Supplementary Material for What we need is explicit controllability: Training 3D gaze estimator using only facial images

In this supplementary material, we provide more details, results and discussions about our method.

1. More Details

1.1. Gaze Targets on the Virtual Screen

In this part, we introduce how to place a virtual screen in front of our learned 3D Head. First, we calculate the screen-to-eye distance d based on the type of dataset. For instance, d is set as 1.0 meter in EVE to approximate the typical distance between face and screen. For the Columbia dataset, d is set as 2.5 meter by following its data collection process. Next, we calculate the horizontal field of view fov_x and the vertical field of view fov_y for the screen by using:

$$\begin{aligned} fov_{\text{x}} &= 2 \max_{i=1}^{N} |arcsin(\boldsymbol{g}_{\text{x}}^{i})| \\ fov_{\text{y}} &= 2 \max_{n=1}^{N} |arcsin(\boldsymbol{g}_{\text{y}}^{i})| \end{aligned} \tag{1}$$

where N is the total number of the input facial images, $\boldsymbol{g}^i$ is the average unit vector of the pseudo 3D gaze directions of the left eye $\boldsymbol{g}^i_{\text{left}}$ and right eye $\boldsymbol{g}^i_{\text{right}}$, $\boldsymbol{g}^i_{\text{x}}$ and $\boldsymbol{g}^i_{\text{y}}$ are the x and y components of $\boldsymbol{g}^i$, respectively.

1.2. Gaze Sample Generation

To generate facial images with corresponding gaze directions, we first sample the 3DMM and lighting parameters from the training images of each subject. Then, we place a virtual screen in front of the face (see Sec. 1.1) and randomly generate gaze targets on it, where each eyeball is rotated according to the line connecting the eyeball center to the gaze target (see Fig. 1). Third, we calculate a new camera pose by performing interpolation between two camera poses sampled from the training images. Finally, we render a new face image based on the new camera pose and normalize it to the size of 224×224 based on [5].

1.3. Network Architectures of Ω

Our lighting model Ω takes the lighting encoding $\boldsymbol{f}_{\text{light}}$ of each image and the positional encoding $\boldsymbol{f}_{\text{pos}}$ of the 3D Gaussian as input and outputs the color $\boldsymbol{c}$ of the corresponding 3D Gaussian. During the training phase, $\boldsymbol{f}_{\text{light}}$ and $\boldsymbol{f}_{\text{pos}}$ are initialized as zero vectors and are gradually optimized through backpropagation to learn the lighting information and high-frequency texture details from different images.

As shown in Fig.2, Ω is a 4-layer MLP, where the first three layers use the LeakyReLU activation function, and the last layer uses the sigmoid function to produce colors within the range of (0, 1). In practice, we set the dimensions of $\boldsymbol{f}_{\text{pos}}$ and $\boldsymbol{f}_{\text{light}}$ as 72 and 27, respectively.

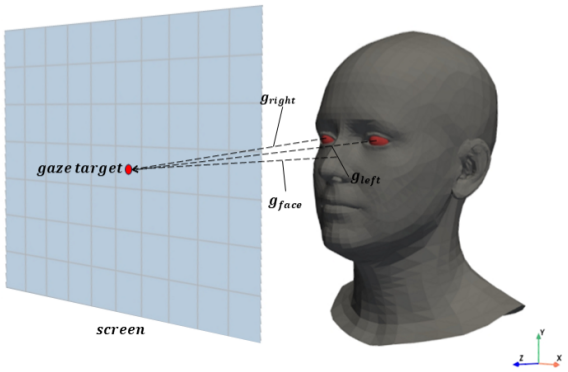

Figure 1. To generate pseudo gaze directions, we first construct a virtual screen in front of the face. Next, we randomly sample gaze targets on it. The eyeballs are then rotated according to the direction of the line connecting the centers of the left and right eyeballs to the selected gaze target, thereby determining the gaze direction. Finally, the rotated eyes, along with the original 3D representation, are used to generate the corresponding facial images.

