

REFA: Real-time Egocentric Facial Animations for Virtual Reality

Qiang Zhang Tong Xiao Haroun Habeeb Larissa Laich Sofien Bouaziz
 Patrick Snape Wenjing Zhang Matthew Cioffi Peizhao Zhang Pavel Pidlypenskyi
 Winnie Lin Luming Ma Mengjiao Wang Kunpeng Li Chengjiang Long
 Steven Song Martin Prazak Alexander Sjöholm Ajinkya Deogade Jaebong Lee
 Julio Delgado Mangas Amaury Aubel

Reality Labs at Meta, Menlo Park, USA

Abstract

We present a novel system for real-time tracking of facial expressions using egocentric views captured from a set of infrared cameras embedded in a virtual reality (VR) headset. Our technology facilitates any user to accurately drive the facial expressions of virtual characters in a non-intrusive manner and without the need of a lengthy calibration step. At the core of our system is a distillation based approach to train a machine learning model on heterogeneous data and labels coming from multiple sources, e.g. synthetic and real images. As part of our dataset, we collected 18k diverse subjects using a lightweight capture setup consisting of a mobile phone and a custom VR headset with extra cameras. To process this data, we developed a robust differentiable rendering pipeline enabling us to automatically extract facial expression labels. Our system opens up new avenues for communication and expression in virtual environments, with applications in video conferencing, gaming, entertainment, and remote collaboration.

1. Introduction

Virtual reality (VR) transports users into simulated environments mimicking or enhancing real-world experiences. To achieve this, a head mounted display (HMD) presents three dimensional environments to users, along with other sensory feedback such as sound. Through these immersive experiences, users can interact with and explore computer-generated environments in a realistic manner. For example, users can tour a virtual museum, navigate through a digital city, or play a video games in VR.

One particular important key feature in VR is the sense of *social presence*, *i.e.* people feeling that they are meaningfully interacting with others. This ability of feeling present with other people and form or deepen social connections is what makes VR truly engaging. However, to facilitate

natural and intuitive social interactions, the development of accurate motion tracking technologies reproducing users' motions in real-time are required.

In particular, tracking facial motions is a key technology for social presence. This is achieved by capturing real-time video data of a person's face using cameras and then tracking specific features such as the mouth, nose, and eyes. By monitoring the movements of these features over time, face tracking detects and tracks facial expressions, such as smiles, frowns, and eyebrow raises. This signal can then be used to drive actions in a virtual environments, like ① the challenge posed by occlusion of the user's face by the HMD, which makes it difficult to obtain an unobstructed capture of the face, ② the complexity and cost of adding extra cameras to VR devices, and ③ the compute restrictions inherent to mobile platforms.

Contributions. In this paper, we present an innovative system enabling accurate tracking of a user's facial expressions and movements using infrared (IR) cameras directly embedded within an HMD. Our contributions are ① the placement of IR cameras and LEDs on an HMD through simulation, ② an automated ground-truth generation pipeline allowing the collection of a large dataset using a lightweight capture process, ③ an iterative distillation framework allowing to train our machine learning (ML) model with heterogeneous and noisy labels acquired from different sources, and ④ an end-to-end system with auto-calibration and automated failure detection.

2. Related Work

The animation of digital characters through facial performance capture is a widely used technique within the computer graphics industry and has been a subject of ongoing research for many years. Pioneer works like Active Appearance Models (AAMs) [8, 11, 32, 42] and 3D morphable models (3DMM) [3, 4, 28] have been widely effective at registering faces in images by optimizing low-dimensional

coefficients of linear subspaces for both shape and appearance.

While AAM and 3DMM techniques effectively capture facial information within a linear model, the industry has predominantly embraced blendshapes subspaces [29]. This preference arises from the semantic nature of blendshapes, enabling more meaningful and intuitive manipulation of facial expressions. This representation has been adopted with success by a large number of recent face tracking techniques [5, 39, 46], including consumer products such as Apple’s ARKit and Meta Spark.

Beyond optimization based techniques, recent approaches [18, 23, 37, 41, 49] have leveraged deep learning techniques to regress low-dimensional coefficients from facial images. These methods typically train a convolutional neural network using large datasets to estimate the face shape, such as blendshape weights, from the input image.

The accuracy limitations inherent to low-dimensional linear subspaces have led to the development of alternative approaches that directly generate detailed face shape and appearance as meshes and textures. To achieve high-quality results, some methods rely on a multi-camera rig [10, 47] for capturing face shape and appearance. Other works focus on generating face shape and appearance from consumer devices, such as from RGB images [13, 15, 17, 22, 40], and from RGBD data [2, 6, 50].

Another related topic involves generating face images from various inputs, such as text [7, 16], audio [9, 24, 35, 36], or another face image (namely deepfake or faceswap) [34]. These techniques typically employ generative adversarial networks (GANs) to synthesize facial images. Additionally, some recent approaches utilize methods like Neural Radiance Fields (NeRF) [1] or stable diffusion [26] for face image generation. These advancements have paved the way for generating face images from different modalities, enabling applications in text-to-face, audio-to-face, and image-to-face synthesis.

The most relevant work to ours is [31, 45] which leverages an auto-encoder to drive avatars from cameras mounted on a VR headset. This model is trained using data captured from a large multi-camera rig, allowing for high-fidelity social interaction in virtual reality. Instead of training the model to decode both geometry and appearance, our model only decodes the geometry in blendshape format. This not only reduces computational costs but also allows for driving avatars of different appearances or styles, which means that 3rd party developers can use their own rigs with our model. Furthermore, we simplify the multi-camera rig used to train the model to a phone capture, which is available off the shelf. This enables us to collect a large and diverse dataset and train a model that can generalize to a broader population.

