FreeStyleGAN: Free-view Editable Portrait Rendering with the Camera Manifold - Supplemental Materials

THOMAS LEIMKÜHLER and GEORGE DRETTAKIS, Université Côte d'Azur and Inria, France

ACM Reference Format:
Thomas Leimkühler and George Drettakis. 2021. FreeStyleGAN: Free-view Editable Portrait Rendering with the Camera Manifold - Supplemental Materials. *ACM Trans. Graph.* 40, 6, Article 224 (December 2021), 4 pages. https://doi.org/10.1145/3478513.3480538

In this supplemental document we provide multi-view capture instructions, as well as methodology details on camera calibration and geometry reconstruction. We also present specifics of the FFHQ alignment process, how to project manifold coordinates to the valid range and how to sample it, and our definition of a frontal pose. Finally, we elaborate on our two-stage training procedure, provide additional comparisons and ablations, and show a comparison to an encoder-based embedding approach.

1 CAPTURE INSTRUCTIONS

Here we provide the instructions we gave to the non-professional models and photographers who helped us with capturing the datasets.

1.1 Instructions for the Model

- You need to be as static as possible. Sit on a chair and use the backrest (make sure it is not visible above the shoulders).
- Choose a point in front of you to look at during the entire session.
- Make a neutral relaxed face, eyes open, mouth closed. No smiling please.

1.2 Instructions for the Photographer

- Don't use a flash. Avoid casting hard shadows onto the model with your body. Lighting should not be too harsh, shades on the face are fine and even appreciated.
- Have a distance of about 1-2 meters to the model.
- Take 10-25 pictures of the face and full upper body. Frame every view such that that top of the head is slightly below the upper image boundary and the belly button marks the lower image boundary. Please capture:
  - One frontal view.
  - 5-10 views on an ellipse, about 0.25 to 0.5 meters around the frontal view (red in Fig.1).
  - 5-10 views on an ellipse, about 1.0 to 1.5 meters around the frontal view (blue in Fig.1).
  - Add more views at will.
- Capture positions don't have to be exact.

Authors' address: Thomas Leimkühler, thomas.leimkuehler@mpi-inf.mpg.de; George Drettakis, Université Côte d'Azur and Inria, France, george.drettakis@inria.fr.

All results in the paper were produced using our own captures, except for the first row in Fig. 10, which uses material from Milborrow et al. [2010].

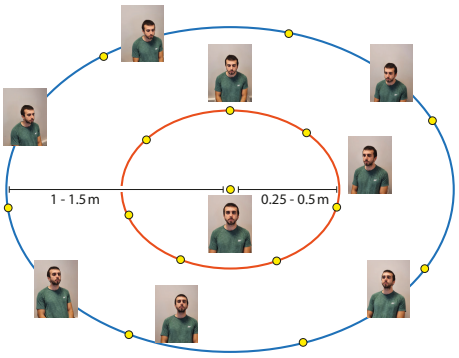

Fig. 1. Figure provided to the photographer to illustrate the capture distribution, encouraging a stratified set of camera poses. Yellow dots mark example capture positions.

2 3D CALIBRATION AND RECONSTRUCTION

We obtain camera calibration and the geometric proxy using off-the-shelf software [CapturingReality 2016]. First, the virtual input cameras are calibrated using structure from motion [Snavely et al. 2006], then a multi-view stereo algorithm estimates the 3D shape using dense pixel correspondences [CapturingReality 2016; Goesele et al. 2007], followed by a meshing step to obtain a triangle mesh of the face. We smooth the mesh [Sorkine 2005] to get rid of high-frequency reconstruction noise. The output of this process are the reconstructed triangle mesh, the calibrated cameras and possibly resampled input images with a lens distortion correction applied. Even though more than 10-25 cameras are usually recommented for 3D reconstruction, quality is high enough for our method, despite taking the photos casually without a rig.

