Supplementary Material for PanoHead: Geometry-Aware 3D Full-Head Synthesis in 360°

Sizhe An[1,2] Hongyi Xu[1] Yichun Shi[1] Guoxian Song[1] Umit Y. Ogras[2] Linjie Luo[1]
[1]ByteDance Inc. [2]University of Wisconsin-Madison

We provide implementation details in Section A, additional experimental details in Section B, more visual results in Section C, and limitations in Section D.

A. Implementation Detail

A.1. Tri-grid

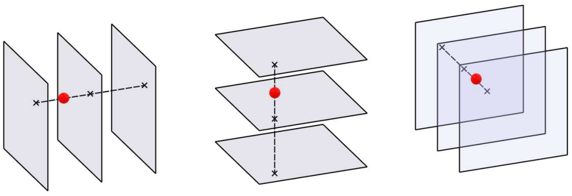

Figure S1. Tri-grid visualization

Figure S1 shows how tri-grid works in all three axes. In practice, for each axis, we format the multiple feature planes as a 5-D tensor ($N \times C \times D \times H \times W$ and $D = 3$) input for the grid_sample function, which leads to a tri-linear interpolation [1].

A.2. Dataset

We train our model with a combination of FFHQ [4] (70K images, $|yaw| \in [0°, 60°]$), K-hairstyle dataset [7] (4K images, $|yaw| \in [120°, 180°]$), and an in-house large-pose head image collection (15K images, $|yaw| \in [60°, 180°]$). For K-hairstyle dataset, we first use WHENet [9] to obtain yaw, pitch, and roll angles. Then the yaw angle is replaced with ground truth horizontal labels provided by the dataset, with adding noises to avoid concrete values. We mirror all the training images and additionally repeat all the back images ($|yaw| > 90°$) by four times for a balanced camera distribution.

A.3. Loss

Our entire pipeline, including generator **G**, neural renderer **R**, and the discriminator **D** is trained using non-saturating GAN loss with R1 regularization [8], following StyleGAN2 [5] and EG3D [1]. Additionally, for training of our foreground-aware tri-discriminator, we regularize the gradient norm of the head segmentation mask with an additional R1 regularization loss $\mathcal{L}_{R1_{mask}}$. We also apply the regularization loss $\mathcal{L}_{cam} = \|\Delta c_{cam}\|^2$ to prohibit far drifting of the self-adapted camera from its original location. More formally, our loss function is shown as follows:

$$\mathcal{L} = \mathcal{L}_{EG3D} + \lambda_{R1_{mask}} \mathcal{L}_{R1_{mask}} + \lambda_{cam} \mathcal{L}_{cam} \quad (1)$$

Our method is implemented using PyTorch 1.12. $\lambda_{R1_{mask}}$ and λ_{cam} are set to 1 and 10, respectively. Empirically we set the camera pose swapping probability to 0.7 instead of 0.5 in the original EG3D with observed better image synthesis quality from non-conditional camera pose.

B. Experimental Analysis

B.1. ID Score

For identity consistency evaluation, we use the identity similarity score (ID) by calculating the average AdaFace [6] cosine similarity score from paired synthesized face images. To this end, we first generate 1000 paired images rendered from different camera poses with the same latent code z. Given an arbitrary pair of images I_p and I_q, we evaluate their Cosine similarity g as:

$$g = \frac{f_p \cdot f_q}{\|f_p\| \|f_q\|}, \quad (2)$$

, where f_p, f_q are the facial feature embeddings for I_p and I_q respectively. We set the random camera poses range at $|yaw| < 45°$ and $|pitch| < 15°$ to have reasonably good quality facial images since AdaFace was trained with facial images dataset.