3. Overview

Our system utilizes 5 cameras that are integrated within a HMD (see Sec. 4). These cameras capture infrared images at a resolution of 400x400 and a frame rate of 30 Hz. Our goal is accurately predict facial expressions from these camera images in real-time using an ML model. As a representation for the facial expression we choose 3D blendshape coefficients which can be utilised to animate digital avatars. These 3D blendshape models (see Sec. 5) are a compact representation widely-used in previous work [3] and in the industry. To train our ML model we rely on three heterogeneous sources of data: ① real, ② synthetic, and ③ artist driven, each providing unique benefits (see Sec. 6). To train our on-device model using these datasets gathered from different domains we employ an iterative distillation process (see Sec. 7). To make our system robust to in the wild usage, we implement other key features as part of our end-to-end system design such as an online calibration step as well as a failure detection mechanism. Finally, we provide an extensive set of qualitative and quantitative evaluations to showcase the effectiveness of our approach (see Sec. 8).

4. Hardware Design

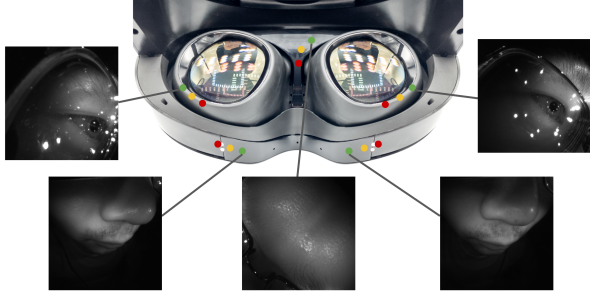
We install 5 infrared cameras on our HMD, which are capable of capturing images at a resolution of 400×400 and a frame rate of 30 Hz. As shown in Fig. 1, two cameras are assigned to track eye and eyebrow movements, two are responsible for monitoring mouth movements, and the other one camera is designated to capture images of the glabella area, *i.e.*, the region between the eyebrows. To cater to consumer use, the sensors are integrated within the hardware form factor, making it challenging to obtain a set of cameras with a clear view of the face. To resolve this issue, we used a PCA model generated from 800 facial scans of 30 expressions [43], which enabled us to assess different configurations for a wide range of facial structures and determine an adequate camera placement. To measure the quality of the different configurations, we measure the following metrics:

Visibility \mathcal{V} : This metric quantifies the visibility of key facial regions \mathcal{R} . For a camera \mathcal{C} , the visibility of a region is determined by the cosine of the angle between the surface normal \mathbf{n} of the face region and the optical axis of the camera \mathbf{o}

$$\mathcal{V}_{\mathcal{C}} = \frac{1}{|\mathcal{R}|} \sum_{r \in \mathcal{R}} \mathbf{n}_r^T \mathbf{o}_{\mathcal{C}}. \quad (1)$$

The weight ranges between 0 and 1, where a higher value indicates better visibility.

Range of Motion \mathcal{M} : This metric assesses the ease with which a key set of facial expression \mathcal{E} can be captured by the camera \mathcal{C} . It is determined by projecting the facial keypoints \mathbf{K} onto the 2D image space (denoted by $\mathcal{P}(\cdot)$) for both neutral and another facial expression $e \in \mathcal{E}$, calculating the ℓ_2



Eye	Green	Orange	Red
Visibility \mathcal{V}	0.508	0.494	0.473
Range of Motion \mathcal{M}	5.184	3.782	4.203
Mouth	Green	Orange	Red
Visibility \mathcal{V}	0.213	0.133	0.104
Range of Motion \mathcal{M}	12.906	7.349	7.349
Glabella	Green	Orange	Red
Visibility \mathcal{V}	0.268	0.351	0.361
Range of Motion \mathcal{M}	9.201	8.143	6.274

Figure 1. Our HMD is equipped with five face cameras, two for eye and eyebrow regions, two for mouth, and one for glabella. Note that we mirror the left eye and left mouth images. A multitude of camera configurations have been considered during the design of the HMD. Among these configurations, the one highlighted in green has been implemented, which has better visibility and range of motion metrics than the configurations highlighted in orange or red (Orange or red configurations seems to have better visibility in glabella, but they pose conflicts with users’ glasses frames and the HMD’s Inter-pupil distance adjustment mechanism).

distance between these points, and then averaging across all the expressions

$$\mathcal{M}_C = \frac{1}{|\mathcal{E}|} \sum_{e \in \mathcal{E}} \|\mathcal{P}(\mathbf{K}_e) - \mathcal{P}(\mathbf{K}_{neutral})\|_F. \quad (2)$$

The units of measurement are pixels, and a higher value indicates a greater range of motion, which is desirable.

We also introduced head pose variations when computing these metrics so as to make our camera placement robust to donning preferences. By simulating and incorporating these variations, we aimed to ensure that the sensors perform reliably and accurately regardless of how the headset is donned by the user. Figure 1 shows a few considered camera locations and their corresponding metrics.

5. 3D Face Representation

A central component of our system is a blendshape model [3] that provides a low-dimensional representation of the user’s expression space based on Ekman’s Facial Action Coding System (FACS) [12]. Our blendshape model

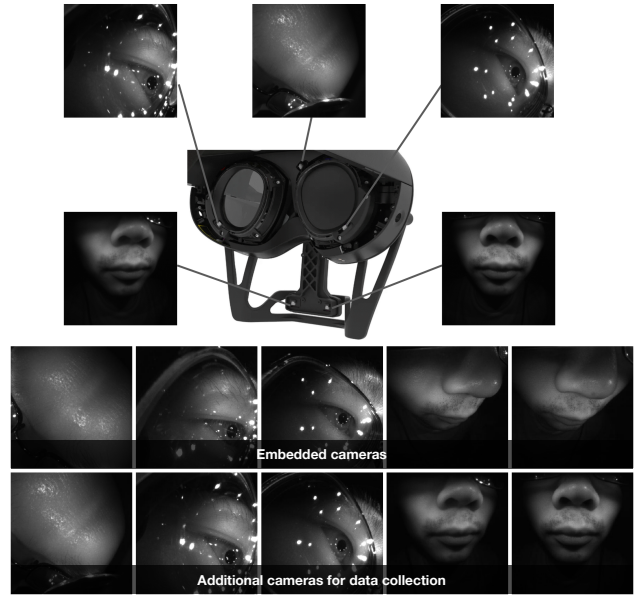


Figure 2. Our data collection HMD is equipped with additional five cameras, offering better visibility of the face than the embedded cameras. This camera setup allows us to improve the quality of the generated pseudo ground truth.

