REFA: Real-time Egocentric Facial Animations for Virtual Reality Supplementary

1. End-to-end System Implementation

1.1. Head Pose

Our model does not output head pose, instead we utilize the 6-DoF head pose reported by HMD's SLAM service.

1.2. Eye Tracking

Our face tracking system also relies on the gaze output by an eye tracking module. This eye tracking module's core is a deep learning model which is trained end-to-end with a dataset of HMD's eye camera images and ground truth gaze.

The gaze controls both the movement of eye balls as well as 8 eyelid blendshape coefficients (4 per eye). The coefficients of those eyelid blendshapes are linearly mapped from the gaze angle (capped at $\pm$ 30°) for each eye separately.

1.3. Real-Time User Calibration

Although the model is trained with a large-scale dataset, it still cannot perfectly generalize to every new subject due to the inherent ambiguity between the user's identity and expression. For example, subjects with naturally narrower eyes can be confused with a partial eye closure expression on subjects with larger eyes (Fig. 1).

We proposed a real-time user calibration algorithm to address this challenge. The algorithm tracks the most recent 30 seconds' blendshape coefficients estimated by the ML model and assumes the most frequent "true" value of each blendshape coefficient should be 0. For each blendshape i, we generate a histogram of its coefficients over the last 30 seconds and extract the mode of the histogram m_i. Let b_i be the blendshape coefficient for the current frame, the calibrated coefficient is computed as

$$\tilde{b_i} = \frac{b_i - m_i}{1 - m_i}. \tag{1}$$

Note the assumption (the most frequent "true" value of a certain blendshape should be 0 during 30 seconds) only makes sense to a subset of blendshapes (*e.g.*, *eyebrow raise*) but not others (*e.g.*, *jaw drop*, as people may keep talking). We only apply the calibration to those blendshapes where the assumption holds.

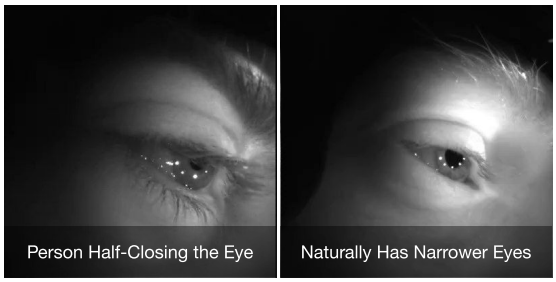

Figure 1. Inherent ambiguity between identity and expression. Left: a person half-closing their eyes, where the *eye closure* blendshape should be activated to about 0.5. Right: a person naturally having narrower eyes when neutral, where the *eye closure* blendshape should not be activated.

We further customize the calibration for *eye closure* blendshapes to account for cases where the person has naturally narrower eyes. We know that humans blink for about 5% of time. Let j be the *eye closure* blendshape, m_j be the mode of the histogram for the blendshape j, M_j be the top 2% of the histogram, and b_j be the blendshape coefficient for the current frame, the calibrated coefficient for the *eye closure* blendshape is computed as

$$\tilde{b_j} = \frac{b_j - m_j}{M_j - m_j}. \tag{2}$$

1.4. Fallback Mechanism

After deploying our face tracking models onto the headset, we encountered various challenging situations where the camera-based tracking failed. The most common failures were due to lacking visibility of users' facial features either because of poor donning or occlusion (*e.g.*, people touching their lower face temporarily, or people having a thick moustache or wearing a facial mask), as demonstrated in Fig. 2.

To address the challenges and ensure a less disruptive user experience, we propose a fallback mechanism, as illustrated in Fig. 3. First, we develop an uncertainty estimation module to check how confident the camera-based tracking result is. When the confidence is low, we either fallback

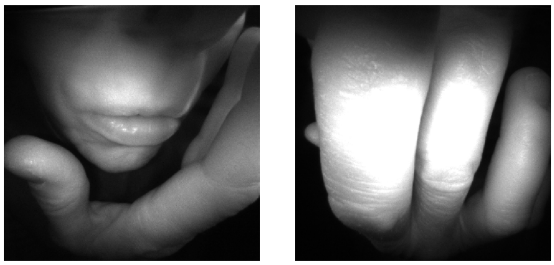

Figure 2. Two sample mouth camera images where our camera-based face tracking could be inaccurate or fail due to hand occlusions

