Supplementary Materials of Paper 3883

Anonymous CVPR submission

Paper ID 3883

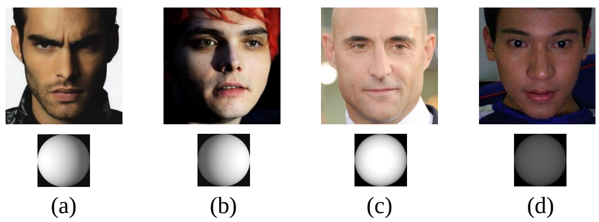

Figure 1. The selected real-world lighting reference images. (a) Left-side lighting. (b) Right-side lighting. (c) Bright frontal lighting. (d) Dark lighting.

1. More Details of Evaluation Protocol

In Section 5.1 of the article, we introduce the evaluation protocol for analysing the predicted geometry and texture. To calculate the metric Cosine-O, we directly compute the cosine similarity between the original image $\mathbf{I}_i$ and the rendered one $\hat{\mathbf{I}}_i$ in the representation of pretrained ArcFace [3] latent space. This metric is also widely-used in LAP [16] and VariTex [2]. To further analyse the robustness and consistency of the image formation procedure, we calculate another two metrics Cosine-P and Cosine-L. Cosine-P is computed between $\mathbf{I}_i$ and a rotated rendered image $\hat{\mathbf{I}}_i^{\omega}$ with a pose ω. For each method, we uniformly sample 13 yaw angles in $[-90°, +90°]$ (every $15°$ once) and 7 pitch angles in $[-30°, +30°]$ (every $10°$ once) for ω, and render 20 images with these poses. Then we calculate the mean cosine similarity of ArcFace representation between these images and the original one. In this way, the metric Cosine-P is able to reveal the robustness of a method on pose variation. For Cosine-L, we relight the rendered image with 4 different lights each of which is predicted by the corresponding method from an unseen real-world reference image (shown in Fig. 1), and calculate the mean cosine similarity of ArcFace representation between relit images and the original one. In this way, the Cosine-L can indicate if the method guarantees a consistent identity under light variation.

2. The Resterization Module

In section 4.2 of the paper, we introduce the proposed rasterization module. Actually, as our texture is not RGB but implicit, we cannot directly use differentiable renderer such as Neural Mesh Renderer (NMR) [10] to perform rasterization for it. Following [2, 15, 16], we use a grid sampling function [9] to solve this problem. First, we use NMR to rasterize only the depth map d_i, obtaining a version $\dot{d}_i = f_R(d_i, \omega_i)$ of the depth map as seen from the input viewpoint. With the warped depth $\dot{d}_i$, we can inverse the function f_R to find the warp field from the observed viewpoint to the canonical viewpoint. Then, with the warp field, we can use grid sampling function f_{sam} [9] to bilinearly sample the shaded canonical implicit texture $\hat{b}_i, \hat{b}_i^c$, obtaining $\dot{b}_i, \dot{b}_i^c$ in the observed viewpoint which is 2D spatially-aligned to the input image $\mathbf{I}_i$.

3. More Implementation Details

To implement the rasterization module, following [15] and [16], we set the Field of View (FOV) as $10°$. Our PhyDIR framework is trained with the loss in Eqn. (7) of the article. Actually, when training the neural reasoning networks Φ^b, Φ^n, we compute $\mathcal{L}_{shape}$ and $\mathcal{L}_l$ using the pretrained 3D proxy. At the early stage of training, we remove the $\mathcal{L}_{tex}$ because the Φ^b has not converged with stable results at this time. After the Φ^b starts to predict reasonable implicit texture maps, we add $\mathcal{L}_{tex}$ to constrain the texture consistency.

At the stage of geometry learning, as described in Sec. 4.4, we use a new Φ^d which contains extra upsampling-conv layers than the corresponding proxy network of Unsup3D [15] or LAP [16]. These extra layers are utilized to upsample the predicted canonical depth to the size of 256×256. We use a same architecture as the proxy to build Φ^{ω}, Φ^l. Φ^{ω}, Φ^l and the new Φ^d can be trained from the weights of proxy or from scratch. During the training of these 3D networks, we freeze the Φ^n and Φ^b, and compute $\mathcal{L}_{shape}$ and $\mathcal{L}_l$ using the pretrained 3D proxy. After this stage, we perform the joint training of all the networks. During the joint training stage, we compute $\mathcal{L}_{shape}$ and $\mathcal{L}_l$ using our 3D networks for self-enhancing and regularizing.