B.2. MSE of Alpha Matte

Our tri-discrimination enables image generation with disentangled foreground and background. Alpha matte, as known as soft masking, precisely classifies per-pixel image elements such as face, hair, or other classes. To evaluate

[1] https://pytorch.org/docs/stable/generated/torch.nn.functional.grid_sample.html

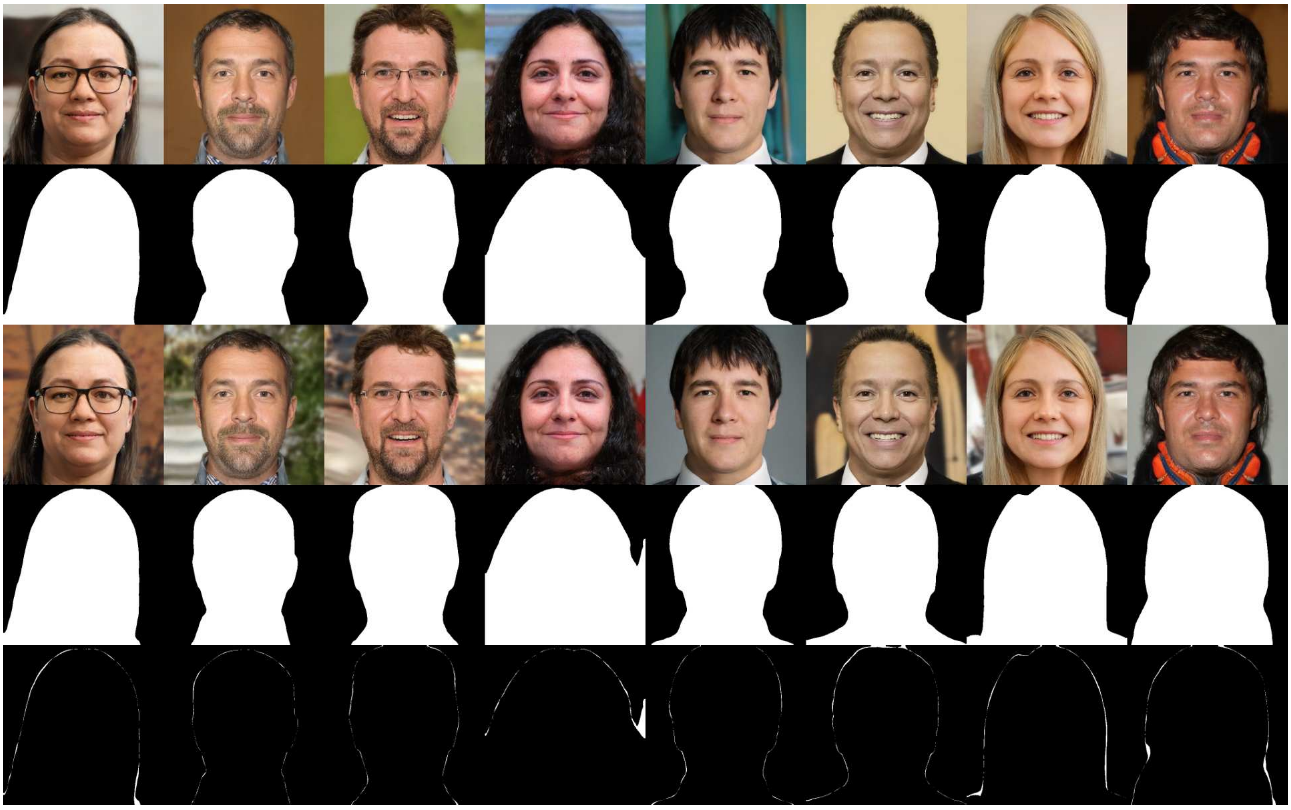

Figure S2. Alpha matte MSE visualization. From top to bottom, it shows (a) images with the first background, (b) corresponding segmentation masks, (c) images with a second different background, (d) corresponding segmentation masks, and (e) pixel-wise differences between two above masks.