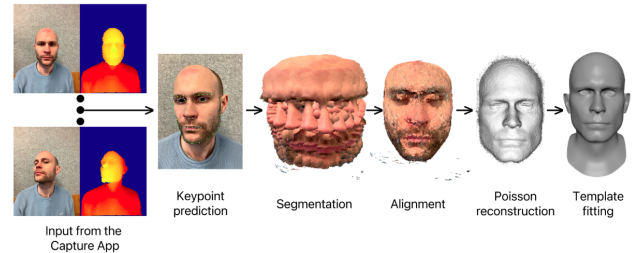


Figure 3. Our expression fitting pipeline takes RGBD frames as input and proceeds through a series of steps to generate a fitted mesh as output. We use this process to fit a set of facial expressions, from which we then create a subject-specific rig using example-based facial rigging [30].

contains 53 bases, which correspond to 3D meshes that can be combined linearly to produce new facial expressions. To combine these bases, we use a weight vector of blendshape coefficients $\mathbf{b} \in \mathbb{R}^{53}$, where each entry falls within the range of $[0.0, 1.0]$. A weight of 0.0 indicates an inactive expression, while a weight of 1.0 signifies full activation and is the maximum movement a person can perform. Our model also incorporates eye gaze vectors $\mathbf{g}_l \in \mathbb{R}^2$ and $\mathbf{g}_r \in \mathbb{R}^2$ for the left and right eyes, respectively. These vectors are also used to device an additional set of 8 eye following blendshapes. Additionally, the face’s rigid motion is parameterized using a translation vector $\mathbf{t} \in \mathbb{R}^3$ and a rotation matrix $\mathbf{R} \in \text{SO}(3)$.

6. Data Generation

To train our system we require a large dataset of camera frames labeled with blendshape coefficients. We collect a real-world dataset of 18k subjects providing 3 trillion frames. To annotate this extremely large number of frames, we develop an automated self-supervised approach based on a differentiable renderer (see Sec. 6.1). Further, a set of technical artists annotated a key set of frames manually to provide semantic labels (see Sec. 6.2). Finally, we complement our real-world dataset using synthetic data providing exact labels for challenging or long-tail cases (such as facial hair, glasses, donning variation, *etc.*, see Sec. 6.3).

6.1. Real Data

6.1.1 Capture Process

To collect a large dataset of a diverse set of subjects we develop a lightweight capture setup based on a mobile phone containing a depth sensor (iPhone 12) as well as a modified HMD. Capturing accurate views of the face presents a fundamental challenge with our embedded camera setup, primarily due to the close placement of the cameras, resulting in occlusions that obstruct the field of view. To alleviate this issues, we developed a data collection HMD equipped with an additional five ground truth cameras with better visibility including two boom cameras capturing the lower face from a frontal angle, two eye ground truth cameras, and one glabella ground truth camera (see Fig. 2). The inclusion of these additional ground truth cameras, in conjunction to the five embedded ones, offers alternative view angles significantly enhancing the visibility of the face (see Fig. 2).

Mobile phone capture. We utilize the mobile phone to gather a diverse set of 60 individual facial expression scans. Subjects are instructed to hold specific facial expressions while making slight head movements in front of the mobile phone, enabling us to collect RGBD frames from multiple angles.

HMD capture. Using our modified HMD, we request subjects to engage in a series of facial motions. This allows us to capture approximately 40 minutes of motion sequences encompassing a diverse range of expressions and speech sequences.

6.1.2 Generating subject-specific blendshape rig from mobile phone data

The process of generating facial blendshapes involves a multi-step optimization problem, performed individually for each subject. For each captured expressions, we first predict 100 facial keypoints [33] per frame, and extract a segmentation mask for the face [20]. We then align the RGBD frames using rigid ICP [6] and merge the result using Poisson reconstruction [25], obtaining a reasonable in-

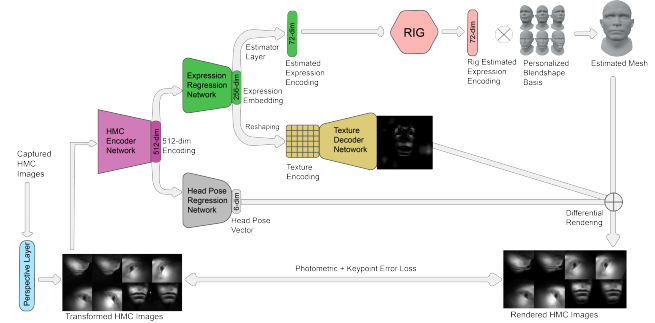


Figure 4. System diagram of estimating blendshape coefficients from HMD images, based on the subject-specific blendshape rig.

tegrated mesh (see Fig. 3). In order to maintain topological consistency across different subjects’ meshes, we employ a PCA model [43] that is fitted in conjunction with head pose estimation. This is followed by a refinement step using a Laplacian non-rigid deformation technique [6]. Finally, we compute the personalized blendshape rig using example-based facial rigging [30].

