Artist-Friendly Relightable and Animatable Neural Heads

Supplementary Material

In this supplemental document we discuss additional qualitative results, experimental details and ablations in Section 6, and end with limitations and future work in Section 7.

6. Experiments

6.1. Additional Qualitative Results

Here we illustrate additional results of novel animation sequences and lighting interpolation & extrapolation.

Novel Animations. As our model takes only 3D meshes as input, it can generate novel controllable animations driven by different sources. Two different examples are shown in Fig. 9. In the first example (top 2 rows), an artist-created blendshape animation is used to drive our model. We show our result alongside the 3D mesh animation. In the second example (bottom 2 rows), we drive our model from monocular video input. Here we track the performance of the actor using a recent landmark-based 3D facial reconstruction method [5], and then feed the tracked meshes into our model. We show our result alongside the original video input. Given these two examples, we can see that our model can generate realistic renderings of various novel expressions outside of the training sequences.

Figure 9. Our method can render novel expressions driven by performances from different sources, *e.g.* artistically-created blendshape animations (top) or monocular face capture from a mobile phone (bottom).

Lighting Interpolation and Extrapolation. To evaluate how the model interpolates and extrapolates novel lighting directions, we render in Fig. 10 a point light orbiting around the head within a horizontal plane at a radius of 3 meters, starting from the left side of the face. As shown in the figure, our model can smoothly interpolate inside the range of the training lighting directions (0° to 180°), with coherently moving shadows and specular highlights. Although our 32 lights are spread out to cover only the frontal hemisphere, the model can extrapolate well to at least 20° towards the backside. While we see extrapolation artifacts beyond that, our model learns to predict reasonable shadow distributions even directly behind the captured subject.

6.2. Experiment Details

Here we provide further details regarding the quantitative evaluation in the main text, and a thorough explanation of how we render images with LatLong environment maps.

Quantitative Evaluation. For the quantitative evaluation experiment in Section 4.2, we leave out 3 lights for each subject. Fig. 11 shows the LatLong images of the captured light probe, where each LED bar is represented with a single position/direction. Lights used in training are illustrated as green dots and held-out lights are labeled in red. Note that we did not leave out any lights on the boundary as TRAvatar [51] cannot extrapolate outside the training lighting directions. We also compute the Delaunay triangulation of the training lights and use barycentric coordinates to evaluate held-out lights for TRAvatar*, as shown in Fig. 11. Similar to ReNeRF [50], we optimize per LED-bar intensity during training. When testing novel lights, we scale the predicted renders to match the scale of the ground truth images for all frames before computing the metrics in Table 1. We leave out the free dialog performance of Subject 1, and a scripted line of Subject 2 and Subject 3 as the validation set for novel performances. All numbers in Table 1 are computed in linear RGB space.

Rendering with LatLong Environment Maps. When rendering novel environment maps, our model follows the approach of image-based relighting. More specifically, we downsample the LatLong into uniformly distributed directions, render each of these directional lights, then compute a weighted sum in image space, with weights given by the light intensities. We drop the 10% of the lights with the lowest intensities to speed up rendering. To avoid extrapolation artifacts (Fig. 10), we mask out directions that fall into the

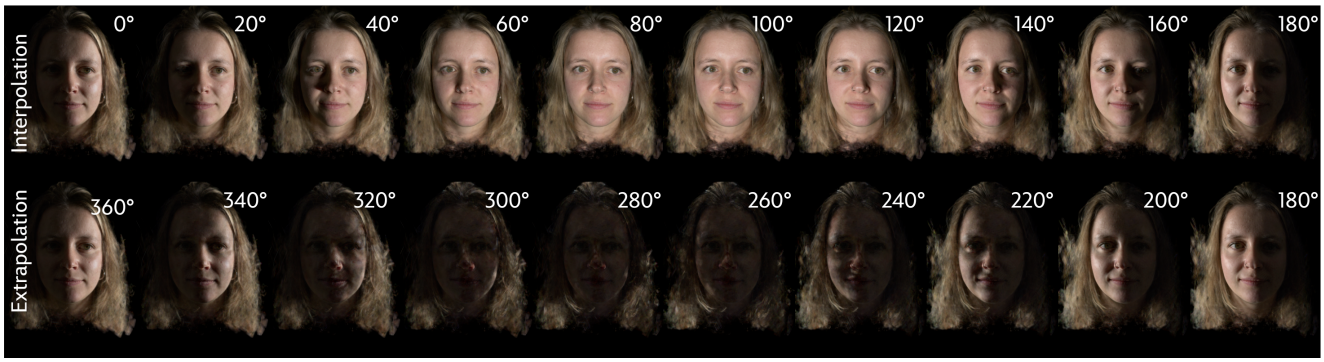

Figure 10. Relighting results of a point light orbiting 360° around the head. The top row (0° to 180°) shows our model can interpolate smoothly within the range of the training lighting directions. The bottom row (180° to 360°) shows our model can extrapolate to at least 20° on both sides and predict reasonable shadow distributions far outside the training data.

