Zhaojiang Lin, Jun Saito, Wenping Wang, and Taku Komura. Faceformer: Speech-driven 3d facial animation with transformers. *arXiv preprint arXiv:2112.05329*, 2021. 8

[14] Yao Feng, Haiwen Feng, Michael J. Black, and Timo Bolkart. Learning an animatable detailed 3d face model from in-the-wild images. *CoRR*, abs/2012.04012, 2020. 1, 4, 6, 7, 10, 11

[15] Pablo Garrido, Michael Zollhöfer, Dan Casas, Levi Valgaerts, Kiran Varanasi, Patrick Pérez, and Christian Theobalt. Reconstruction of personalized 3d face rigs from monocular video. *ACM Trans. Graph.*, 35(3), 2016. 2

[16] Pablo Garrido, Michael Zollhöfer, Chenglei Wu, Derek Bradley, Patrick Pérez, Thabo Beeler, and Christian Theobalt. Corrective 3d reconstruction of lips from monocular video. *ACM Trans. Graph.*, 35(6), 2016. 2

[17] Philip-William Grassal, Malte Prinzler, Titus Leistner, Carsten Rother, Matthias Nießner, and Justus Thies. Neural head avatars from monocular rgb videos. *arXiv preprint arXiv:2112.01554*, 2021. 8

[18] Ivan Grishchenko, Artsiom Ablavatski, Yury Kartynnik, Karthik Raveendran, and Matthias Grundmann. Attention mesh: High-fidelity face mesh prediction in real-time. *CoRR*, abs/2006.10962, 2020. 2

[19] Jianzhu Guo, Xiangyu Zhu, Yang Yang, Yang Fan, Zhen Lei, and Stan Li. *Towards Fast, Accurate and Stable 3D Dense Face Alignment*, pages 152–168. 2020. 1, 2, 6, 7, 10, 11

[20] Rıza Alp Güler, George Trigeorgis, Epameinondas Antonakos, Patrick Snape, Stefanos Zafeiriou, and Iasonas Kokkinos. Densereg: Fully convolutional dense shape regression in-the-wild, 2017. 2

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. 2

[22] *A 3D Face Model for Pose and Illumination Invariant Face Recognition*, Genova, Italy, 2009. IEEE. 2

[23] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. 7

[24] Biwen Lei, Jianqiang Ren, Mengyang Feng, Miaomiao Cui, and Xuansong Xie. A hierarchical representation network for accurate and detailed face reconstruction from in-the-wild images, 2023. 1, 2, 6, 7, 10, 11

[25] J. P. Lewis, Matt Cordner, and Nickson Fong. Pose space deformation: A unified approach to shape interpolation and skeleton-driven deformation. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, page 165–172, USA, 2000. ACM Press/Addison-Wesley Publishing Co. 4

[26] Tianye Li, Timo Bolkart, Michael. J. Black, Hao Li, and Javier Romero. Learning a model of facial shape and expression from 4D scans. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 36(6):194:1–194:17, 2017. 2, 4, 6, 3, 5, 7

[27] Ilya Loshchilov and Frank Hutter. Fixing weight decay regularization in adam. *CoRR*, abs/1711.05101, 2017. 6

[28] Araceli Morales, Gemma Piella, and Federico M. Sukno. Survey on 3d face reconstruction from uncalibrated images. *CoRR*, abs/2011.05740, 2020. 2

[29] Arsha Nagrani, Joon Son Chung, Weidi Xie, and Andrew Zisserman. Voxceleb: Large-scale speaker verification in the wild. *Computer Science and Language*, 2019. 8

[30] Andrés Prados-Torreblanca, José M Buenaposada, and Luis Baumela. Shape preserving facial landmarks with graph attention networks. In *33rd British Machine Vision Conference 2022, BMVC 2022, London, UK, November 21-24, 2022*. BMVA Press, 2022. 2

[31] Aashish Rai, Hiresh Gupta, Ayush Pandey, Francisco Vicente Carrasco, Shingo Jason Takagi, Amaury Aubel, Daeil Kim, Aayush Prakash, and Fernando de la Torre. Towards realistic generative 3d face models, 2023. 2

[32] Zeyu Ruan, Changqing Zou, Longhai Wu, Gangshan Wu, and Limin Wang. SADRNet: Self-aligned dual face regression networks for robust 3d dense face alignment and reconstruction. *IEEE Transactions on Image Processing*, 30: 5793–5806, 2021. 2, 6, 7, 10, 11

[33] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks, 2019. 2

[34] Soubhik Sanyal, Timo Bolkart, Haiwen Feng, and Michael Black. Learning to regress 3d face shape and expression from an image without 3d supervision. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2, 5, 7, 3

[35] Jiaxiang Shang, Tianwei Shen, Shiwei Li, Lei Zhou, Mingmin Zhen, Tian Fang, and Long Quan. Self-supervised monocular 3d face reconstruction by occlusion-aware multi-view geometry consistency. *arXiv preprint arXiv:2007.12494*, 2020. 2, 6, 7

[36] Zachary Teed and Jia Deng. RAFT: recurrent all-pairs field transforms for optical flow. *CoRR*, abs/2003.12039, 2020. 3, 6, 1

[37] Justus Thies, Michael Zollhöfer, Marc Stamminger, Christian Theobalt, and Matthias Nießner. Face2face: Real-time face capture and reenactment of rgb videos, 2020. 1, 2

