Monocular Facial Appearance Capture in the Wild

Supplementary Material

In this supplementary document we begin by discussing the details of our capture protocol and initial tracking steps in Section 7. We provide more implementation details in Section 8. In Section 9 we highlight the particulars of our implementation of the related methods we use for comparisons, and elaborate on the synthetic dataset we used for evaluation. Finally, we add additional results and failure cases in Section 10.

7. Dataset Details

7.1. Capture Protocol

We record videos at 25 fps using a Canon EOS 1200D camera fixed on a tripod. Depending on the distance to the subject, the camera is mounted with either a 35mm or 60mm lens and we use the corresponding focal length as a known parameter when calibrating the camera. The subjects are asked to slowly rotate their head 20 to 30 degrees to the left, right, up, and down directions. We do not record large head rotations as we noticed that the estimated head poses tend to be inaccurate on side view. Fig. 10 (row 1) shows some example frames of a dataset, where the left-most and right-most frames are the most extreme head rotations in this sequence. Note that even though we do not see a side face silhouette in the data, our method can recover the correct nose shape from shading as we have shown in Fig. 8. The datasets we captured span across multiple days at different locations.

7.2. Initial Tracking

In the initial tracking stage, we estimate an initial mesh and per frame head poses. The initial mesh is parameterized using blendweights of a PCA face basis computed from the dataset of Chandran *et al*. [10], which includes 50 eigen faces for identity and 25 for expression. Our tracking algorithm uses the combination of a landmark [11] loss and a photometric loss, similar to Qian [53]. The only difference is that we solve for a global expression code since we assume the expression does not change in the same sequence. We apply a weight of 100 on the landmark loss and a weight of 30 on the photometric loss for all our datasets.

We further obtain an initial albedo estimate by averaging the projected texture across all the frames, and then we compute a piece-wise constant version based on a predefined face segmentation as in Rainer et al. [54]. The specular intensity map is initialized as a grayscale version of the initial diffuse albedo.

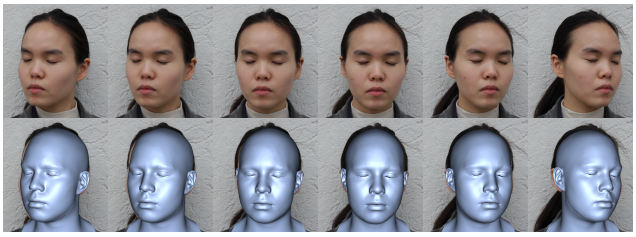

Figure 10. Example frames and initial tracking of a dataset. The first row shows in the input frames and the second row is the initial tracked geometry overlaid on the input.

8. Implementation Details

We capture 500 to 800 frames for each subject. We then uniformly sample around 250 frames to use in the inverse rendering. We did not observe a performance gain or drop when using all of the frames. The images are cropped to 1K resolution. We also solve the texture maps at 1K resolution. The environment map is a cubemap with resolution $6\times256\times256$ at the largest mip-level, and resolution $6\times8\times8$ at the smallest mip-level. For each pixel, we draw 256 light samples and 256 BRDF (cosine) samples for the diffuse render, and 64 samples to estimate the view-dependent specular visibility. We employ the Adam optimizer [19] with a learning of 0.1 for vertex positions and the environment map, and 0.001 for the textures. We use the differentiable rasterizer from Laine *et al*. [37] to obtain the primary visibility and the OptiX [52] engine for ray tracing. Each subject is trained for 6000 iterations which takes around 2 hours on a Nvidia RTX 3090 GPU. The weights for Eq. 10 are defined as

$$\begin{aligned} &\lambda_{\text{mask}} = 0.1, \lambda_{\text{Lap}} = 10, \lambda_{\text{light}} = 0.1, \\ &\lambda_{\text{rough}} = 0.1, \lambda_{\text{diffuse}} = 0.01. \end{aligned} \tag{11}$$

9. Experiment Details

9.1. Implementation of the Related Methods

Next we describe the steps performed to run the comparisons. We use the default parameters of FLARE [4]. We noticed that the FLARE geometry is very bumpy, hence we tried setting a larger weight on the Laplacian mesh regularizer. However, this resulted in flatter face geometry and did not improve the results.

For NextFace [15–17], we obtained the best results in our experiments using only three frames that cover the whole face region. A similar behavior was observed by Azinović *et al*. [1] in their experiments.

The original capture protocol used in SunStage [62] has a different format compared to ours; they record only a frontal

view with the person rotating 360 degrees in place. We thus adapt the preprocessing code to the one from FLARE when running SunStage. We also do not solve for the focal length of the camera and set it as the ground truth value. In our experiments, the shape does not change much from the initial DECA [22] result, in both the coarse alignment stage and the photometric optimization stage of SunStage.