CVPR
#3883

4. Evaluation of State-of-the-art Method

For the methods that report their results on the benchmarks, we directly use the reported numbers if the setting is the same with ours. For other results, we reproduce the method using the official released code, or implement the model with the provided pre-trained weights. For the graphics-renderer-based methods, we use the official code of Unsup3D [15], LAP [16], D3DFR [5] and DECA [6]. Their project pages can be easily found in GitHub. For the neural rendering and 3D-aware generative methods, we compare with the ones with released code, including DFG [4], PIRender [12] and VariTex [2]. All of these methods are very recent state-of-the-art. To make them address the real images, we use a GAN inversion method proposed in Image2StyleGAN [1] with 2000 iterations of each image. This setting is enough for each method to inverse images into their latent space. Note that, there are also other neural rendering methods such as StyleRig [14] and StyleRenderer [11], but either of them release the code or pre-trained weights. This make it very difficult to make a fair comparison with them. Even though, we believe that extensive analyses have been performed in the paper with enough strong baseline approaches.

5. Evaluation of Relighting

As we disentangle and model the facial light, PhyDIR is able to control the lighting effect of the rendered image. In the article, we have made qualitative comparisons with the state-of-the-art methods. Here we perform qualitative evaluations. Following Hou *et al.* [8], we use Multi-PIE [7] dataset. For each Multi-PIE subject and each session, we randomly select one of the 19 images as the source image and one as the target image, which serves as the relighting ground truth. The target image's lighting is predicted by each method, then used to relight the source image. This leads to a total of 921 relit images. Same as [8], we use Si-MSE [17] and DSSIM as the metrics.

Method	Si-MSE	DSSIM
Unsup3D [15]	0.0344	0.2130
LAP [16]	0.0319	0.1978
D3DFR [5]	0.0419	0.3422
DFG [4]	0.0301	0.2015
DPR [17]	0.0282	0.1818
Hou *et al.* [8]	0.0220	0.1605
Ours (Unsup3d-proxy)	0.0238	0.1781
Ours (LAP-proxy)	0.0230	0.1667

Table 1. Relighting evaluation of different methods.

To make a fair comparison, we only calculate the metrics in facial regions for the face modeling methods. The results are illustrated in Table 1, where our method outperform most of the approaches. Note that, the algorithm in [8] is specially proposed for 2D portrait relighting and trained on Multi-PIE, while our method is able to tackle 3D face modeling and is trained on other dataset. Even confronting a more challenging problem, our performance is competitive. These observations further demonstrate our effectiveness on light and face modeling.

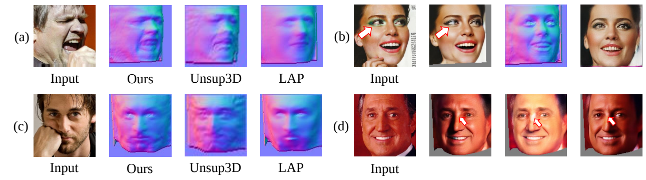

Figure 2. Failure case of our method. (a) Extreme expression. (b) Heavy make-up. (c) Large artifact. (d) Extreme lighting.

6. Limitation & More Results

In the paper, we have shown results on some challenging cases, e.g., faces of large pose/expression, side light and non-Caucasian races. Our method is able to address these conditions. However, for some possible extreme factors, the method may provide unsatisfactory results.