[38] Zhang Tianke, Chu Xuangeng, Liu Yunfei, Lin Lijian, Yang Zhendong, Xu Zhengzhuo, Cao Chengkun, Yu Fei, Zhou Changyin, Yuan Chun, and Yu Li. Accurate 3d face reconstruction with facial component tokens. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. 2, 7

[39] Kenny T. R. Voo, Liming Jiang, and Chen Change Loy. Delving into high-quality synthetic face occlusion segmentation datasets. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2022. 6, 3

[40] Kaisiyuan Wang, Qianyi Wu, Linsen Song, Zhuoqian Yang, Wayne Wu, Chen Qian, Ran He, Yu Qiao, and Chen Change Loy. Mead: A large-scale audio-visual dataset for emotional talking-face generation. In *ECCV*, 2020. 8, 6

[41] Yue Wang and Justin M. Solomon. Prnet: Self-supervised learning for partial-to-partial registration, 2019. 2, 6, 7, 8

[42] Erroll Wood, Tadas Baltrusaitis, Charlie Hewitt, Matthew Johnson, Jingjing Shen, Nikola Milosavljevic, Daniel Wilde, Stephan Garbin, Chirag Raman, Jamie Shotton, Toby Sharp, Ivan Stojiljkovic, Tom Cashman, and Julien Valentin. 3d face reconstruction with dense landmarks, 2022. 1, 2, 4, 6, 7, 8, 3, 5

[43] Chenglei Wu, Derek Bradley, Markus Gross, and Thabo Beeler. An anatomically-constrained local deformation model for monocular face capture. *ACM Trans. Graph.*, 35 (4), 2016. 2

[44] Cheng-hsin Wuu, Ningyuan Zheng, Scott Ardisson, Rohan Bali, Danielle Belko, Eric Brockmeyer, Lucas Evans, Timothy Godisart, Hyowon Ha, Xuhua Huang, Alexander Hypes, Taylor Koska, Steven Krenn, Stephen Lombardi, Xiaomin Luo, Kevyn McPhail, Laura Millerschoen, Michal Perdoch, Mark Pitts, Alexander Richard, Jason Saragih, Junko Saragih, Takaaki Shiratori, Tomas Simon, Matt Stewart, Autumn Trimble, Xinshuo Weng, David Whitewolf, Chenglei Wu, Shoou-I Yu, and Yaser Sheikh. Multiface: A dataset for neural face rendering. In *arXiv*, 2022. 1, 5, 2, 6

[45] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. In *Neural Information Processing Systems (NeurIPS)*, 2021. 3, 1

[46] Jinbo Xing, Menghan Xia, Yuechen Zhang, Xiaodong Cun, Jue Wang, and Tien-Tsin Wong. Codetalker: Speech-driven 3d facial animation with discrete motion prior, 2023. 8, 7

[47] Haotian Yang, Hao Zhu, Yanru Wang, Mingkai Huang, Qiu Shen, Ruigang Yang, and Xun Cao. Facescape: a large-scale high quality 3d face dataset and detailed riggable 3d face prediction, 2020. 2, 5, 6, 4

[48] Hongwei Yi, Hualin Liang, Yifei Liu, Qiong Cao, Yandong Wen, Timo Bolkart, Dacheng Tao, and Michael J. Black. Generating holistic 3d human motion from speech, 2023. 2

[49] Changqian Yu, Changxin Gao, FlowFace-INSTA to the baseline MPT-INSTA Jingbo Wang, Gang Yu, Chunhua Shen, and Nong Sang. Bisenet V2: bilateral network with guided aggregation for real-time semantic segmentation. *CoRR*, abs/2004.02147, 2020. 8, 5

[50] Jianhui Yu, Hao Zhu, Liming Jiang, Chen Change Loy, Weidong Cai, and Wayne Wu. CelebV-Text: A large-scale facial text-video dataset. In *CVPR*, 2023. 8

[51] Shifeng Zhang, Xiangyu Zhu, Zhen Lei, Hailin Shi, Xiaobo Wang, and Stan Z. Li. S^{3}fd: Single shot scale-invariant face detector, 2017. 3

[52] Yufeng Zheng, Victoria Fernández Abrevaya, Xu Chen, Marcel C. Bühler, Michael J. Black, and Otmar Hilliges. I M avatar: Implicit morphable head avatars from videos. *CoRR*, abs/2112.07471, 2021. 2

[53] Yufeng Zheng, Wang Yifan, Gordon Wetzstein, Michael J. Black, and Otmar Hilliges. Pointavatar: Deformable point-based head avatars from videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 8

[54] Zhenglin Zhou, Huaxia Li, Hong Liu, Nanyang Wang, Gang Yu, and Rongrong Ji. Star loss: Reducing semantic ambiguity in facial landmark detection, 2023. 2

[55] Xiangyu Zhu, Zhen Lei, Xiaoming Liu, Hailin Shi, and Stan Z. Li. Face alignment across large poses: A 3d solution. *CoRR*, abs/1511.07212, 2015. 2

[56] Wojciech Zielonka, Timo Bolkart, and Justus Thies. Instant volumetric head avatars. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4574–4584, 2022. 8, 4, 6

[57] Wojciech Zielonka, Timo Bolkart, and Justus Thies. Towards metrical reconstruction of human faces, 2022. 1, 2, 4, 5, 6, 7, 8

[58] Michael Zollhöfer, Justus Thies, Darek Bradley, Pablo Garrido, Thabo Beeler, Patrick Péerez, Marc Stamminger,

Matthias Nießner, and Christian Theobalt. State of the art on monocular 3d face reconstruction, tracking, and applications. 2018. 1, 2