3 REVIEW OF FFHQ ALIGNMENT

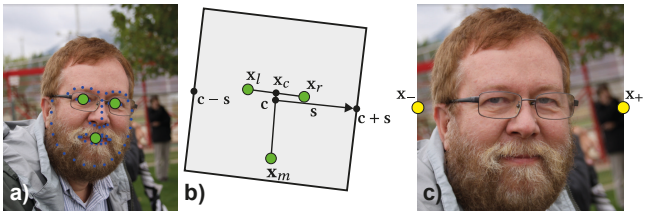

Fig. 2. The 2D alignment performed in the FFHQ dataset. *a)* Raw facial feature points (blue dots) are detected and aggregated to obtain representative eye and mouth positions (green dots). *b)* Geometric features are used to determine the square crop window (grey, not shown to scale) with center $\mathbf{c}$ and vector $\mathbf{s}$ giving orientation and size. *c)* The resulting aligned image.

The FFHQ dataset was constructed by first collecting images from Flickr, followed by a cleanup step and alignment, where first 68 facial features are found [Kazemi and Sullivan 2014] (blue dots in Fig. 2a). Then the eye and mouth features are aggregated to obtain representative eye positions $\mathbf{x}_l$ and $\mathbf{x}_r$, as well as a mouth position $\mathbf{x}_m$ in the image (green dots in Fig. 2a). From these three points a crop window is computed as follows (Fig. 2b): The center of the window is computed as a convex combination of the eye midpoint $\mathbf{x}_c = 0.5\,(\mathbf{x}_l + \mathbf{x}_r)$ and the mouth, as in $\mathbf{c} = \lambda\mathbf{x}_c + (1-\lambda)\mathbf{x}_m$, where $\lambda = 0.9$ in the reference implementation. The eye and mouth positions are also used to determine the orientation of the square crop window:

$$\hat{\mathbf{s}} = \mathbf{x}_r - \mathbf{x}_l + \mathrm{rot}_{90^\circ}(\mathbf{x}_m - \mathbf{x}_c)$$

is used as the horizontal orientation of the crop window, where rot_{90° denotes a counter-clockwise rotation of 90°. The size of the crop window should encompass the entire head, and the heuristic approach in the reference implementation reads as

$$\mathbf{s} = \max\left(2\|\mathbf{x}_l - \mathbf{x}_r\|, 1.8\|\mathbf{x}_m - \mathbf{x}_c\|\right)\frac{\hat{\mathbf{s}}}{\|\hat{\mathbf{s}}\|}.$$

Given the above crop window geometry, the original image is resampled to obtain the final aligned output image (Fig. 2c).

4 MANIFOLD RANGE PROJECTION

Given the rotational components of a manifold coordinate $q = [\theta, \phi]^T$, the closest point q^* in the valid region of the manifold is

$$q^* = \begin{cases} q & \text{if } c_l(\theta) \le \phi \le c_u(\theta) \\ [g, c_u(g)]^T & \text{if } c_l(\theta) > \phi > c_u(\theta) \\ [h_u, c_u(h_u)]^T & \text{if } c_l(\theta) < c_u(\theta) < \phi \\ [h_l, c_l(h_l)]^T & \text{if } c_u(\theta) > c_l(\theta) > \phi \end{cases}$$

where

$$g = \mathrm{sign}(\theta)\sqrt{\frac{b_l - b_u}{a_u - a_l}},$$

$$h_i = \sqrt[3]{\frac{\theta}{4a_i^2} + D_i} + \sqrt[3]{\frac{\theta}{4a_i^2} - D_i}, \quad \text{and}$$

$$D_i = \sqrt{\left(\frac{1 + 2a_i(b_i - \phi)}{6a_i^2}\right)^3 + \left(\frac{\theta}{4a_i^2}\right)^2},$$

with $c_i(\theta) = a_i\theta^2 + b_i$ and $i \in \{u, l\}$.

5 MANIFOLD SAMPLING

To sample the valid range of the camera manifold during training, we employ the inverse-CDF method. We observe that $p(\theta)$ is proportional to the difference of the two bounding parabolas:

$$p(\theta) = \frac{1}{Z}\left(c_u(\theta) - c_l(\theta)\right) = \frac{1}{Z}\left(\Delta a\theta^2 + \Delta b\right),$$

where $\Delta a = a_u - a_l$ and $\Delta b = b_u - b_l$. Further, θ is defined between the intersections of the two bounding parabolas, i. e., $\theta \in [-g, g]$, where $g = \sqrt{-\frac{\Delta b}{\Delta a}}$. Therefore,

$$Z = \int_{-g}^{g}\left(\Delta a\theta^2 + \Delta b\right)\mathrm{d}\theta = 2\left(\frac{\Delta a}{3}g^3 + \Delta b g\right).$$

