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

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

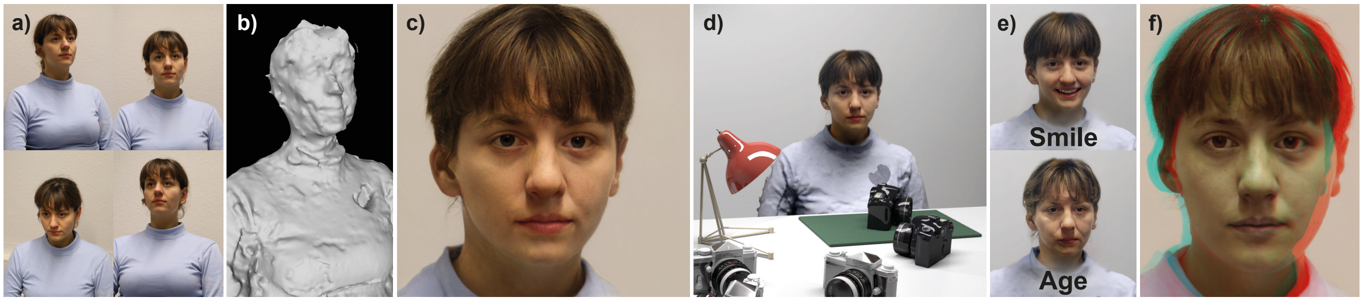

Fig. 1. We introduce a new approach that generates an image with StyleGAN defined by a *precise 3D camera*. This enables faces synthesized with StyleGAN to be used in 3D free-viewpoint rendering, while also allowing all semantic editing provided by GAN methods. Our method takes as input multiple views of a person (examples in *a*), used to reconstruct a coarse 3D mesh (*b*). To render a novel view, we identify the closest camera which corresponds to an image that StyleGAN can generate (*c*). We lift this view to 3D and obtain free-viewpoint renderings with arbitrary camera models, which allows the integration of our renderings into synthetic 3D scenes (*d*). We inherit the high-quality semantic editing capabilities from StyleGAN (*(e)* smile or aging), and enable stereoscopic rendering (*f*). Our method can be integrated into any rendering pipeline and - for the first time - marries generative image modeling with traditional rendering.

Current Generative Adversarial Networks (GANs) produce photorealistic renderings of portrait images. Embedding real images into the latent space of such models enables high-level image editing. While recent methods provide considerable semantic control over the (re-)generated images, they can only generate a limited set of viewpoints and cannot explicitly control the camera. Such 3D camera control is required for 3D virtual and mixed reality applications. In our solution, we use a few images of a face to perform 3D reconstruction, and we introduce the notion of the GAN *camera manifold*, the key element allowing us to precisely define the range of images that the GAN can reproduce in a stable manner. We train a small face-specific neural implicit representation network to map a captured face to this manifold and complement it with a warping scheme to obtain free-viewpoint novel-view synthesis. We show how our approach – due to its precise camera control – enables the integration of a pre-trained StyleGAN into standard 3D rendering pipelines, allowing e.g., stereo rendering or consistent insertion of faces in synthetic 3D environments. Our solution proposes the first truly free-viewpoint rendering of realistic faces at interactive rates, using only a small number of casual photos as input, while simultaneously allowing semantic editing capabilities, such as facial expression or lighting changes.

CCS Concepts: • **Computing methodologies** → **Image-based rendering**; **Neural networks**.

Additional Key Words and Phrases: Portrait editing, Camera models

ACM Reference Format:
Thomas Leimkühler and George Drettakis. 2021. FreeStyleGAN: Free-view Editable Portrait Rendering with the Camera Manifold. *ACM Trans. Graph.* 40, 6, Article 224 (December 2021), 15 pages. https://doi.org/10.1145/3478513.3480538

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

1 INTRODUCTION

Recent Generative Adversarial Networks (GAN) can generate stunningly realistic images of faces from a latent code [Karras et al. 2019, 2020b]. Combined with recent methods to project a real photo to such a latent code [Abdal et al. 2019; Tewari et al. 2020b] they also allow semantic image editing, e.g., controlling facial expression or relighting [Abdal et al. 2021; Deng et al. 2020; Härkönen et al. 2020]. This opens the ground-breaking potential of replacing the expensive and complex process of capturing [Debevec et al. 2000], animating and rendering human faces [Seymour et al. 2017] by simply using GAN-generated images based on photos. Unfortunately, two central components are missing for this process to become reality: currently there is no way to consistently generate a realistic GAN image of a person in a Computer Graphics (CG) shot as defined by a *precise and complete 3D camera* (3D position, 3D rotation, and field of view), and current methods can only generate a limited set of viewpoints. We present the first solution that solves these problems, allowing highly realistic, *editable* faces to be correctly rendered for a any given 3D camera, and thus seamlessly integrated into standard CG imagery. Our method opens the way to using such highly realistic, editable face renderings for a wide range of applications such as virtual or augmented reality, immersive video-conferencing or even immersive remote training.