PanoHead's capability of fore-background decoupling, we calculate the pixel-wise mean square error (MSE) between alpha matte estimated from a fixed identity while switching the background, as shown in Figure S2. Specifically, a pair of images are synthesized with the same foreground latent code but under different backgrounds. We obtain their head segmentation masks from DeepLabV3 ResNet101 network [2] 'person' class. Pixel-wise MSE of these two masks represents how much a fixed foreground is changed while switching the background. Our metrics are evaluated with 1000 pairs of images.

C. More Results

C.1. Style-mixing application

We show a head style-mixing application using PanoHead's learnt latent space. Specifically, given two latent code w_a and w_b, we concatenate part of layers from w_a and the rest layers from w_b to obtain a style-mixed image. As shown in Figure S3, in PanoHead's latent space, the first four layers of w mainly control the haircut shape, 4th to 8th layers represent the facial appearances, whereas the rest layers change the skin tone.

C.2. View-consistent generation

We show more 360° full head image synthesis in Figure S5, S6 and S7. Our model is able to generate diverse images in terms of genders, races, and appearances. Please also refer to our supplementary video for more high-resolution results.

D. Limitations and Future Work.

In addition to the limitations we mention in the main paper, we provide failure cases visualization in Figure S4. The rows (a) to (d) show artifacts on the back head, with unnatural hair appearance pattern ((a), (d)) or even noisy back-head geometry ((b),(c)). In (e), even though with short frontal hair, our model smoothly transits to a long haircut when rotating to the back. We also observe the appearance of visual artifacts on the face when transiting the camera to large poses. In (f), the model fails to learn a complete hat but interprets the hat as part of the hairstyle. Since our model is trained with cropped head images only, it occasionally struggles to model authentic shoulder area. The flickering texture issue is also noticeable in the original EG3D model. We consider one potential issue is due to the StyleGAN2 [5] synthesis network. Switching to StyleGAN3 [3] architecture would help preserve high-frequency details. We also notice that our model tends to generate perfectly symmetric back head, which is rarely the case in reality. We acknowledge that our model is trained with limited data and styles, thus with underlying inevitable bias. We believe large-scale high-quality well-annotated full-head datasets can resolve most of the aforementioned issues.

References

[1] Eric R Chan, Connor Z Lin, Matthew A Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas Guibas, Jonathan Tremblay, Sameh Khamis, et al. Efficient geometry-aware 3d generative adversarial networks. *CVPR*, 2022. 1

[2] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017. 2

[3] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. *NeurIPS*, 2021. 3

[4] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, 2019. 1

[5] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *CVPR*, 2020. 1, 3

[6] Minchul Kim, Anil K Jain, and Xiaoming Liu. Adaface: Quality adaptive margin for face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18750–18759, 2022. 1

[7] Taewoo Kim, Chaeyeon Chung, Sunghyun Park, Gyojung Gu, Keonmin Nam, Wonzo Choe, Jaesung Lee, and Jaegul Choo. K-hairstyle: A large-scale korean hairstyle dataset for virtual hair editing and hairstyle classification. In *2021 IEEE International Conference on Image Processing (ICIP)*, pages 1299–1303. IEEE, 2021. 1

[8] Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. Which training methods for gans do actually converge? In *International conference on machine learning*, pages 3481–3490. PMLR, 2018. 1

[9] Yijun Zhou and James Gregson. Whenet: Real-time fine-grained estimation for wide range head pose. In *BMVC*, 2020. 1