We illustrate several failure cases in Fig. 2. The condition (a) is extreme expression which is challenging for non-parametric methods. Unsup3D [15] and LAP [16] fail on such a case. As our method leverages these two approaches as proxies, it also suffer from lack of 3DMM prior. In Fig. 2-(b), we observe that heavy make-ups also influence the reconstruction results. As some make-ups may bring colors or appearances that hardly appear in the dataset, the method struggles to correctly understand them, predicting improper local details. In Fig. 2-(c), we show the influence of the large artifacts. As no 3DMM assumption is used, the methods struggle to correctly tackle the artifacts. In Fig. 2-(d), we show a image with extreme lighting effect and appearance. As such an effect or appearance hardly appears in the dataset, PhyDIR may provide unusual artifacts when editing the facial lighting condition. In summary, as PhyDIR is data-driven and non-parametric, it learns the statistics of the dataset or domain with out a reliable shape assumption. Besides the aforementioned cases, some other infrequent cases such as giving a light from bottom of a face, very dark conditions or side poses, may also make the method provide imperfect predictions. On the other hand, PhyDIR currently is able to tackle diffuse modeling but specularity modeling, as we use no statistical model such as Albedo MM [13]. Further, as the neural image formation procedure is difficult to analytically described, we cannot guarantee a totally robust rendering process. We further show more results of PhyDIR in Figs. 3 and 4.

References

[1] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan++: How to edit the embedded images? In *CVPR*, pages 8296–8305, 2020. 2

[2] Marcel C Bühler, Abhimitra Meka, Gengyan Li, Thabo Beeler, and Otmar Hilliges. Varitex: Variational neural face textures. *arXiv preprint arXiv:2104.05988*, 2021. 1, 2

[3] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *CVPR*, pages 4690–4699, 2019. 1

[4] Yu Deng, Jiaolong Yang, Dong Chen, Fang Wen, and Xin Tong. Disentangled and controllable face image generation via 3d imitative-contrastive learning. In *CVPR*, pages 5154–5163, 2020. 2

[5] Yu Deng, Jiaolong Yang, Sicheng Xu, Dong Chen, Yunde Jia, and Xin Tong. Accurate 3d face reconstruction with weakly-supervised learning: From single image to image set. In *CVPRW*, 2019. 2

[6] Yao Feng, Haiwen Feng, Michael J Black, and Timo Bolkart. Learning an animatable detailed 3d face model from in-the-wild images. *ACM Transactions on Graphics (TOG)*, 40(4):1–13, 2021. 2

[7] Ralph Gross, Iain Matthews, Jeffrey Cohn, Takeo Kanade, and Simon Baker. Multi-pie. *Image and Vision Computing*, 28(5):807–813, 2010. 2

[8] Andrew Hou, Ze Zhang, Michel Sarkis, Ning Bi, Yiying Tong, and Xiaoming Liu. Towards high fidelity face relighting with realistic shadows. In *CVPR*, pages 14719–14728, 2021. 2

[9] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. *NeurIPS*, 28:2017–2025, 2015. 1

[10] Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada. Neural 3d mesh renderer. In *CVPR*, pages 3907–3916, 2018. 1

[11] Jingtan Piao, Keqiang Sun, Quan Wang, Kwan-Yee Lin, and Hongsheng Li. Inverting generative adversarial renderer for face reconstruction. In *CVPR*, pages 15619–15628, 2021. 2

[12] Yurui Ren, Ge Li, Yuanqi Chen, Thomas H Li, and Shan Liu. Pirenderer: Controllable portrait image generation via semantic neural rendering. In *ICCV*, pages 13759–13768, 2021. 2

[13] William AP Smith, Alassane Seck, Hannah Dee, Bernard Tiddeman, Joshua B Tenenbaum, and Bernhard Egger. A morphable face albedo model. In *CVPR*, pages 5011–5020, 2020. 2

[14] Ayush Tewari, Mohamed Elgharib, Gaurav Bharaj, Florian Bernard, Hans-Peter Seidel, Patrick Pérez, Michael Zollhofer, and Christian Theobalt. Stylerig: Rigging stylegan for 3d control over portrait images. In *CVPR*, pages 6142–6151, 2020. 2

[15] Shangzhe Wu, Christian Rupprecht, and Andrea Vedaldi. Unsupervised learning of probably symmetric deformable 3d objects from images in the wild. In *CVPR*, pages 1–10, 2020. 1, 2

[16] Zhenyu Zhang, Yanhao Ge, Renwang Chen, Ying Tai, Yan Yan, Jian Yang, Chengjie Wang, Jilin Li, and Feiyue Huang. Learning to aggregate and personalize 3d face from in-the-wild photo collection. In *CVPR*, pages 14214–14224, 2021. 1, 2