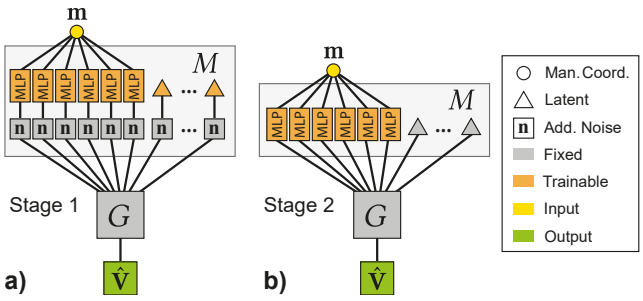

Fig. 3. We employ a progressive training schedule. *a)* In the first stage, we only use the input images as training data. We train MLPs that map manifold coordinates (yellow dot) to the first 6 StyleGAN latents, and we directly optimize for the remaining static latents (orange triangles). All latents fed to G are subject to random perturbations (boxes labelled **n**) during training. *b)* In the second stage, we fine-tune the MLPs by augmenting the training data with IBR and fix the static latents (grey triangles).

The CDF is given by

$$P(\theta) = \int_{-g}^{\theta} p(\theta')\mathrm{d}\theta' = \frac{1}{Z}\left(\frac{\Delta a}{3}\theta^3 + \Delta b\theta\right) + \frac{1}{2}.$$

To obtain a valid manifold sample $[\theta, \phi, d]^T$, we draw three canonical uniform random samples $\boldsymbol{\xi}$. Then $\theta = P^{-1}(\xi_1)$, which we evaluate numerically, and $\phi = \xi_2\,(c_l(\theta) - c_u(\theta)) + c_l(\theta)$. Finally, we linearly remap ξ_3 to obtain d.

6 FRONTAL POSE CALIBRATION

For the definition of the manifold (Sec. 4.1) and its range (Sec. 4.2), as well as the 3D alignment of the face mesh (Sec. 4.4), we need to consistently define how 3D eye and mouth positions are related to manifold coordinates θ and ϕ. For calibration, we consider the frontal pose $\theta = \phi = 0$. The horizontal orientation is straightforward: We set $\theta = 0$ when both eye locations have the same depth. For the vertical orientation, there exists no obvious frontal pose. We therefore set $\phi = 0$ when the depth of the mouth is one eighth of the interocular distance smaller than the depth of the eye midpoint. This configuration is somewhat arbitrary and could be replaced by any suitable alternative. Note, however, that the exact orientation does not change results as long as we are consistent with these definitions.

7 DETAILS ON THE TRAINING PROCEDURE

We found a progressive training schedule, which splits training into two stages, to produce results of highest quality.

7.1 Stage 1: Sparse Input Views

In the first stage, we only use the aligned input views as training data and optimize all trainable parameters (Fig. 3a). Following Karras et al. [2020], we initialize all latents with $\mu_{\mathbf{w}} = \mathbb{E}_{\mathbf{z}}H(\mathbf{z})$. This initializes the optimization with the "average face" in latent space W and is obtained by running 1000 random codes $\mathbf{z}$ through the mapping network H. We run our optimization for 7500 iterations using Adam [Kingma and Ba 2014] with default parameters and a batch size of 2. We start with a learning rate of 0.005 and decay it exponentially

by a factor of 0.98 every 200 iterations. Again following Karras et al. [2020], we allow the optimization to escape local minima by adding stochastic Gaussian random noise to the latents in each training iteration. The optimization starts with a noise standard deviation of $\sigma = 0.1$ and decays as described in Karras et al. We use the following weights for our loss terms: $\lambda_{\text{LPIPS}} = 100$, and $\lambda_{\text{id}} = 1$. Following ideas from Tewari et al. [2020], we vary λ_{prior} over time: For the first 2500 iterations we set $\lambda_{\text{prior}} = 10$ to ensure a reasonable embedding close to W. For the remaining iterations we set $\lambda_{\text{prior}} = 0.1$, which allows the optimization to explore the extended space W^+. Intuitively, this training stage provides sparse anchors for the MLP, which is responsible for pose changes and at the same time optimizes the latents of the static GAN layers with the highest-possible quality training data. Training this stage takes 35 minutes on an NVIDIA RTX6000.