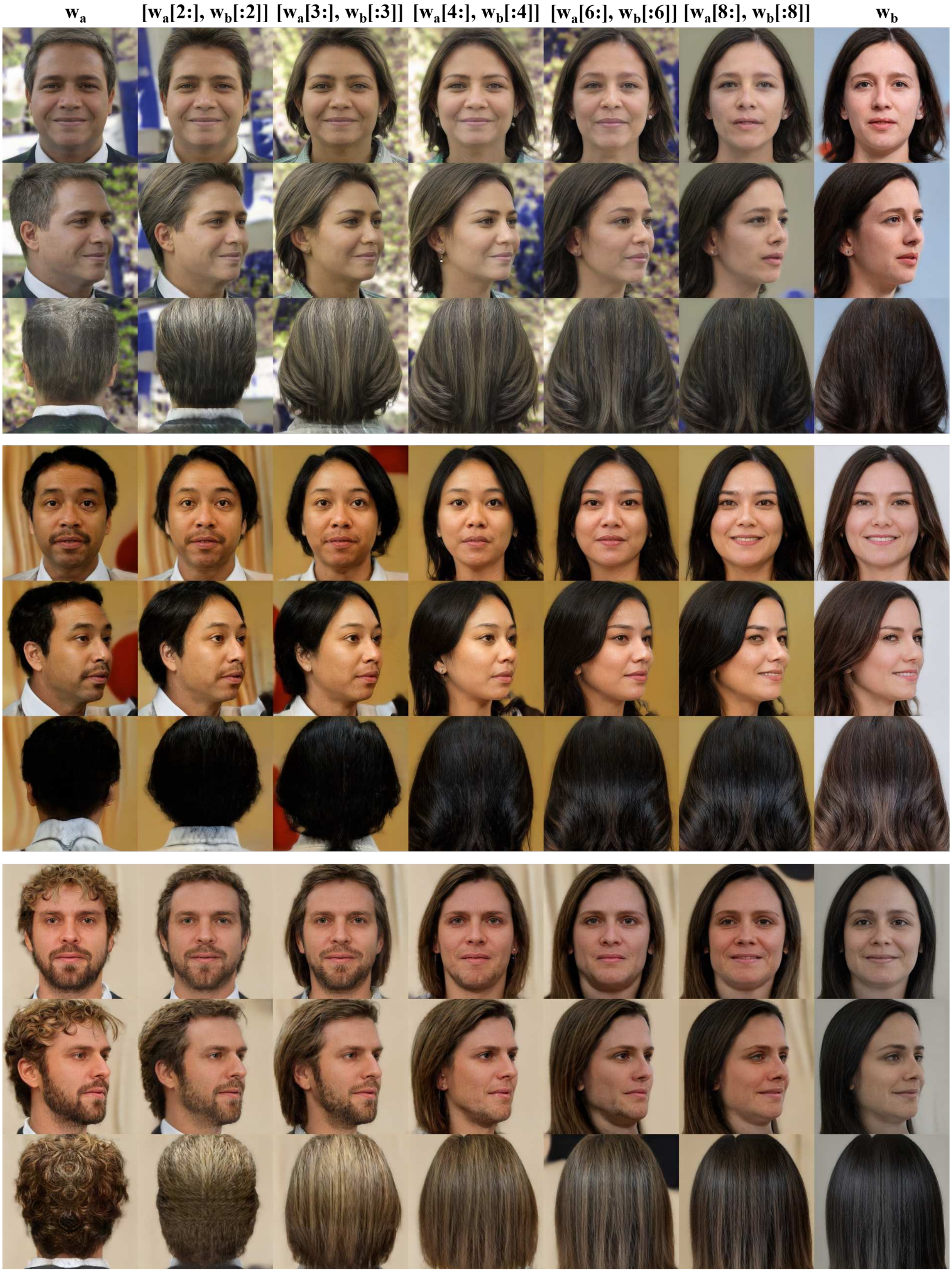

Figure S3. Style-mixing results. Each column represents a head synthesized with the corresponding mixed w latent code at 3 different camera views.