6.1.3 Estimating blendshape coefficients from HMD images

Given a subject-specific blendshape rig, we solve a “self-supervised” learning problem per subject to establish correspondences between input HMD images and output blendshape coefficients. We parameterize our problem with a Convolutional Neural Network (CNN) $\mathcal{N}_\theta : (\mathbf{I}) \rightarrow (\mathbf{b}, \mathbf{R}, \mathbf{t}, \mathbf{T})$, with the goal of predicting per frame blendshape coefficients \mathbf{b} , head pose (\mathbf{R}, \mathbf{t}) and texture \mathbf{T} from the input images \mathbf{I} , which has been proven beneficial [44]. By leveraging the blendshape coefficients and head pose information, we are able to reconstruct the mesh. This reconstructed mesh, along with the corresponding texture, can then be rasterized to reconstruct the input images. Our rasterizer is differentiable [44] and we optimize for the network’s weights θ using Adam [27]. We use three losses, ① a keypoint ℓ_2 reprojection error, ② the ℓ_2 pixel differences between the input and reconstructed views, and ③ a ℓ_1 sparsity regularization of the blendshape coefficients. See Fig. 4 for a detailed architecture of our approach.

The source of our blendshape bases are artist-provided sculpted meshes, with each mesh corresponding to a FACS shape [12]. These bases do not form a set of independent vectors and in some cases the same mesh can be generated using different blendshape coefficients. The ℓ_1 sparsity regularization used during the network optimization helps to regularize this problem but does not fully solve it. To further improve, we add a set of semantic “rig constraints” during the optimization.

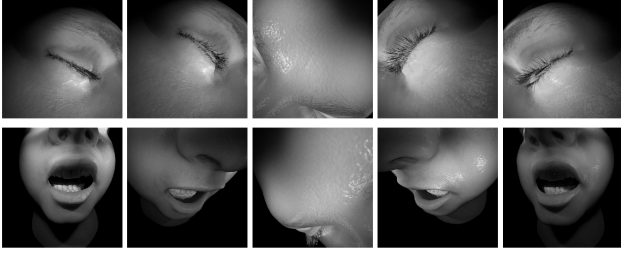


Figure 5. Example of a synthetic frame used to train our ML model

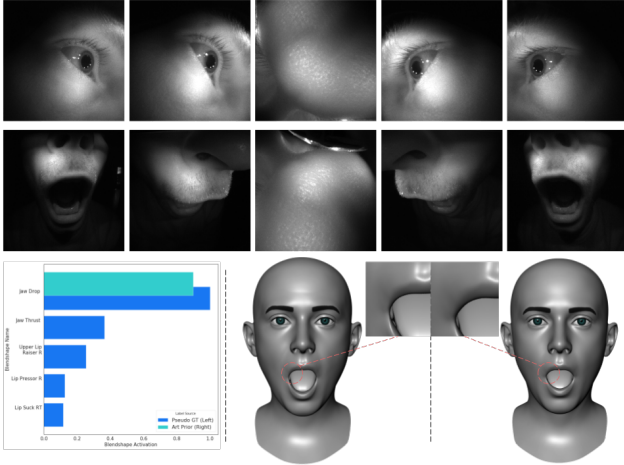


Figure 6. An example frame for the peak “Jaw Drop Max” expression. The left avatar is the pseudo ground truth generated based on the method described in Sec. 6.1.3. The right avatar represents the art priors. The histogram in the bottom right displays top-5 activated blendshapes for this frame. Note that the art priors are sparse compared with the pseudo ground truth. While the art priors are not perfectly accurate, they can be semantically meaningful for peak expression frames.

6.2. Art Priors

In our data collection, we include multiple segments where participants begin from a neutral pose, performing straightforward expressions by following a reference photo/video prompt that helps the person to mimic, and then revert to the neutral position. For each of these peak expressions, we ask FACS experts and skilled artists to define a set of expected blendshape coefficient activations, as demonstrated in Fig. 6. While not perfectly accurate, these labels can be considered as art priors, which are used later in training (Sec. 7.2) and evaluating (Sec. 8) the on-device ML models.

6.3. Synthetic Data

We generate a large synthetic dataset of roughly 25 million frames from 800 identities collected with a multi-view capture system and rigged with the methodology described in Sec. 6.1.2 (see Fig. 5). We retarget animation sequences

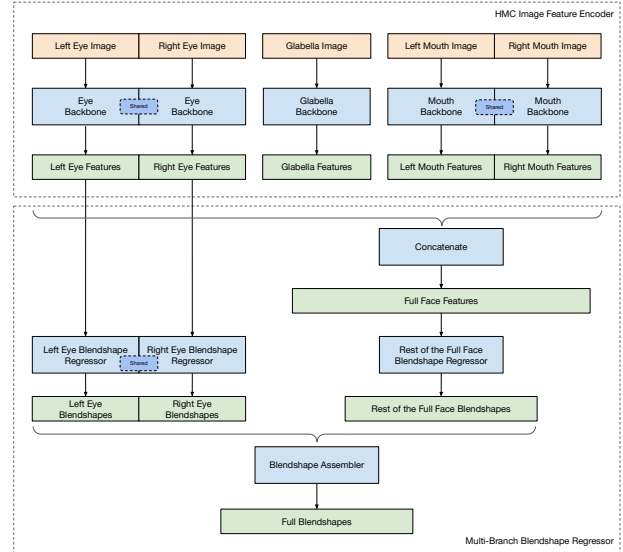


Figure 7. Architecture of the on-device ML model

generated by artists and sample ~ 30000 frames per identity. We increase the variation of the artist generated animation sequences by also including segments that were rare in our real dataset. To improve realism and increase diversity, we augment the data with facial hair and glasses. The hair generation process involved procedurally growing hair and fitting it to the scalp of the rigged model. As for glasses, we employed 3D assets created from a set of frontal scans to generate a wide variety of glasses designs, which were then fitted to the rigged model. Although a visual domain gap persists between real and synthetic data, this synthetic dataset offers perfect labels that can be effectively utilized to improve the accuracy of our ML model.

7. On-Device ML Model

7.1. Architecture

Using the dataset described in Sec. 6, we train a CNN that is specifically designed for efficient on-device inference. As illustrated in Fig. 7, the model consists of a headset-mounted camera (HMC) image feature encoder and a multi-branch blendshape regressor.