NextFace and SunStage have no code for relighting in their release, so we did not compare relighting performance against these two baselines. The statistics in Table 1 are averaged over frames used in the optimization, *i.e.*, all the frames for our method, FLARE, SunStage, and only three frames for NextFace.

9.2. Synthetic Dataset

We render a synthetic dataset with Lambertian material for the ablation study in Fig. 6. The assets, *i.e.* ground truth diffuse albedo, mesh and environment maps are shown in row 1 of Fig. 11, and example frames are shown in row 2. The generated head poses are similar to those from a real dataset. We perform the same initial tracking algorithm using only the landmark loss on a front-facing frame. Instead of solving per frame head poses as for a real dataset, we use the ground truth head poses for the synthetic dataset in the inverse rendering stage.

10. Additional Results

Next we present additional results for the ablation and show some challenging situations for our algorithm.

Visualization of the Visibility. First we provide additional visualizations of the view-dependent visibility under different roughness in Fig. 12. This highlights the areas that are impacted by the visibility computation. When the roughness is small (mirror material), this visibility term is close to a binary mask and when the roughness is large (diffuse material), it gets closer to a view-independent ambient occlusion term. Note that the approximation error of Eq. 6

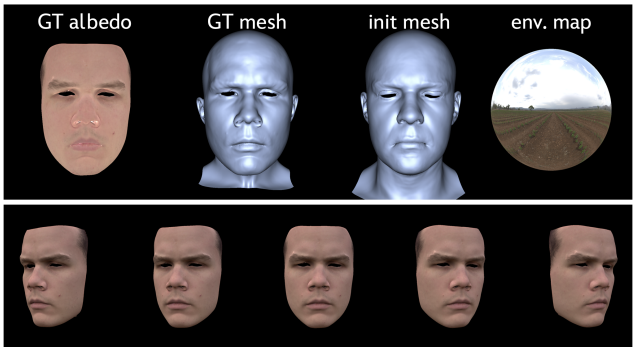

Figure 11. Assets and example frames of the synthetic dataset.

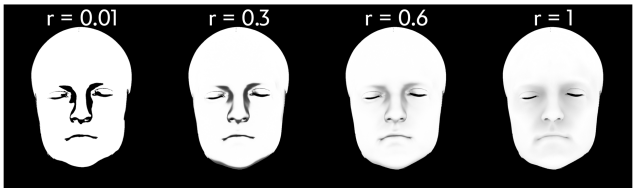

Figure 12. Visualization of the estimated view-dependent specular visibility term with different roughness values.

is small for a smaller roughness value and big for a larger roughness value. Applying the same approximation for the diffuse component would lead to a large error. We also show an example in Fig. 13 of how our visibility-modulated split-sum approximation can be used in other rendering tasks as a practical way to add self-shadowing to glossy objects.

Figure 13. Rendering of a glossy Happy Buddha.

Failure Cases. One of the limitations of our method is that we rely on good head pose estimation from the initial tracking stage. If the initial tracking is incorrect or imprecise, see (Fig. 14 row 2), a misalignment of the render and the input image (Fig. 14 row 3 column 3) occurs, making the face appear distorted. We tried optimizing head poses in the inverse rendering stage but the results are often jittery over time and the textures are blurrier. Therefore, we decided to rely solely on the initial tracking for the head pose. Note however, that our method still produces reasonable results even when the head poses are inaccurate. In these cases, our method explains the discrepancies between the tracked mesh and the input image using texture.

Although in theory our method can work under arbitrary static lighting conditions, there are challenging cases when our method still does not produce a good enough appearance decomposition. One such example is shown in Fig. 15. In this example, the left side of the face is overexposed while the right side is much darker in all frames. While the render still matches the input image, the reconstructed diffuse albedo and specular intensity maps contain a consid-

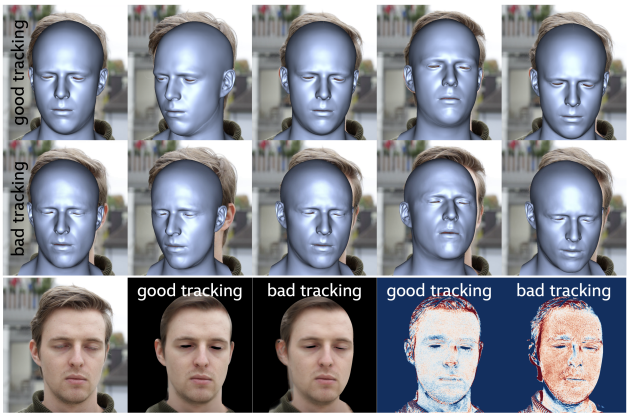

Figure 14. Results from good and bad head pose estimation from the initial tracking stage. The render error maps are displayed with a scale of -0.05 0.05.