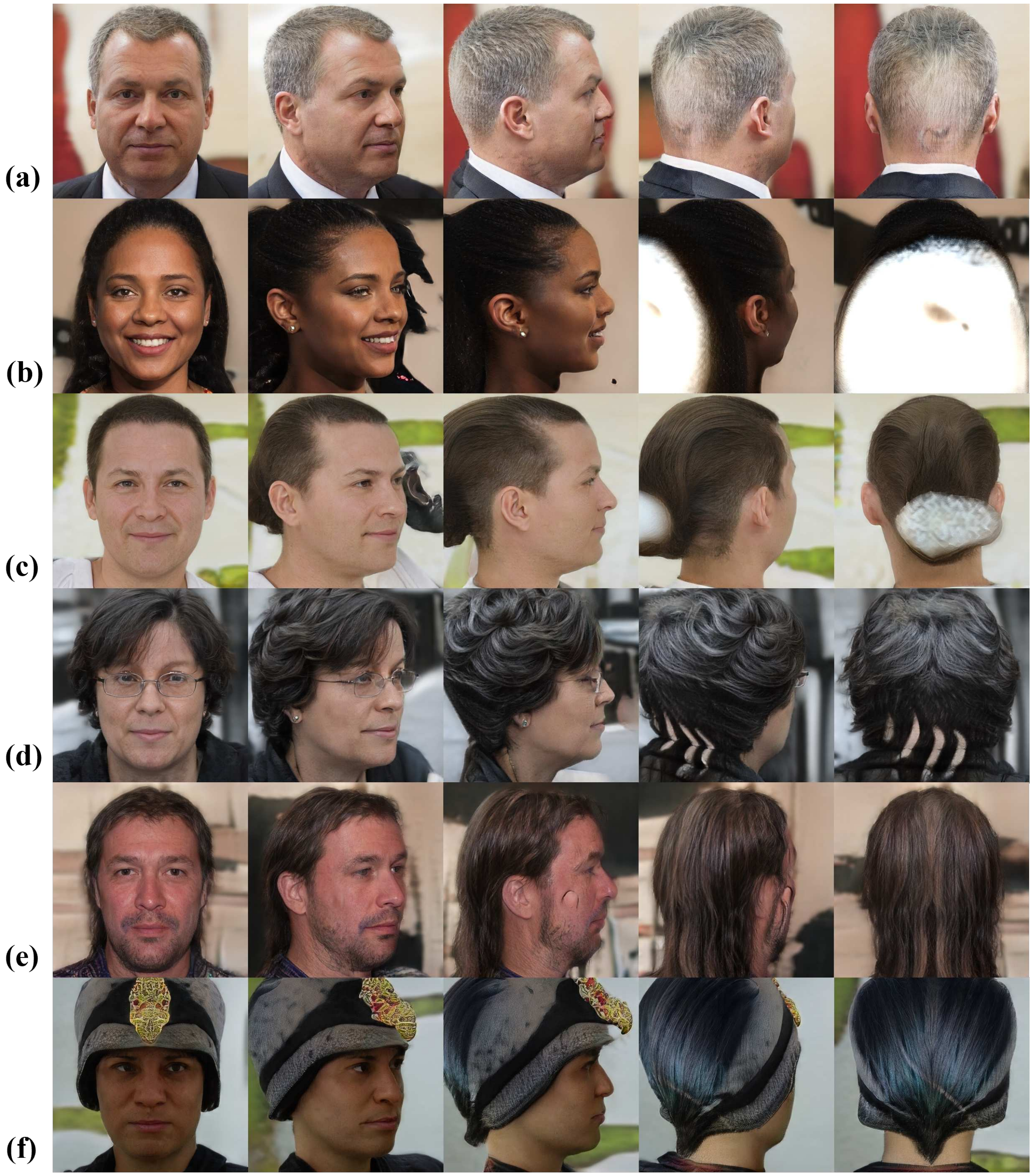

Figure S4. Failure cases. (a) to (d): artifacts on the back head. (e): unnatural mixing of short front and long back haircut. (f): fail to model the complete hat.