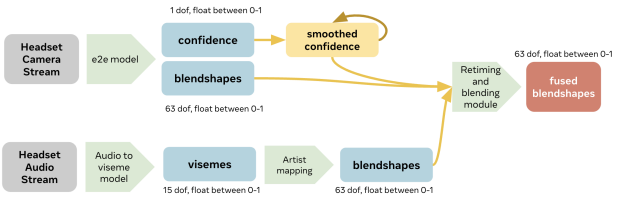

Figure 3. Diagram of the fallback pipeline. Since the camera images and audio signals have different frame rates, we use retiming to better align their estimated blendshape coefficients.

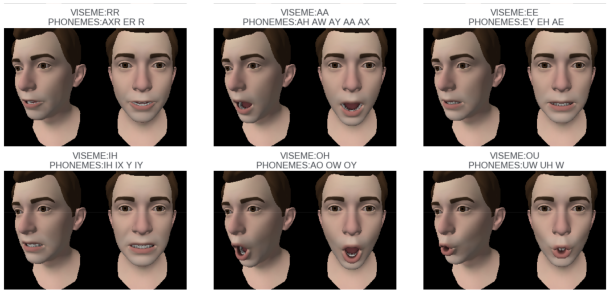

Figure 4. Sample mappings between phonemes, visemes, and blendshapes

to neutral (by just muting the blendshape coefficients) if no audio signals or fallback to blendshapes driven by audio signals using publicly available systems such as Oculus LipSync [4].

We utilize an existing audio-to-blendshape system to estimate phonemes from the audio signals. Then, we map the phonemes to visemes, and then to artist-defined blendshapes, as demonstrated in Fig. 4. For uncertainty estimation, we train a confidence regressor head that takes in the image features and predicts a scalar confidence value in $[0, 1]$. We train this regressor with a binary cross entropy loss on a manually annotated corner-case dataset composed of common failure cases such as mouth occlusion, heavy facial hair, poor headset donning, etc.

As demonstrated in Fig. 5, our system can properly detect the failure and fallback to audio-driven blendshape animation when the camera-based face tracking fails due to camera occlusion.

Figure 5. A user making an "O" sound while covering up the camera with their hands. In such cases when the camera-based tracking fails but audio is available, our system can properly detect the failure and fallback to drive the lower face blendshape coefficients by audio signals.

2. Evaluation Metrics

2.1. Qualitative Evaluation (QE)

In addition to the automatically computed quantitative metrics, we also employ human annotators to qualitatively evaluate the face tracking results. We render the tracking results as avatars alongside the corresponding camera images, as demonstrated in Fig. 6, and then send the rendered video to human annotators to rate whether the tracking results match the expressions performed.

QE helps reveal more nuanced tracking errors or artifacts that cannot be easily captured by the quantitative metrics. We summarize the common errors and artifacts into several standard QE annotation questions, such as whether the teeth contact, blinking, winking, mouth lip movements, *etc*., are well tracked.

We qualitatively evaluate our final ML model on the test dataset of 16,695 recordings from 795 subjects. The QE results suggest that 7.8% of all the recordings have quality issues, among which the most frequent errors are teeth contact (32.9%), blinking (15.3%), mouth closure (15.1%), and winking (5.67%). This helps us identify the areas for future improvements.

2.2. User Experience Research (UXR)

We further directly evaluate the holistic end-user experience in real-world VR applications through UXR studies. To this end, we build VR Apps for 1-on-1 conversation and small group meetings. Our HMD has our face tracking solution enabled, while the baseline is leveraging an existing VR device (such as Meta's Quest) that only has audio-driven face