HMC image feature encoder. We employ a ResNet-like [19] CNN backbone to extract features from each of the HMC images. We flip the images acquired from the left eye and the left mouth cameras allowing us to share the parameters between the left and right, eye and mouth backbone models, respectively. The input images are resized to 224×224 . The 512-channel output feature maps are of resolution 7×7 , which are then averaged-pool into a feature vector of 512 dimensions for each image.

Multi-branch blendshape regressor. We estimate the final

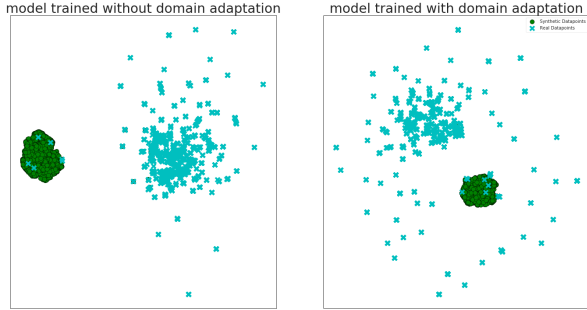


Figure 8. Visualize the ML model’s feature space (with UMAP [38]) for real data (blue crosses) and synthetic data (green dots). Left: model trained **without** domain adaptation. Right: model trained **with** domain adaptation. Domain adaptation better aligns the features between the real and synthetic data domains.

blendshape coefficients by assembling the per-image features from the different branches. The left eye features are fed into a multi-layer perceptron (MLP) to estimate the left eye blendshapes. Similarly for the right eye, which shares the same MLP parameters as the left eye. Finally, features from all the cameras are concatenated to estimate the blendshapes for the rest of the face. In our dataset the left and right eyes are often open or closed simultaneously leading the ML model to learn such predominant correlation if simply concatenating all the features and estimating all blendshape coefficients at once. Our network design enables to better capture rare asymmetric upper face expressions (such as winking) by explicitly breaking correlations in the network architecture.

7.2. Training Framework

We train our ML model end-to-end with an ℓ_1 loss between the estimated blendshapes and the labels using the Adam optimizer [27]. However to efficiently use the synthetic dataset we need to modify our training framework to accommodate for the domain gaps between the different modalities.

Domain Adaptation. As noted in Sec. 6.3, although the synthetic data looks fairly close to the real data, a significant domain gap still exists. The distinction between real and synthetic data becomes evident when observing the disparity in feature distribution, as demonstrated in Fig. 8.

This existing domain gap poses a challenge for the ML model to effectively leverage the knowledge present in both the synthetic and real datasets. To address this challenge, we follow the gradient reversal technique proposed in [14] which performs feature-level domain adaptation during the ML model training. Specifically, we first build a mini-batch with equal numbers of real and synthetic data frames. Then, we train an additional domain discriminator with binary cross entropy loss based on the image features. Finally, the gradient from the discriminator is reversed before back-

propagating to the feature encoders. As demonstrated in Fig. 8, this procedure better aligns the features between the two domains, letting the ML model further benefit from the samples additionally provided by the synthetic data.

Iterative Distillation. During our experiments, we discovered that naively training the ML model on real data labels resulted in expressions that appeared “muted”. For instance, when a person fully raised their eyebrows, the avatar’s eyebrows would only exhibit a slight movement. This discrepancy arose because real data labels have been generated automatically on a per-subject basis using a self-supervised learning technique (Sec. 6.1.3). As a consequence, these labels contain inherent noise and outliers, which forces the network to produce some amount of averaging to fit the noisy distribution of blendshape coefficients.

Inspired by student-teacher approaches that have been used to learn from noisy labels [21, 48], we propose an iterative training algorithm to address this challenge (see Algorithm 1). As part of the iterative training process, we also incorporate the Art Priors (Sec. 6.2) to refine and improve the network’s ability to generate more accurate and meaningful results.

Our process starts with the initial real data R_0 and synthetic data S . Then, we repeat the distillation process for T iterations as follows:

ALGORITHM 1: Iterative distillation algorithm

Data: Initial pseudo-labeled real data R_0 , synthetic data S

Result: Final on-device ML model m

for $t \leftarrow 1$ **to** T **do**

Model Pool $M \leftarrow \text{Train}(R_{t-1}, S)$
 $M \leftarrow M \cup \text{Train}(\text{PostProcess}(R_{t-1}, S))$
 Best Performed Models $M^* \leftarrow \text{Select}(M)$
 $R_t \leftarrow \text{EnsembleInference}(M^*, R_{t-1})$

end

$m \leftarrow \text{Train}(\text{PostProcess}(R_T), S)$

In each iteration, we first train a set of models with the pseudo-labeled real data and the synthetic data, with different random seeds. Second, we post-process the pseudo labels with Range-of-Motion calibration and temporal smoothing, and train another set of models. Third, we select several best performed models based on quantitative metrics (Sec. 8.1). Last, we infer the best performed models over the real data and ensemble the inference results to be the new pseudo labels.

Empirically, the quantitative metrics often stop to improve after 5 or 6 rounds. We thus train the final on-device ML model with the last post-processed pseudo-labeled real data and the synthetic data.

This algorithm can be viewed as using the ML models to iteratively refine the initial pseudo labels toward the “true” ground truth. Since the true ground truth is unknown, the quantitative metrics serve as a proxy to measure how close we are.

8. Evaluation

To thoroughly and comprehensively evaluate the effect of our face tracking solution to the **end-user experience**, we propose and develop three tiers of evaluation approaches.

Quantitative Metrics. Automatically compute several heuristic blendshape-based metrics that we found correlated with the user experience.

Qualitative Evaluation (QE). Render the tracking results as avatars alongside the corresponding camera images. Send to human annotators to rate whether the tracking results match the expressions performed.