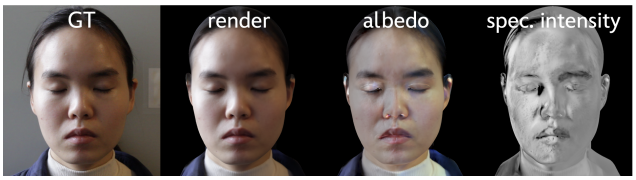

Figure 15. Poor appearance decomposition in challenging lighting conditions. While the render still matches the ground truth image, the albedo map contains some minor baked-in lighting.

erable amount of baked-in lighting. Note however, that our model still manages to disentangle a major part of the lighting from the appearance, *i.e*. the brightness on the left and right sides of the diffuse albedo is similar. Capturing the same subject under multiple different lighting conditions can potentially improve the disentanglement [2], which we leave as future work.

Effects of λ_{geo}. We use $\lambda_{geo} = 19$ based on [50], but this smoothing term can be tuned for different subjects to get better geometry. An ablation for different values of λ_{geo} is shown in Fig. 16. Note that a large λ_{geo} leads to loss of details, but a small λ_{geo} can lead to self-intersections.

Novel Views in the Capture Environment. We show novel view renders in the capture environment in Fig. 17. We primarily focused on the facial regions for this project. Artifacts around the boundaries can be improved with some engineering efforts, such as better masking and better initial alignment. Reconstruction of the hair and shoulders is also an interesting area for future work, which can improve the overall quality.

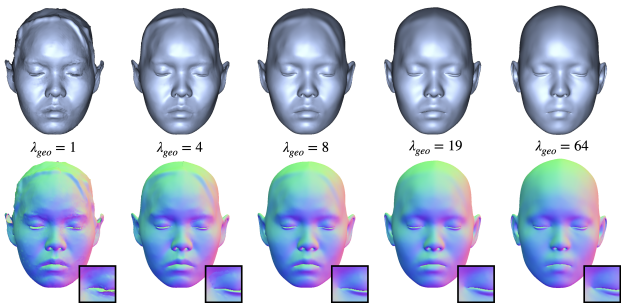

Figure 16. Effects of λ_{geo}. The shaded meshes are shown on the top row with the corresponding normals visualized on the bottom.

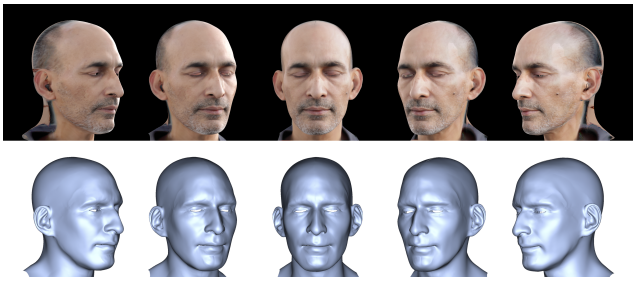

Figure 17. Novel views rendered in the capture environment (top row) along with the corresponding meshes (bottom row).

Comparison with CoRA. CoRA [28] requires a more constrained capture setup with a co-located light and camera in a dark room. We ran our method and CoRA on a subject using the CoRA capture protocol, see Fig. 18, demonstrating that our method achieves equally good reconstruction given the same data. Additionally, our method also works in more generic environments.

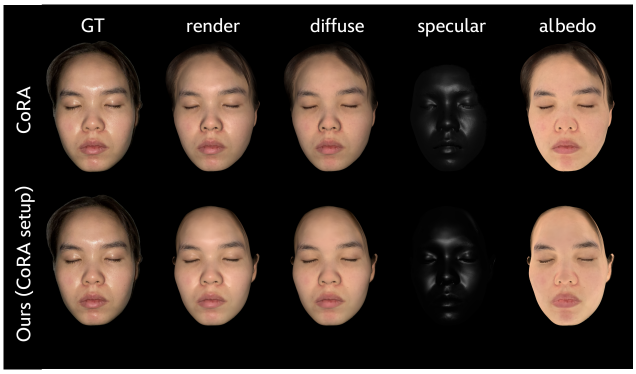

Figure 18. Comparison with CoRA on their data protocol.