7.2 Stage 2: Dense Manifold

In the second stage, we provide samples from the entire manifold as training data using a mixture of ULR renderings (85%) and input views (15%). Now we fix the latents of the static detail layers to prevent high-frequency IBR artifacts from impacting them (Fig. 3b). Only the MLP weights are refined at this stage, essentially filling in the pose gaps between the input views. We run this stage for 750 iterations with the same optimization parameters as before, but omitting the random noise and using the following (constant) weights for our loss terms: $\lambda_{\text{LPIPS}} = 0$, $\lambda_{\text{id}} = 1.5$, and $\lambda_{\text{prior}} = 0.1$. Disabling the LPIPS loss term is motivated by the fact that it is very sensitive to IBR artifacts and no obvious way exists to incorporate uncertainty akin to Eq. 6 into the multiscale VGG architecture [Liu et al. 2018] without altering its perceptual prediction quality. However, we found $\mathcal{L}_{\ell_1}$ as the sole image quality loss sufficient for this fine-tuning stage. Training this stage takes 4 minutes.

8 ADDITIONAL COMPARISONS

In addition to the comparisons shown in the main paper, we present more details on the comparisons to other image-based rendering methods in Fig. 4 and Tbl. 1. We compute image quality using the PSNR, SSIM [Wang et al. 2004], and E-LPIPS [Kettunen et al. 2019] metrics of the facial region. We see that - not surprisingly - the neural IBR methods win most of the competitions. This comes at the cost of static scenes, which cannot be edited. Due to time constraints, we did not train a separate NeRF++ model for each leave-one-out image set. The numerical results on facial landmarks and image quality are therefore heavily skewed in favor of the method. Please see the supplemental video for novel-view camera paths.

The method of Siarohin et al., which allows semantic editing in the form of facial expressions, does not perform well in the free-viewpoint setting for the metrics we considered. We observe that our approach allows free-view synthesis with camera accuracy in the order of magnitude of the IBR methods, while obtaining decent image quality and facial identity scores – while inheriting the full editing potential of StyleGAN. Note that the comparison to IBR is only possible because our method for the first time allows precice camera control of GAN imagery.

Fig. 4. Comparisons to free-viewpoint rendering methods.

9 LOSS TERM ABLATIONS

Fig. 5 shows the effect of our individual loss terms. Excluding the LPIPS term reduces image sharpness, while the identity loss preserves slight face identity shifts. The prior loss increases photo-realism. We observe that all of our loss terms are necessary to produce results of high fidelity.

10 ENCODER-BASED EMBEDDINGS

General embedding strategies have been explored, which do not require face-specific optimizations to obtain latent codes [Richardson et al. 2021]. While these encoder-based approaches open up exciting research directions, the quality of the resulting embeddings is currently insufficient (see Fig. 6).

REFERENCES

CapturingReality. 2016. RealityCapture. www.capturingreality.com. [accessed 20-July-2020].

Michael Goesele, Noah Snavely, Brian Curless, Hugues Hoppe, and Steven M Seitz. 2007. Multi-view stereo for community photo collections. In *2007 IEEE 11th International Conference on Computer Vision*. IEEE, 1–8.

Table 1. Comparing camera accuracy (measured using facial landmarks), image quality, and face recognition error against free-view rendering methods.

Method	Semantic Editing	Facial Landmarks		Image Quality			Recognition Error↓
		Alignment↓	Detection Rate↑	PSNR↑	SSIM↑	E-LPIPS↓	
[Hedman et al. 2018]	✗	.023	99%	26.4	.875	.010	.07
[Riegler and Koltun 2020]	✗	.027	100%	23.6	.845	.013	.08
[Zhang et al. 2020][1]	✗	.018	100%	30.6	.853	.015	.24
[Siarohin et al. 2019]	✓[2]	.254	42%	11.7	.531	.061	.23
Ours	✓	.068	100%	20.9	.717	.017	.14

[1] Due to time constraints, we did not train a separate model for each leave-one-out image set, but only one model using all images per subject.
[2] Editing is restricted to facial animations using a driving video.

Fig. 5. Loss term ablation. Only our full loss formulation gives best results.