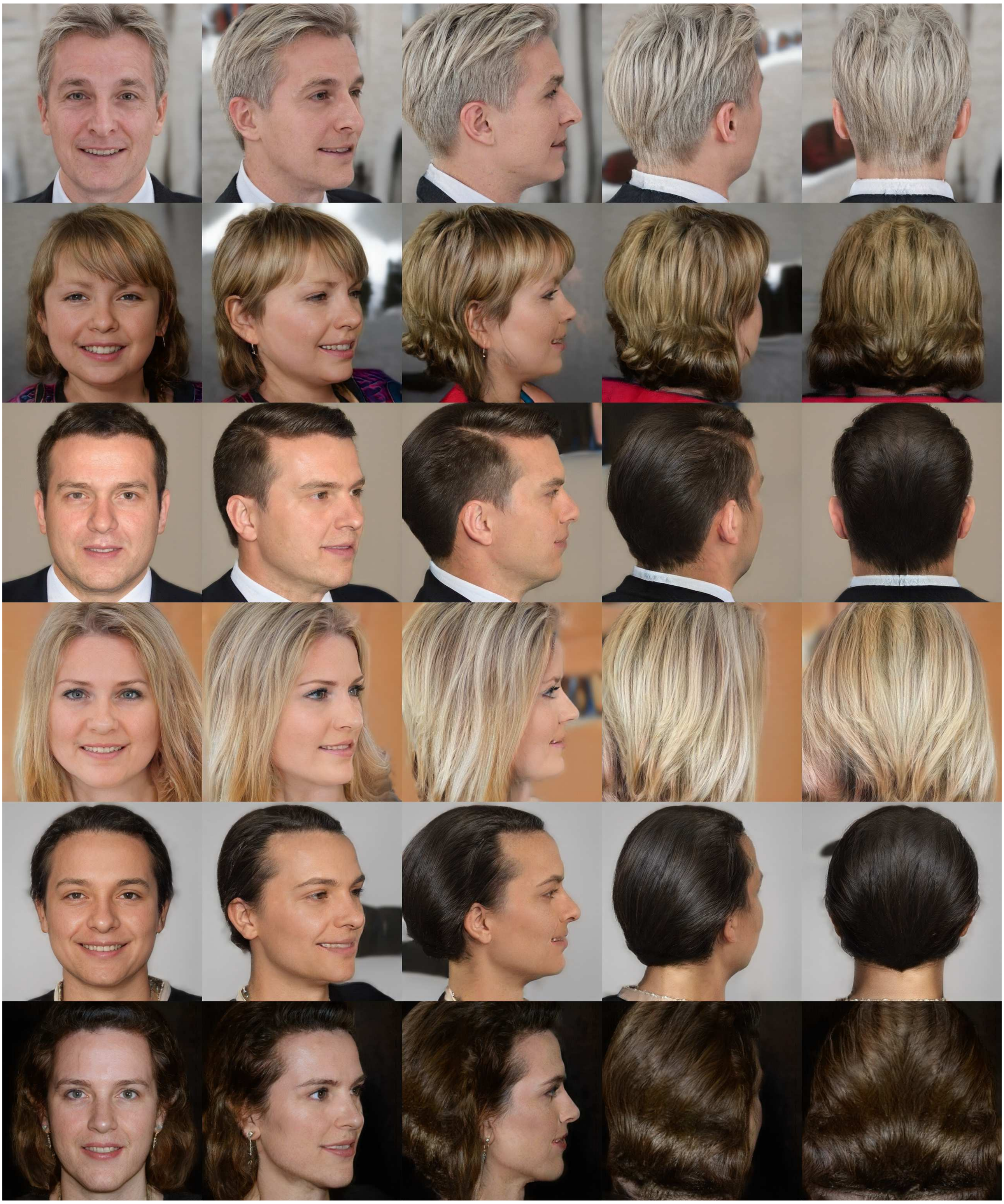

Figure S5. Additional 360° view-consistent full head image synthesis with various hairstyles.