User Experience Research (UXR). Build VR Apps for 1-on-1 conversation and small group meetings. Integrate our face tracking solution to the Apps and allow it to be turned on/off. Recruit a group of diverse users to try out the VR Apps and thoroughly evaluate the experience through questionnaires.

Based on the quantitative metrics, we further analyze how the iterative distillation (Sec. 8.2) and the training dataset size impact on the accuracy of the ML model. Lastly, we discuss the limitations of our system.

8.1. Quantitative Metrics

Conventional metrics, such as comparing the ℓ_1/ℓ_2 distance between the estimated and ground truth (GT) blendshape coefficients or mesh vertices, are less effective in our case. Because 1) we only have the pseudo GT, not necessarily the true GT, and 2) the conventional metrics do not correlate well with the end-user experience, *e.g.*, when the user is gently smiling, the tracking results of larger smiling and gently frowning could lead to similar mesh errors, but they mean very differently to the user.

To address the challenges, we propose a set of heuristic metrics that are comprehensive and correlate with the end-user experience:

- **Semantic Accuracy.** Measures if certain key blendshapes are activated enough as expected for each peak expression. The expected blendshape activation for each peak expression is predefined by artists (Sec. 6.2). This is the **most important** metric reflecting the expressivity of the tracker.
- **Neutralness.** Measures if the blendshape coefficients are below certain thresholds when the user stays neutral.
- **Smoothness.** Measures if the blendshape activation curves are smooth, by measuring the mean of the second order derivatives.

Table 1. We evaluate the quantitative metrics for the pseudo GT, the initial ML model trained only with the pseudo GT, and the final ML model trained with the various strategies elaborated in Sec. 7.2. While at a moderate trade-off on *Neutralness* and *Smoothness*, the final model significantly improves on *Semantic Accuracy* and *Mouth Closure*, which improves the overall end-user experience in practice.

	<i>Semantic Accuracy</i>	<i>Neutralness</i>	<i>Smoothness</i>	<i>Eye Closure</i>	<i>Mouth Closure</i>
Pseudo GT	0.392	0.912	0.940	0.916	0.703
Init. Model	0.403	0.902	0.832	0.944	0.856
Final Model	0.700	0.774	0.868	0.936	0.905

- **Eye Closure.** Measures if the avatar’s eyes are fully closed when the user fully closes their eyes, including winking.
- **Mouth Closure.** Measures if the avatar’s mouth is fully closed when the user fully closes their mouth.

The metrics are first computed for each expression recording, then averaged across all the expressions for each subject, and finally averaged across all the subjects as the dataset-level metrics. This three-level (recording / subject / dataset) aggregation not only gives us an overall evaluation of the system, but also allows us to effectively locate certain problematic recordings or subjects to improve the system.

Note that the metrics do not rely on the pseudo ground truth. In fact, we can use the metrics to evaluate both the ML model and the pseudo ground truth, as shown in Tab. 1. This is important to our proposed Iterative Distillation training strategy (see Sec. 7.2).

8.2. Impact of Iterative Distillation

As briefly illustrated in Tab. 1, the iterative distillation is important for model accuracy, *e.g.*, improving the *Semantic Accuracy* from 0.4 to 0.7. Here we study its impact more thoroughly from several aspects.

Iterations are necessary. We first validate that multiple iterations of distillation are necessary to improve the model accuracy before having marginal gains. As demonstrated in Fig. 9, the *Semantic Accuracy* has large improvements in the first two rounds and keeps increasing till Round 5.

Distillation as label denoising. The distillation process can be viewed as denoising the pseudo labels. We validate this by visualizing the ML model’s latent feature space along the distillation iterations. As demonstrated in Fig. 10, there are some outliers in the Round 1 model’s feature space which correspond to wrong blendshape coefficient estimation. As the distillation iteration goes, they gradually “move” into the distribution of inliers, and eventually have correct blendshape coefficient estimation.

Model ensemble is important. Lastly, we validate that model ensemble in the iterative distillation (Algorithm 1) is important. We conduct an ablation study that only one

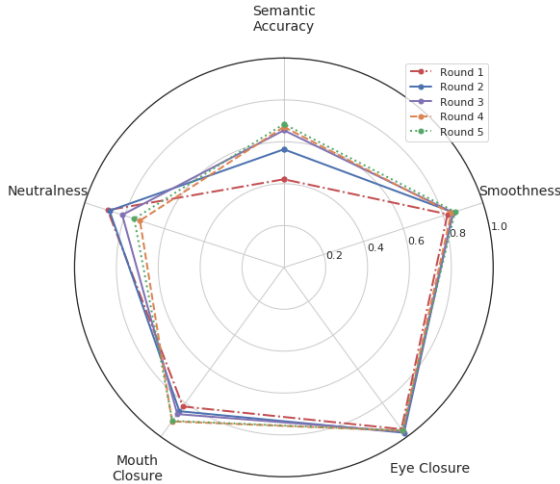


Figure 9. Quantitative metrics of the model trained at the end of each distillation iteration. The key metric, *Semantic Accuracy*, keeps increasing till Round 5.

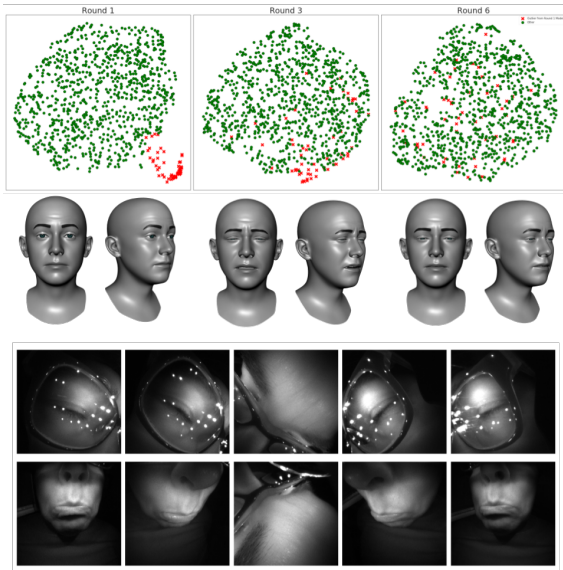


Figure 10. Iterative distillation can be viewed as denoising pseudo labels. The first row shows the model’s latent feature space (with UMAP [38]) at different iterations. Red dots indicate outlier data points, among which we find an outlier sample and visualize its blendshape estimation in the second row. The bottom two rows show as reference the camera images of that outlier sample frame.

best performed model is selected in each distillation iteration, *i.e.*, we train an initial model, using it as a teacher to train another model, and repeat. Tab. 2 shows the comparison between using model ensemble or not, where we can clearly see that using model ensemble performs much better on most of the metrics.