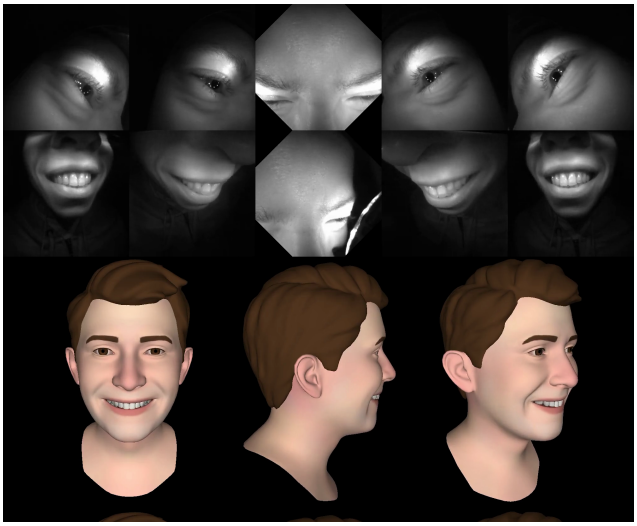

Figure 6. A sample frame in a rendered video for qualitative evaluation. The top two rows show the camera images, while the bottom row shows the face tracking result rendered as avatars from different views. Human annotators answer several standard questions such as whether the teeth contact and mouth lips are well tracked.

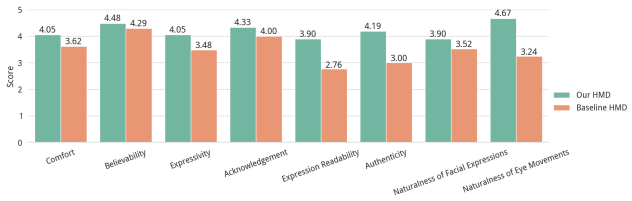

Figure 7. Comparison of the UXR study scores between our HMD (with our face tracking technology) and the baseline VR device (only has audio-driven face tracking). Our HMD outperforms the baseline HMD on all the axes.

tracking. We evaluate if our HMD enhances social presences and leads to more meaningful social interactions compared with the baseline VR device.

We design a *Triad Interaction* task for the UXR study. Three participants meet in VR to first play an ice-breaker game and then deliberate on a business ethics scenario. Participants repeat the meeting twice, one with our HMD and the other with the baseline VR device, in a random order. After each meeting, participants provide feedback on the social presence they felt during the interaction, by scoring 1 to 5 along the axes of *Comfort*, *Believability*, *Expressivity*, *Acknowledgement*, *Expression Readability*, *Authenticity*, *Naturalness of Facial Expressions*, and *Naturalness of Eye Movements*.

We recruit 21 diverse participants for the UXR studies. As shown in Fig. 7, our HMD outperforms the baseline VR device on all the axes. Overall, 90% of all the participants prefer our HMD over the baseline VR device for social interaction in VR.

Table 1. Impact of the number of training subjects on the accuracy of the ML model. "2950U" stands for downsampling to 2950 subjects for training.

Training Set	*Semantic Accuracy*	*Neutralness*	*Smoothness*	*Eye Closure*	*Mouth Closure*
0.1%, 9U	0.532	0.456	0.766	0.918	0.849
0.3%, 27U	0.581	0.594	0.810	0.857	0.829
0.5%, 41U	0.595	0.703	0.827	0.831	0.875
1%, 93U	0.631	0.613	0.825	0.882	0.887
10%, 750U	0.656	0.703	0.848	0.921	0.922
20%, 1482U	0.661	0.705	0.849	0.924	0.885
40%, 2950U	0.662	0.724	0.841	0.925	0.898
80%, 5662U	0.663	0.684	0.841	0.925	0.898
100%, 7161U	0.675	0.735	0.847	0.950	0.894

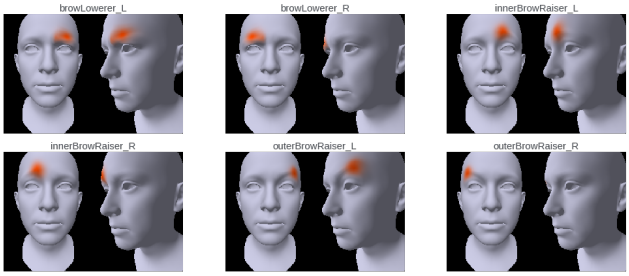

Figure 8. 6 blend shapes for eyebrow

3. Impact of Dataset Size

After thorough data cleaning, we construct a dataset consisting of 7,161 subjects for training and 795 subjects for testing. In this study, we aim at assessing the impact of the training dataset size on the accuracy of the ML model. We downsample the number of training subjects from 100% to 80%, 40%, ..., 0.1%, retrain the ML model respectively, and evaluate the results on the same test set. Note that the experiments have variations due to randomness in subject downsampling and model initialization. We report the results averaged across multiple runs, as shown in Tab. 1.