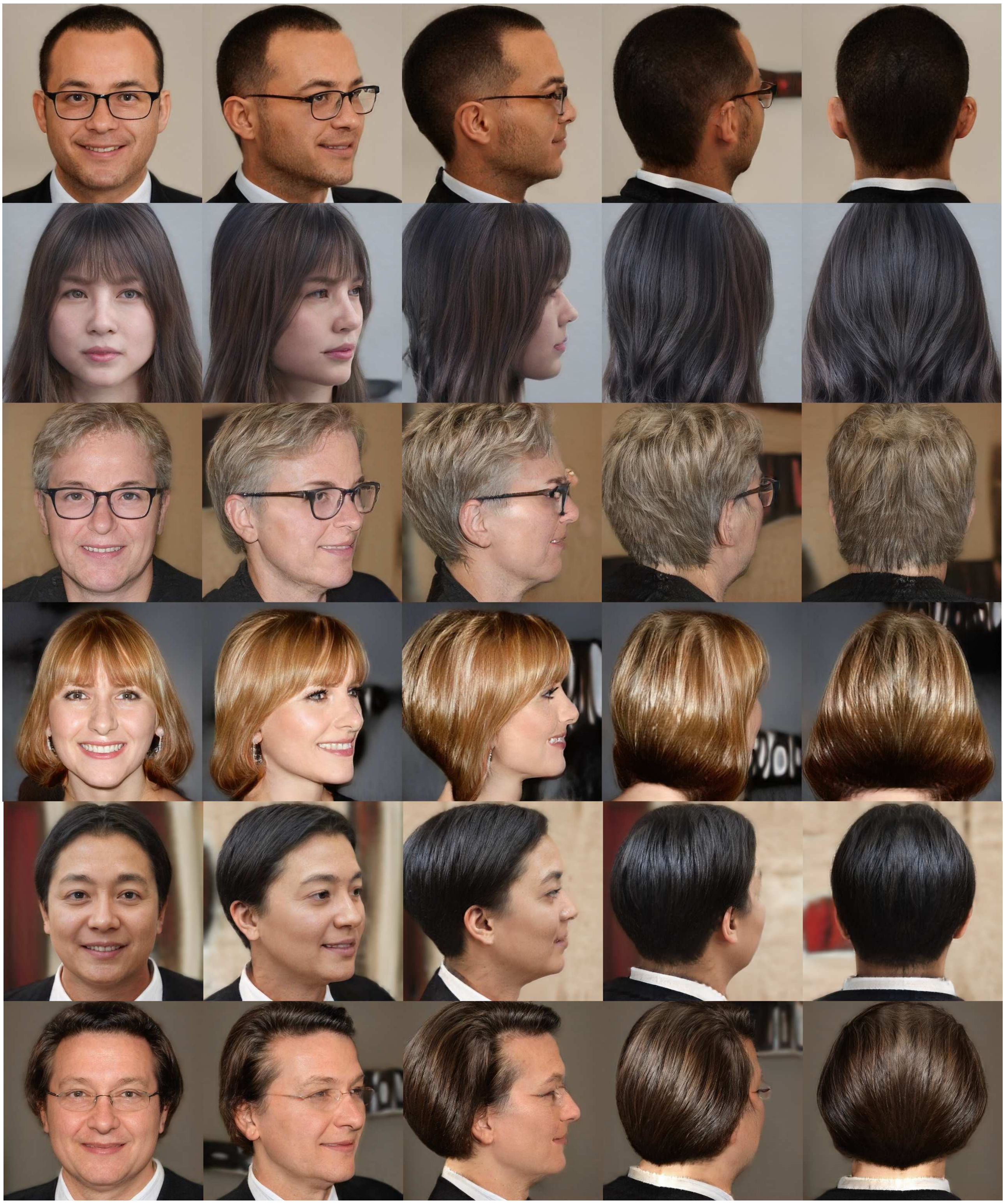

Figure S6. Additional 360° view-consistent full head image synthesis with various hairstyles.