We build on the StyleGAN architecture [Karras et al. 2019, 2020b] that is inherently a 2D image generator. Recent methods [Abdal et al. 2021; Deng et al. 2020; Tewari et al. 2020a] can render head poses parameterized by two angles, but have no way to generate an image for a specific and complete 3D camera, ignoring at least five degrees of freedom (DoF). This results in a small subspace of 3D camera poses; the nature and limits of this subspace have never been precisely defined, much less extended.

Our first challenge is thus to characterize the subspace of 3D camera parameters that StyleGAN can represent, enabling the design of a method for free viewpoint face rendering. We analyze this subspace, carefully defining what we call the *camera manifold*. This definition is inherently linked to the alignment step performed on the face database used to train the StyleGAN portrait model, severely constraining the 2D locations of eyes and mouth. We also determine the boundaries of the camera manifold, allowing us to project a free-view camera onto it.

Providing full 3D capabilities for StyleGAN requires at least some 3D information; we use a few (10-25), casually captured photos of a face as input and generate coarse face geometry and calibrated cameras to guide part of our method.

Given these photos of a person, our second challenge is finding the latent code to provide to StyleGAN so it can generate the corresponding image. To do this, we need our camera manifold: we first find the closest view *on the manifold*, then train a small, per-face latent representation network to find the latent vector for a given manifold view. We train the network with calibrated input views and image-based renderings using the reconstructed face geometry.

However, many free-viewpoint camera poses are not on the manifold. This raises our third and final challenge, i.e., to provide a method allowing any such view to be synthesized. We do this by warping from the closest camera on the manifold to the desired novel view, using the coarse reconstructed geometry.

These three steps allow us to introduce a fully-operational, interactive system with *precise 3D camera control* while fully exploiting StyleGAN-quality photorealistic face synthesis and corresponding manipulations (see Fig. 1). We consider a pre-trained and fixed StyleGAN2 model in this work, which allows us to run our per-face pre-processing pipeline within 45 minutes on a single GPU.

In summary our contributions are:

- An in-depth study and quantitative definition of the subspace of camera poses that StyleGAN can robustly synthesize, that we call *camera manifold*.
- A method to generate a realistic StyleGAN face based on a precisely defined 3D camera pose on the manifold.
- A warping scheme to render any camera pose freely defined in 3D, fully consistent with semantic editing methods.

Our system allows interactive synthesis of realistic faces, allowing free-viewpoint navigation in 3D while moving around a face casually captured with a handful of photos. We demonstrate interactive sessions with several captured faces, also performing semantic edits in a view-consistent manner – such as changing facial expression, opening/closing eyes/mouth – and also camera model manipulation (e.g., the Vertigo effect), seamless integration with synthetic 3D environments, and the first ever stereoscopic GAN renderings.

2 RELATED WORK

Our solution touches on several vast domains: Generative adversarial networks (GANs), image-based rendering (IBR), face models and portait rendering, and constrained camera models. In what follows, we only review the most closely related work to ours.

2.1 Image Synthesis and Editing with GANs

Generative adversarial networks [Goodfellow et al. 2014] build a statistical model trained to mimic the distribution of training data (often images). In just a few years, GANs have evolved to produce truly photorealistic images at high resolutions. Currently, StyleGAN [Karras et al. 2020a, 2019, 2020b] marks the state of the art in unconditional image generation in narrow domains such as faces, cars or cats. Importantly, a single forward pass through the generator provides photo-realistic imagery, enabling interactive rendering.

Latent codes represent semantically meaningful and reasonably disentangled concepts for many neural models [Karras et al. 2019; Radford et al. 2016; Upchurch et al. 2017]. In StyleGAN, an informed manipulation of latents results in high-level image changes, e.g., in pose, lighting, facial expression, age, gender, etc. The manipulations can correspond to linear [Härkönen et al. 2020; Shen et al. 2020; Tewari et al. 2020a] or non-linear [Abdal et al. 2021; B R et al. 2021] paths in latent space, while disentangled controls can be jointly trained with the generator [Deng et al. 2020]. Our approach is fully compatible with semantic manipulation, such as the method of Härkönen et al. [2020], allowing us to dynamically edit our renderings.