The accuracy improves rapidly as the number of training subjects increases, up until reaching 40% (2,950) training subjects. The improvements become more marginal afterwards. It can be interesting to further explore whether an "emergent" ability [5] may occur with even larger datasets or ML models.

4. Blendshapes

The figures below visualize our blend shape definitions.

5. Semantic Rig Constraints in Pseudo Ground Truth Generation

Our artist-provided template meshes are based on FACS shapes [2]. FACS shapes are not independent vectors – each

Figure 9. 14 blend shapes for eye regions, include 8 controlled by gaze

Figure 10. 8 blend shapes for cheek region

Figure 11. 12 blend shapes for mouth region

Figure 12. 11 blend shapes for upper lips

Figure 13. 9 blend shapes for lower lips

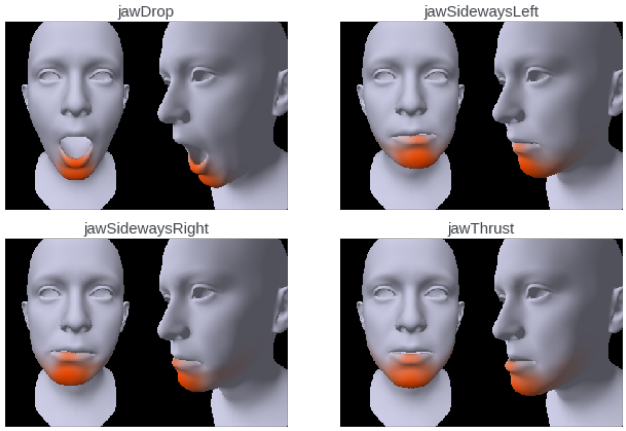

Figure 14. 4 blend shapes for jaw

shape implies specific activation patterns on other shapes, intended to represent the abilities of a human face well. Our implementation uses a set of non-linear constraints to describe these during optimization:

- Mutually exclusive shapes (e.g., mouth_left and mouth_right or eyes_closed and upper_lid_raiser),
- Equally activated shapes, implemented as averaging (e.g., 4 lips_towards shapes should have roughly the same activation),
- Conditional activations: (e.g., lips_towards shapes or the lip_suck shapes can only be active if jaw_open is activated at the same time).

Our top-level metrics don't change much when we compare a run with the rig constraints against a run without the rig constraints. Adding rig constraints restricts the degrees of freedom available for the optimizer, and it is expected to cause "visual" degradation in some cases. However, other frames show visual improvements and the rig constraints enforce an overall improved conformity to the artists rigging expectations. Figure 15 and Figure 16 show two exam-

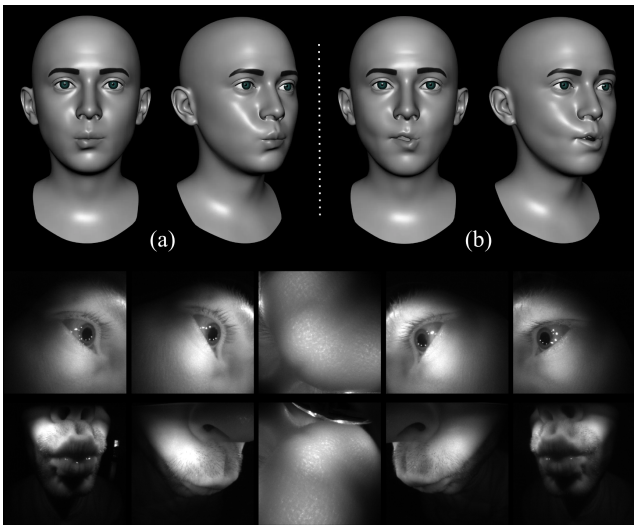

Figure 15. (a) With rig constraints, (b) without rig constraints on the kiss_face sequence. This is a good example of enforcing equal activation in the lipTowards blendshapes and a more visually appealing result with the rig constraints.