Table 2. Comparison between model ensemble or not in iterative distillation

	<i>Semantic Accuracy</i>	<i>Neutralness</i>	<i>Smoothness</i>	<i>Eye Closure</i>	<i>Mouth Closure</i>
w/o Ensemble	0.661	0.601	0.822	0.907	0.926
w/ Ensemble	0.700	0.774	0.868	0.936	0.905

8.3. Limitations

At times, the resolution and placement of our IR cameras may restrict the level of detail that our system can capture. This is especially noticeable for users with obstructed facial features, like lips covered by facial hair or eyebrows hidden by thick glasses. Besides, although we aimed to make our system easily adoptable by implementing a blendshape model, this representation has certain inherent limitations. Since the blendshape bases are manually designed, they might not be the most optimal for achieving a compact representation and capturing subtle movements. Additionally, since the bases can have linear dependencies, multiple sets of blendshape weights can produce similar expressions leading to semantic ambiguities.

9. Conclusion

We have demonstrated the feasibility of achieving high-quality facial animation in real-time using a VR headset without the need of manual assistance, such as user calibration. Robust tracking is achieved by ① embedding a set of IR cameras at strategic locations within the HMD, ② collecting a rich and high-quality dataset of images and labels, and ③ developing a novel training framework to improve the accuracy of our ML model. Enhancing our ML model with audio and temporal information are future research areas that could help increase the realism even under extreme occlusions due to facial hairs or other obstructions. We believe that our work will inspire further contributions in the development of consumer-level VR face trackers and will pave the way for new interaction metaphors and social presence experiences.

References

- [1] ShahRukh Athar, Zexiang Xu, Kalyan Sunkavalli, Eli Shechtman, and Zhixin Shu. Rignerf: Fully controllable neural 3d portraits. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 2
- [2] Linchao Bao, Xiangkai Lin, Yajing Chen, Haoxian Zhang, Sheng Wang, Xuefei Zhe, Di Kang, Haozhi Huang, Xinwei Jiang, Jue Wang, Dong Yu, and Zhengyou Zhang. High-fidelity 3d digital human head creation from rgb-d selfies. *ACM Transactions on Graphics*, 2021. 2
- [3] 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*, 1999. 1, 2, 3

- [4] James Booth, Anastasios Roussos, Stefanos Zafeiriou, Allan Ponniah, and David Dunaway. A 3d morphable model learnt from 10,000 faces. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016. 1
- [5] Sofien Bouaziz, Yangang Wang, and Mark Pauly. Online modeling for realtime facial animation. *ACM Transactions on Graphics (ToG)*, 2013. 2
- [6] Sofien Bouaziz, Andrea Tagliasacchi, Hao Li, and Mark Pauly. Modern techniques and applications for real-time non-rigid registration. In *SIGGRAPH ASIA 2016 Courses*. 2016. 2, 4
- [7] Zehranaz Canfes, M Furkan Atasoy, Alara Dirik, and Pinar Yanardag. Text and image guided 3d avatar generation and manipulation, 2022. 2
- [8] Timothy F. Cootes, Gareth J. Edwards, and Christopher J. Taylor. Active appearance models. *IEEE Transactions on pattern analysis and machine intelligence*, 2001. 1
- [9] Daniel Cudeiro, Timo Bolkart, Cassidy Laidlaw, Anurag Ranjan, and Michael Black. Capture, learning, and synthesis of 3D speaking styles. *Computer Vision and Pattern Recognition (CVPR)*, 2019. 2
- [10] Pengfei Dou and Ioannis A Kakadiaris. Multi-view 3d face reconstruction with deep recurrent neural networks. *Image and Vision Computing*, 2018. 2
- [11] Gareth J Edwards, Christopher J Taylor, and Timothy F Cootes. Interpreting face images using active appearance models. In *Proceedings Third IEEE International Conference on Automatic Face and Gesture Recognition*, 1998. 1
- [12] Paul Ekman and Wallace V Friesen. Facial action coding system. *Environmental Psychology & Nonverbal Behavior*, 1978. 3, 4
- [13] Yao Feng, Haiwen Feng, Michael J. Black, and Timo Bolkart. Learning an animatable detailed 3D face model from in-the-wild images. 2021. 2
- [14] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030, 2016. 6
- [15] Baris Gecer, Stylianos Ploumpis, Irene Kotsia, and Stefanos Zafeiriou. Ganfit: Generative adversarial network fitting for high fidelity 3d face reconstruction. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2
- [16] Baris Gecer, Alexander Lattas, Stylianos Ploumpis, Jiankang Deng, Athanasios Papaioannou, Stylianos Moschoglou, and Stefanos Zafeiriou. Synthesizing coupled 3d face modalities by trunk-branch generative adversarial networks. In *Proceedings of the European conference on computer vision (ECCV)*, 2020. 2
- [17] Baris Gecer, Stylianos Ploumpis, Irene Kotsia, and Stefanos P Zafeiriou. Fast-ganfit: Generative adversarial network for high fidelity 3d face reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 2
- [18] Jianzhu Guo, Xiangyu Zhu, Yang Yang, Fan Yang, Zhen Lei, and Stan Z Li. Towards fast, accurate and stable 3d dense face alignment. In *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIX*, 2020. 2
- [19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016. 5
- [20] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, 2017. 4
- [21] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015. 6
- [22] Alexandru Eugen Ichim, Sofien Bouaziz, and Mark Pauly. Dynamic 3d avatar creation from hand-held video input. *ACM Transactions on Graphics (ToG)*, 2015. 2
- [23] Amin Jourabloo and Xiaoming Liu. Large-pose face alignment via cnn-based dense 3d model fitting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016. 2
- [24] Tero Karras, Timo Aila, Samuli Laine, Antti Herva, and Jaakko Lehtinen. Audio-driven facial animation by joint end-to-end learning of pose and emotion. *ACM Transactions on Graphics (TOG)*, 2017. 2
- [25] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, 2006. 4
- [26] Kihong Kim, Yunho Kim, Seokju Cho, Junyoung Seo, Jisu Nam, Kychul Lee, Seungryong Kim, and KwangHee Lee. Diffface: Diffusion-based face swapping with facial guidance. *arXiv preprint arXiv:2212.13344*, 2022. 2
- [27] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. 2014. 4, 6
- [28] Reinhard Knothe, Brian Amberg, Sami Romdhani, Volker Blanz, and Thomas Vetter. Morphable models of faces. *Handbook of Face Recognition*, 2011. 1
- [29] John P Lewis, Ken Anjyo, Taehyun Rhee, Mengjie Zhang, Frederic H Pighin, and Zhigang Deng. Practice and theory of blendshape facial models. *Eurographics (State of the Art Reports)*, 2014. 2
- [30] Hao Li, Thibaut Weise, and Mark Pauly. Example-based facial rigging. *ACM Transactions on Graphics (Proceedings SIGGRAPH 2010)*, 2010. 3, 4
- [31] Stephen Lombardi, Jason M. Saragih, Tomas Simon, and Yaser Sheikh. Deep appearance models for face rendering. *CoRR*, 2018. 2
- [32] Iain Matthews and Simon Baker. Active appearance models revisited. *International journal of computer vision*, 2004. 1
- [33] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part VIII 14*, 2016. 4
- [34] Ivan Perov, Daiheng Gao, Nikolay Chervoniy, Kunlin Liu, Sugasa Marangonda, Chris Umé, Mr Dpfks, Carl Shift Facenheim, Luis RP, Jian Jiang, et al. Deepfacelab: Integrated, flexible and extensible face-swapping framework. *arXiv preprint arXiv:2005.05535*, 2020. 2