[17] Hao Zhou, Sunil Hadap, Kalyan Sunkavalli, and David W Jacobs. Deep single-image portrait relighting. In *ICCV*, pages 7194–7202, 2019. 2

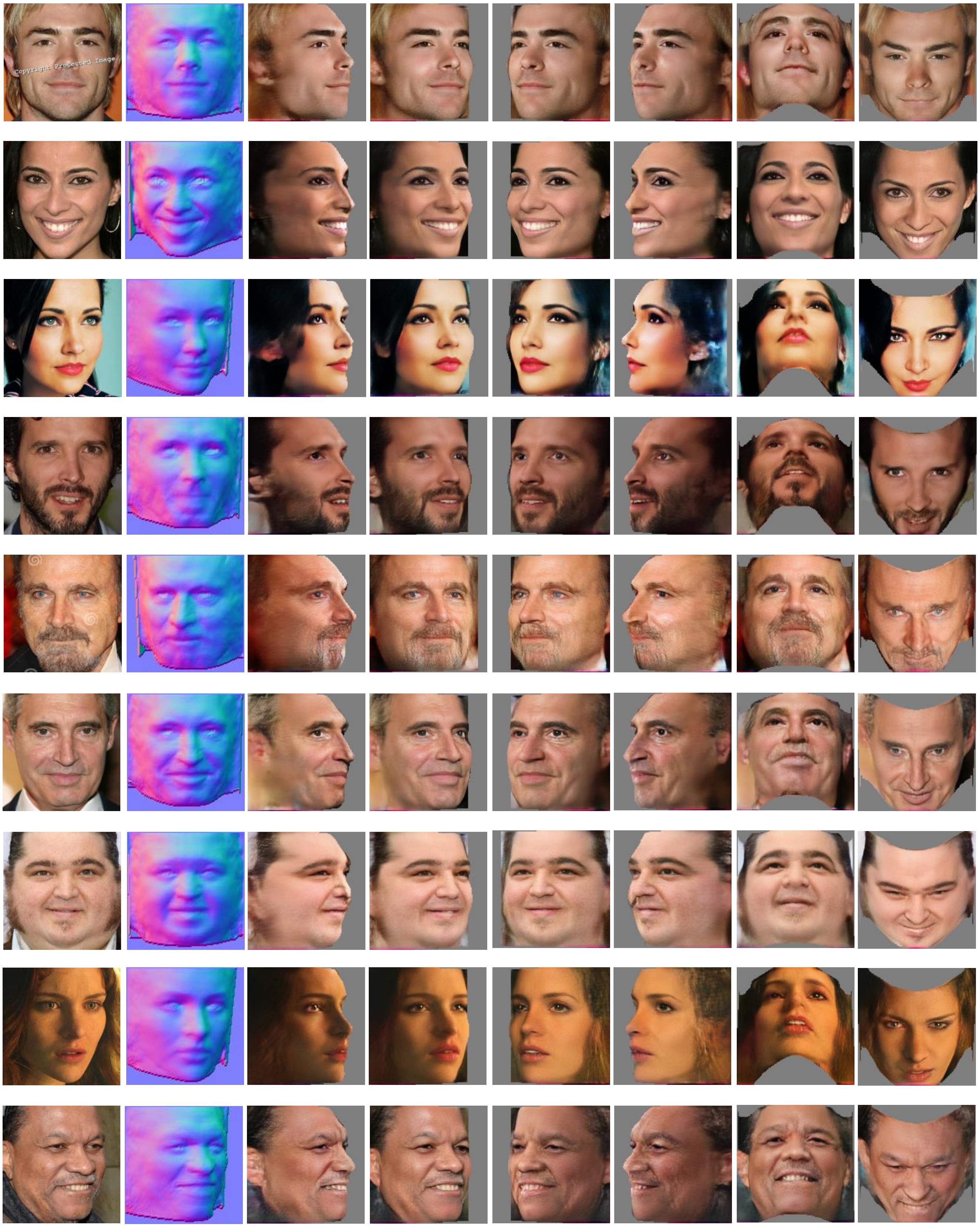

Figure 3. More results on predicted facial shape and pose control.

Figure 4. More results on lighting control.