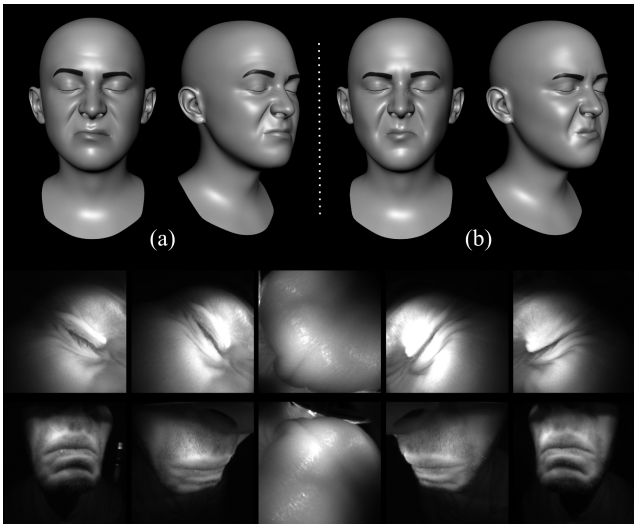

Figure 16. (a) With rig constraints, (b) without rig constraints on the scrunch_face sequence. With this particular rig, the difference in activation patterns are not very "visible" on the final rendering (there are some differences on the mouth shape and eye brows). However, when looking at the activation patterns, we see an incorrect co-activation of mouth-left (0.59) and mouth-right (0.63) without the rig constraints vs mouth-left (0) and mouth-right (0.20) with the rig constraints.

ples for which the blendshape activation patterns conform more to artists' rigging expectations.

6. Details of the On-Device ML Model

6.1. Backbone Architecture

Figure 17 shows the backbone architecture used in the on-device ML model. Note that the left and right eye share the same eye backbone, while the left and right mouth share the same mouth backbone.

6.2. Training Recipe

We use a single machine with 8 NVIDIA P100 or V100 GPUs to train our on-device ML model. The effective batch size is 224, consisting of half real data and half synthetic data. We subsample every 10 frames from the raw video and only keep the segment around peak expressions for each video (frame index percentile in $[35\%, 65\%]$). We set the max iteration to 60K, learning rate to 0.1, and weight decay to 0.0005. We employ a cosine learning rate scheduler [3] with 500 warm-up steps in the beginning.

6.3. Details in Iterative Distillation

Range-of-Motion (RoM) Calibration. We propose a RoM calibration algorithm to post-process the pseudo labels with the Art Priors. The goal is to make the blendshape activation expressive and more consistent across different training subjects.

During data collection, we ask each participant to stay neutral for 1-2 seconds; then perform a certain expression (will show some photo/video as a prompt to help the person mimic the same expression) and hold it for 3 seconds; at last fall back to neutral position. With the Art Priors defined by artists, we expect certain key blendshapes to be close to certain values at the peak expression segments. The overall blendshape activation over time should be similar to a bell curve.

With such priors, we design the algorithm as follows:

1. For each expression video, find the frame segments of neutral $\rightarrow$ ramp-up $\rightarrow$ peak $\rightarrow$ ramp-down $\rightarrow$ neutral, based on the blendshape estimation (either from pseudo ground truth or from the ML models)
2. For each key blendshape of that expression,
   (a) Keep its activation at the neutral segments unchanged
   (b) Scale its activation at the peak segments to be close to the art-expected value
   (c) Scale its activation in the ramp-up and ramp-down segments so that the transition is smooth

In Fig. 18, we demonstrate an example of how the blendshapes for expression "Surprise" are calibrated.

Temporal Smoothing. We apply the OneEuro Filter [1] to smooth each blendshape activation curve independently.

Model Selection. We select from the model pool the top-2 models that have the best *Semantic Accuracy* and *Mouth Closure* metrics for ensembling.