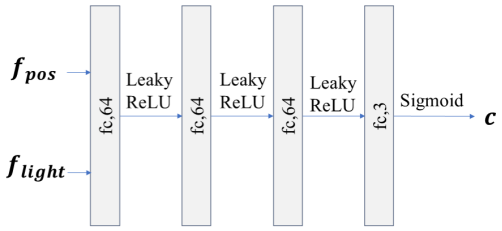

Figure 2. The network architecture of Ω.

2. More Results

2.1. Settings

We set λ_1=λ_4=λ_6=1.0, λ_2=0.75, λ_5=1.5, λ_3=1e-2, $\epsilon_{\text{scal}} = 0.6$ and $\epsilon_{\text{pos}} = 1.0$. These parameters are either inherited from existing works [1, 2] or adjusted to balance the scale of each term. To train the 3D head model of each subject, we use identical or fewer number of images compared with existing methods on each dataset, 600 for EVE, 300 for MPII and 105 for Columbia.

2.2. Head Pose

In this part, we discuss the impact of the head pose in the input facial image, where the facial images are captured from

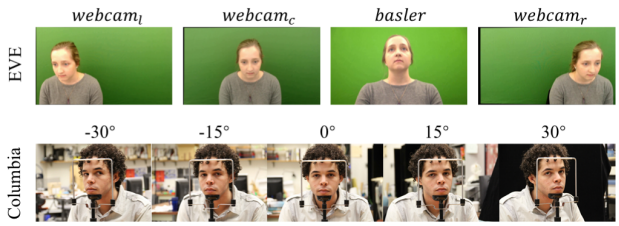

Figure 3. Example of different head poses on EVE and Columbia.

Cam View	webcam$_l$	webcam$_c$	basler	webcam$_r$
Gaze Error	12.45°	**7.39°**	8.80°	12.69°

Table 1. Evaluation results of different head poses on EVE.

Cam View	-30°	-15°	0°	15°	30°
Gaze Error	7.87°	6.45°	**6.08°**	6.64°	6.62°

Table 2. Evaluation results of different head poses on Columbia.

Dataset	EVE	Columbia	MPII
EVE	-	9.17°	9.45°
Columbia	13.75°	-	10.72°
MPII	12.01°	8.23°	-

Table 3. Cross-dataset results on EVE, Columbia and MPII by training on one dataset and evaluate on the other two.

Dataset	M+C→E	E+M→C	E+C→M
Gaze Error	10.35°	7.48°	7.78°

Table 4. Cross-dataset results on EVE (E), Columbia (C) and MPII (M) by training on two datasets and evaluate on the remaining one.

Method	1	2	3	4	5
UnityEyes[4]	23	45	9	58	25
SimGAN[3]	121	155	114	85	93
GaussianAvatars[2]	114	100	159	126	113
Ours	**242**	**200**	**218**	**231**	**269**

Table 5. The voting results of the five volunteers with respect to the eye images generated by using the four methods.

four or five different camera perspectives (see Fig. 3) and the pose variations among different subjects are relatively small for the same camera perspective. The gaze estimation results of different head poses are reported in Tab. 1 and Tab. 2. It is obvious that the gaze estimation performance is sensitive to head pose. The frontal face tends to have better results. Conversely, the lateral face tend to have worse results. The reason may lie in the decreased performance of facial landmark detection.

2.3. Cross-dataset Evaluation

In this part, we report our cross-dataset evaluation results on EVE, MPII and Columbia datasets under two different settings. The results shown in Tab. 3 and Tab. 4 reveal

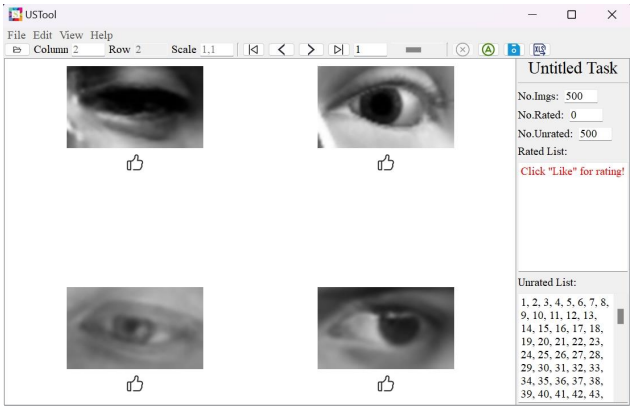

Figure 4. The employed tool for user study.