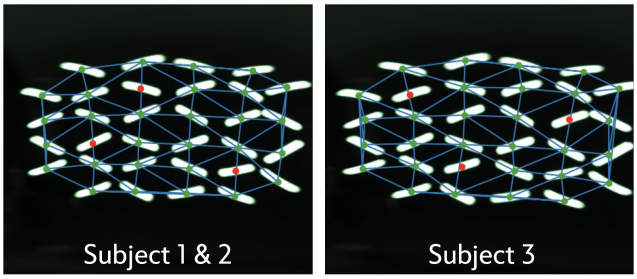

Figure 11. Captured light probe in Latlong format overlaid with computed triangulation of the training lights. Training lights are marked with green dots and held-out lights are marked in red.

45° cone whose axis points to the back. For the rest of the directions in the back hemisphere, we apply an attenuation term $a = \cos^4\theta$ (a is 1 at the side and 0 at the back) on their intensities to render smooth environment map animations. The effective resolution of all environment maps we use is approximately 256 directions. For more environment relighting results, please refer to the supplementary video.

6.3. Ablations

In this section, we compare our method with a modified version that incorporates the spherical codebook proposed in ReNeRF [50]. Instead of representing the lights as 3D vectors $\mathbf{l}_k$, we use 64D OLAT codes learned from a small 3-layer MLP. Each layer has 64 hidden units. The input light directions are position-encoded with a 5th-order spherical harmonics basis. While the differences in overall relighting quality across multiple frames are hardly noticeable, incorporating the learned OLAT codes does improve the shadow boundaries in some cases. However, it also impairs the specular reflections on the eyes. We show such an example of a novel point light relighting in Fig. 12, where the rendering of our method is on the left, and our method with OLAT codes is on the right.

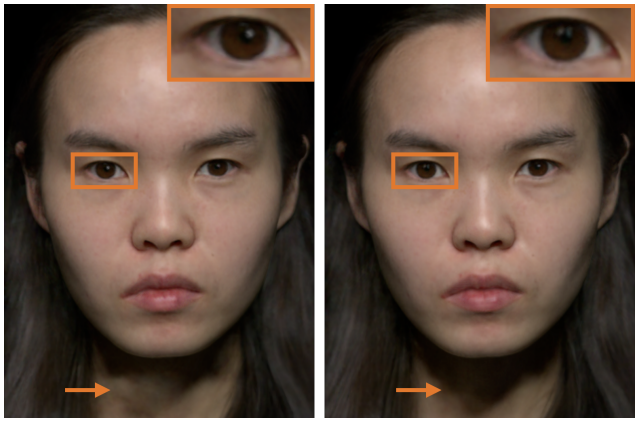

Figure 12. Rendering of a novel point light from our method (left) and our method with learned OLAT codes [50] (right). The learned OLAT codes improve interpolation of shadows but lead to less sharp rendering of specular reflections on the eyes.

7. Limitations & Future Work

One limitation of our method is that it does not achieve real-time performance. It takes about 15s per frame to render an environment map with 256 directions. As mentioned in the main text, we capture motion blur in some of our training frames due to our less expensive setup compared to other methods [2, 28, 51]. This can lead to blurry rendering for fast motions such as blinking. However, we show in Fig. 13 that our model does not overfit to all these blurry pixels and is able to recover some details lost in the captured images. Future work could consider modeling motion blur explicitly to render sharper images. Finally, in Fig. 14, we show failure examples of our method when extrapolating to extreme expressions far from the training data (left), and when the gaze is entangled with the base geometry (right). For future work, we plan to collect more data, which can help with expression extrapolation. Currently, our models

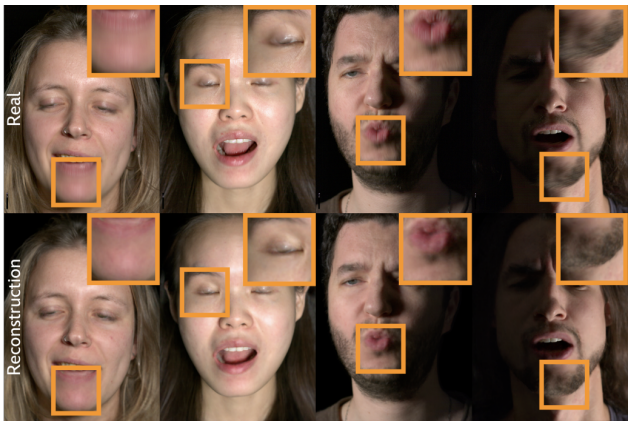

Figure 13. We capture motion blur in the training data due to our low frame rate, which can lead to blurry rendering of fast motions.

Figure 14. Failure cases. Our model breaks when extrapolating to extreme expressions far from the training data (left). And gazes cannot be disentangled from the base meshes (right).

are trained with only 10% of the amount of data as compared to the original MVP algorithm. We also plan to model gazes explicitly to allow for gaze animation.