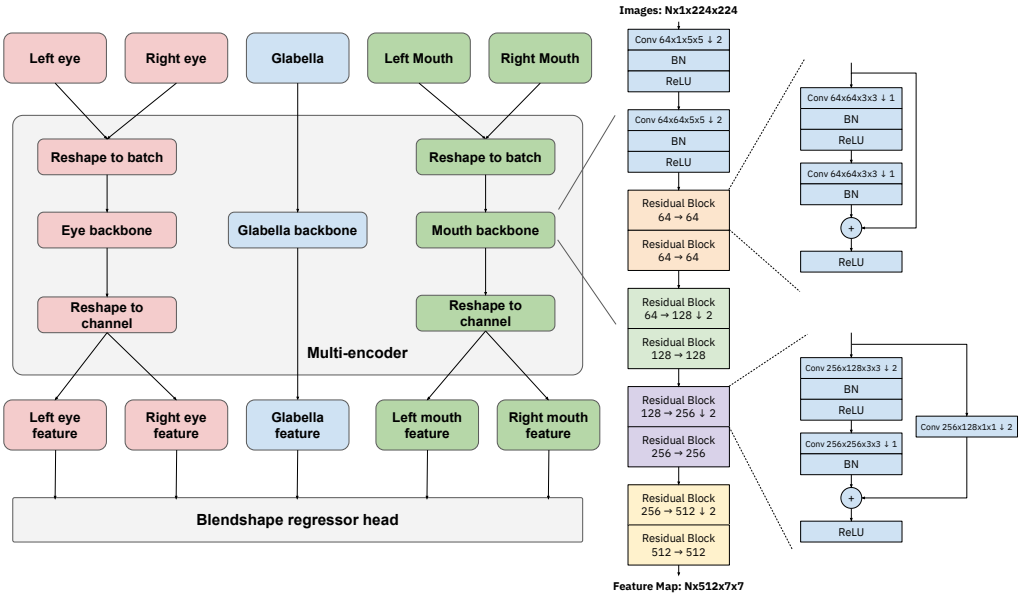

Figure 17. The backbone architecture for encoding each image into a feature map. "Conv 256x128x3x3 ↓ 2" means a convolution layer that has 256 output channels, 128 input channels, 3x3 kernel size, and a stride of 2.

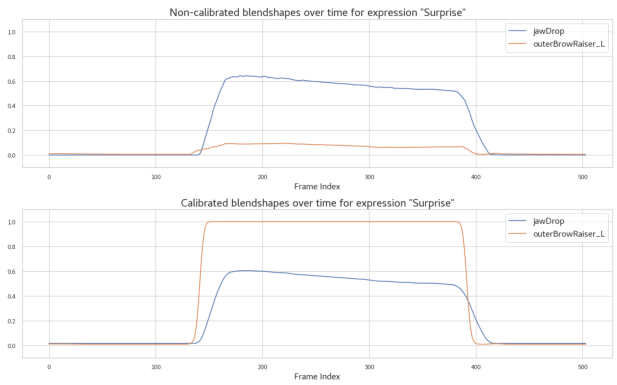

Figure 18. A sample demonstrating how the blendshapes are calibrated. For the "Surprise" expression, the jawDrop blendshape is expected to be at $\sim$ 0.6 and the outerBrowRaiser_L blendshape is expected to be at $\sim$ 1.0. The initial blendshape estimation in the top figure is calibrated toward such art priors. Note that the authentic movement is largely preserved (the jawDrop after calibration still preserves a minor degrading over time, rather than fixing at 0.6 as constant).

Ensemble. We use the average ensemble, *i.e.*, averaging the inference results across the models.

References

[1] Géry Casiez, Nicolas Roussel, and Daniel Vogel. 1 € filter: A simple speed-based low-pass filter for noisy input in interactive systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, page 2527–2530, New York, NY, USA, 2012. Association for Computing Machinery. 5

[2] Paul Ekman and Wallace V Friesen. Facial action coding system. *Environmental Psychology & Nonverbal Behavior*, 1978. 3

[3] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. 2017. 5

[4] Meta Platforms. Oculus lipsync, 2016. 2

[5] Barret Zoph, Colin Raffel, Dale Schuurmans, Dani Yogatama, Denny Zhou, Don Metzler, Ed H. Chi, Jason Wei, Jeff Dean, Liam B. Fedus, Maarten Paul Bosma, Oriol Vinyals, Percy Liang, Sebastian Borgeaud, Tatsunori B. Hashimoto, and Yi Tay. Emergent abilities of large language models. *TMLR*, 2022. 3