- [35] KR Prajwal, Rudrabha Mukhopadhyay, Vinay P Namboodiri, and CV Jawahar. A lip sync expert is all you need for speech to lip generation in the wild. In *Proceedings of the 28th ACM International Conference on Multimedia*, 2020. 2
- [36] Alexander Richard, Michael Zollhöfer, Yandong Wen, Fernando De la Torre, and Yaser Sheikh. Meshtalk: 3d face animation from speech using cross-modality disentanglement. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021. 2
- [37] Elad Richardson, Matan Sela, and Ron Kimmel. 3d face reconstruction by learning from synthetic data. In *2016 fourth international conference on 3D vision (3DV)*, 2016. 2
- [38] Tim Sainburg, Leland McInnes, and Timothy Q. Gentner. Parametric umap: learning embeddings with deep neural networks for representation and semi-supervised learning. *ArXiv e-prints*, 2020. 6, 8
- [39] Shunsuke Saito, Tianye Li, and Hao Li. Real-time facial segmentation and performance capture from rgb input. In *Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII 14*, 2016. 2
- [40] 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
- [41] Anh Tuan Tran, Tal Hassner, Iacopo Masi, and Gérard Medioni. Regressing robust and discriminative 3d morphable models with a very deep neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017. 2
- [42] Georgios Tzimiropoulos, Joan Alabort-i Medina, Stefanos Zafeiriou, and Maja Pantic. Generic active appearance models revisited. In *Computer Vision—ACCV 2012: 11th Asian Conference on Computer Vision, Daejeon, Korea, November 5-9, 2012, Revised Selected Papers, Part III 11*, 2013. 1
- [43] Daniel Vlasic, Matthew Brand, Hanspeter Pfister, and Jovan Popović. Multilinear models for face synthesis. In *ACM SIGGRAPH 2004 Sketches*. 2004. 2, 4
- [44] Shih-En Wei, Jason Saragih, Tomas Simon, Adam W. Harley, Stephen Lombardi, Michal Perdoch, Alexander Hypes, Dawei Wang, Hernan Badino, and Yaser Sheikh. Vr facial animation via multiview image translation. *ACM Trans. Graph.*, 2019. 4
- [45] Shih-En Wei, Jason Saragih, Tomas Simon, Adam W Harley, Stephen Lombardi, Michal Perdoch, Alexander Hypes, Dawei Wang, Hernan Badino, and Yaser Sheikh. Vr facial animation via multiview image translation. *ACM Transactions on Graphics (TOG)*, 2019. 2
- [46] Thibaut Weise, Sofien Bouaziz, Hao Li, and Mark Pauly. Realtime performance-based facial animation. *ACM transactions on graphics (TOG)*, 2011. 2
- [47] Fanzi Wu, Linchao Bao, Yajing Chen, Yonggen Ling, Yibing Song, Songnan Li, King Ngi Ngan, and Wei Liu. Mvf-net: Multi-view 3d face morphable model regression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019. 2
- [48] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *CVPR*, 2020. 6
- [49] Xiangyu Zhu, Zhen Lei, Xiaoming Liu, Hailin Shi, and Stan Z Li. Face alignment across large poses: A 3d solution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016. 2
- [50] Michael Zollhöfer, Michael Martinek, Günther Greiner, Marc Stamminger, and Jochen Süßmuth. Automatic reconstruction of personalized avatars from 3d face scans. *Computer Animation and Virtual Worlds*, 2011. 2