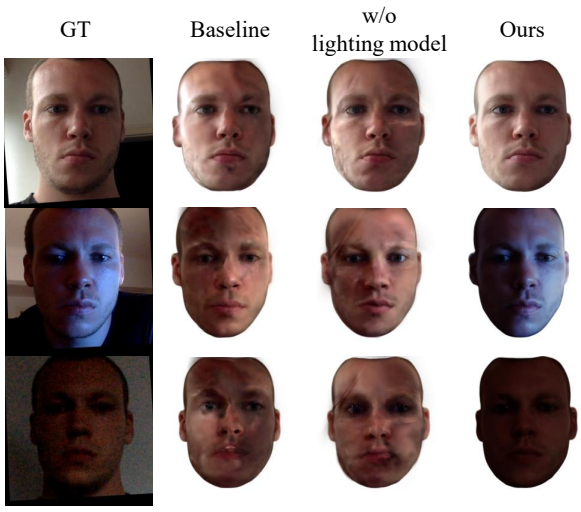

Figure 5. Visual demonstration of the results of our lighting model.

clear performance improvement on the gaze estimation when using more datasets for model training.

2.4. User Study

For user study, we recruited five volunteers by asking them to vote on 500 random samples generated by UnityEyes, SimGAN, GaussianAvatars and our method (see Fig.4). The results is reported in Tab. 5. Accordingly, the eye images generated by our method received the most votes from all five volunteers, which indicates that our method is capable of generating more realistic eye images that better conform to human perception.

2.5. Visual Ablation

Lighting model. As shown in Fig. 5, compared with the baseline [2] and Ours without lighting model, our final results exhibits better ability in capturing the lighting conditions of the input images.

RSR. Fig. 6 show that the introduction of RSR can help

Figure 6. Visual demonstration of the results of RSR. The red dots indicate the projected positions of the mesh pupil points.

to correctly align the pupil points in the mesh with the 3D Gaussian pupil points, and the generated iris images appear more circular.

3. Discussions

Although our method achieves state-of-the-art performance in both gaze prediction and image quality, a performance gap remains between our results and those of supervised methods. The possible reasons may lie in imperfect 3D representations and limited lighting variations for generating facial images. For instance, our lighting model can only capture the existing illumination in the input facial images and does not support relighting. Since our method requires to train a 3D head model for each subject, it may suffer from some extra time costs, but it can improve the flexibility on gaze data collection, and reduce the constrains and requirements for lab-based calibration on head and eyeballs for data collection. Besides, our method is limited to the performance of head tracker, which may restrict the geometrical realism. We intend to solve these limitations in our follow up work.

References

[1] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), 2023. 1

[2] Shenhan Qian, Tobias Kirschstein, Liam Schoneveld, Davide Davoli, Simon Giebenhain, and Matthias Nießner. Gaussianavatars: Photorealistic head avatars with rigged 3d gaussians. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20299–20309, 2024. 1, 2

[3] Ashish Shrivastava, Tomas Pfister, Oncel Tuzel, Joshua Susskind, Wenda Wang, and Russell Webb. Learning from simulated and unsupervised images through adversarial training. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2107–2116, 2017. 2

[4] Erroll Wood, Tadas Baltrušaitis, Louis-Philippe Morency, Peter Robinson, and Andreas Bulling. Learning an appearance-based gaze estimator from one million synthesised images. In *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research Applications*, pages 131–138, 2016. 2

[5] Xucong Zhang, Yusuke Sugano, and Andreas Bulling. Revisiting data normalization for appearance-based gaze estimation. In *Proceedings of the ACM Symposium on Eye Tracking Research Applications*, pages 1–9, 2018. 1