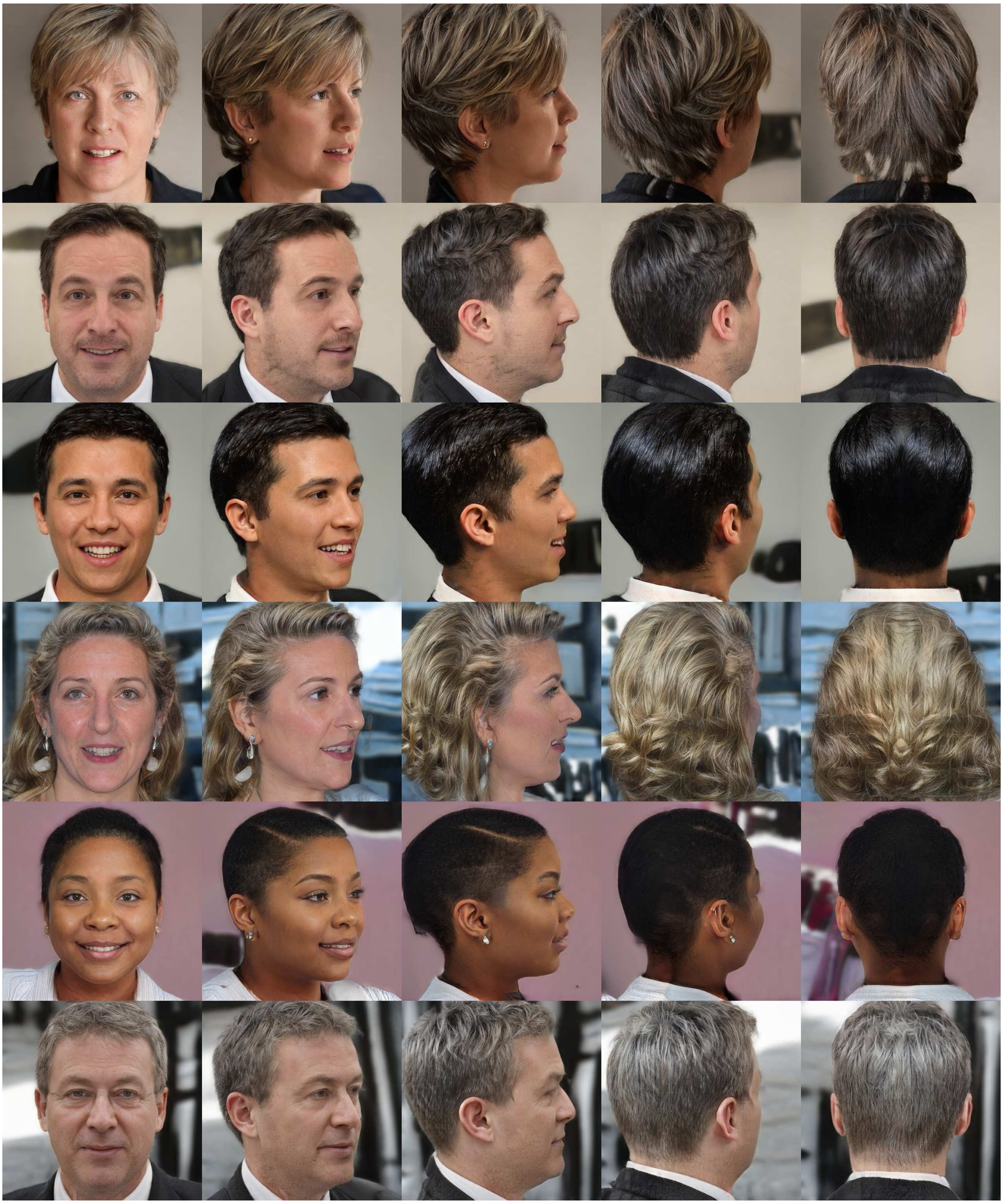

Figure S7. Additional 360° view-consistent full head image synthesis with various hairstyles.