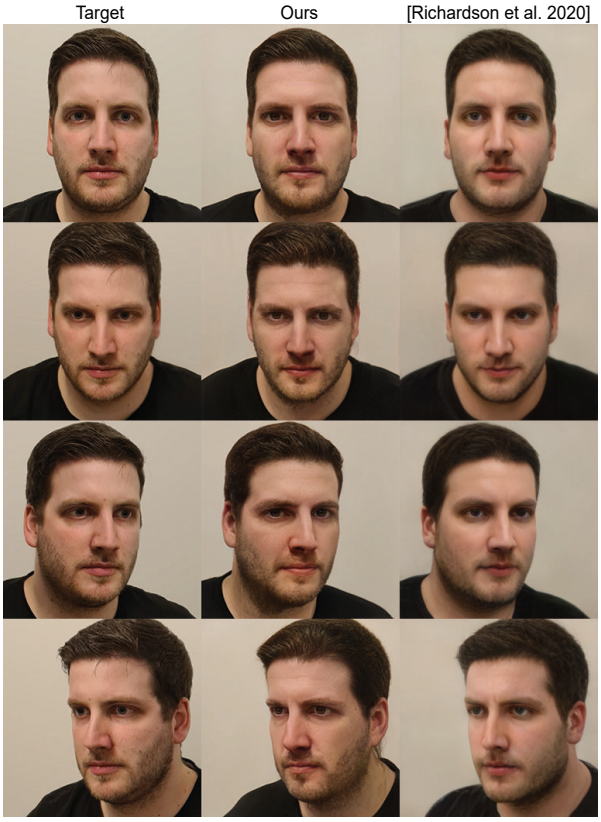

Fig. 6. Comparing our optimization-based embedding strategy (center) with the state-of-the-art encoder-based method of Richardson et al. [2021] (right). We observe that our approach produces a more faithful reconstruction of the target (left).

Peter Hedman, Julien Philip, True Price, Jan-Michael Frahm, George Drettakis, and Gabriel Brostow. 2018. Deep blending for free-viewpoint image-based rendering. *ACM Transactions on Graphics (TOG)* 37, 6 (2018), 1–15.
Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. 2020. Analyzing and improving the image quality of stylegan. In *CVPR*. 8110–8119.
Vahid Kazemi and Josephine Sullivan. 2014. One millisecond face alignment with an ensemble of regression trees. In *CVPR*. 1867–1874.
Markus Kettunen, Erik Härkönen, and Jaakko Lehtinen. 2019. E-LPIPS: robust perceptual image similarity via random transformation ensembles. *arXiv preprint arXiv:1906.03973* (2019).
Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
Guilin Liu, Fitsum A. Reda, Kevin J. Shih, Ting-Chun Wang, Andrew Tao, and Bryan Catanzaro. 2018. Image Inpainting for Irregular Holes Using Partial Convolutions. In *Proceedings of the European Conference on Computer Vision (ECCV)*.
S. Milborrow, J. Morkel, and F. Nicolls. 2010. The MUCT Landmarked Face Database. *Pattern Recognition Association of South Africa* (2010). http://www.milbo.org/muct.
Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or. 2021. Encoding in style: a stylegan encoder for image-to-image translation. In *CVPR*. 2287–2296.
Gernot Riegler and Vladlen Koltun. 2020. Free view synthesis. In *ECCV*. 623–640.
Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. 2019. First Order Motion Model for Image Animation. In *NeurIPS*.
Noah Snavely, Steven M Seitz, and Richard Szeliski. 2006. Photo tourism: exploring photo collections in 3D. In *ACM Siggraph 2006 Papers*. 835–846.
Olga Sorkine. 2005. Laplacian mesh processing. *Eurographics (STARs)* 29 (2005).
Ayush Tewari, Mohamed Elgharib, Mallikarjun BR, Florian Bernard, Hans-Peter Seidel, Patrick Pérez, Michael Zöllhofer, and Christian Theobalt. 2020. PIE: Portrait Image Embedding for Semantic Control. *ACM Transactions on Graphics (TOG, Proc. SIGGRAPH Asia)* 39, 6.
Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing* 13, 4 (2004), 600–612.
Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. 2020. NeRF++: Analyzing and Improving Neural Radiance Fields. arXiv:2010.07492 [cs.CV]