All methods that rely on latent manipulations to change semantic attributes are necessarily bound to the span of the training data, especially when an alignment step is used [Jahanian et al. 2020]. In practice, this means that only a subset of possible camera poses can be synthesized by StyleGAN. Portrait alignment restricts cameras to 3 DoF: two rotations and field of view. While previous work considers only the two rotations, ours is the first method that allows free view navigation with complete (7+ DoF) camera models.

Recent work recovers latent codes from a *single* image [Abdal et al. 2019; Richardson et al. 2021; Zhu et al. 2016] enabling semantic editing of real photos with unprecedented quality [Abdal et al. 2020; Tewari et al. 2020b]. In contrast we are the first to devise a parameterized embedding of multiple views of the same face, which requires a mapping from 3D cameras to latents.

The StyleGAN latent space can be used to represent *any* image with high fidelity – even ones far outside the training data distribution [Abdal et al. 2019] – at the cost of low-quality edits. Consequently, researchers have investigated regularizations to enforce latents closer to the original distribution. This has been done by considering the distribution of the latents directly [Tewari et al. 2020b; Wulff and Torralba 2020; Zhu et al. 2020], or by utilizing semantic knowledge about the images generated [Richardson et al. 2021]. We use a prior as well to find a good compromise between photorealism and identity preservation.

In recent work, GANs have been used as a multi-view generator to aid inverse rendering [Zhang et al. 2021a], or to estimate 3D cameras, shape, and lighting [Pan et al. 2021; Shi et al. 2021]. A different line of work has devised GANs to incorporate 3D information directly, e.g., using 3D geometry [Zhu et al. 2018] or 3D semantic occupancy [Chen et al. 2021] followed by image-to-image translation for final output, or using volumetric generator layers followed by projection for rotations and scaling [Nguyen-Phuoc et al. 2019]. GRAF [Schwarz et al. 2020] and pi-GAN [Chan et al.

2021] produce a 3D radiance field, which can then be rendered using volume rendering. Such GANs with explicit 3D-awareness are promising, but so far fail to generate high-quality results. In contrast to these solutions, we lift a pre-trained 2D GAN to 3D, and thus inherit the advantages of 2D GANs (high quality, high resolution, robust training procedures, etc.) while enhancing the method with precise 3D camera control. 3D-aware generative modeling can be improved by jointly learning the distribution of camera parameters [Niemeyer and Geiger 2021]. Similarly, we estimate this distribution for a pre-trained StyleGAN model to refine our camera manifold.

2.2 Multi-view Free-viewpoint Image-based Rendering

Traditional IBR usually relies on explicit proxy geometry from structure-from-motion/multi-view stereo [Schonberger and Frahm 2016]: Unstructured lumigraph rendering (ULR) [Buehler et al. 2001] uses a geometric proxy to heuristically blend multi-view images. We use this method for training. Recently, *neural rendering* has made significant advances in the quality of novel view synthesis [Tewari et al. 2020c]. Geometry-based methods use deep learning to improve the quality of novel view synthesis [Hedman et al. 2018; Riegler and Koltun 2020], but are restricted to static scenes. Single-object mesh-based neural rendering approaches [Thies et al. 2019a,b] also use proxy geometry for reprojection followed by neural refinement. We share the methodology of using a geometric proxy, and – like these solutions – enable true free-viewpoint rendering. However, we are the first to exploit geometry to steer a GAN for free-viewpoint face rendering.

Neural Volumes [Lombardi et al. 2019] learn a volumetric scene representation, which is rendered using ray marching. They can handle video sequences, but require synchronized and calibrated cameras. Flexible neural scene representations allow high-level scene editing [Li et al. 2020], but they are restricted to changes captured in the input views. In contrast, our approach exploits the expressive space of variations captured in a generative model.

Recent solutions include neural radiance fields [Mildenhall et al. 2020; Zhang et al. 2020], that use implicit scene representations [Sitzmann et al. 2019] and volume rendering techniques for view synthesis. Extensions allow deforming objects [Park et al. 2021] or changing lighting [Martin-Brualla et al. 2021; Srinivasan et al. 2021], offering some degree of semantic control [Zhang et al. 2021b]. These methods have long rendering times (usually tens of seconds per frame) and most of them need dense capture; in contrast, we exploit the power of GANs allowing interactive rendering and use of sparse capture (10-25 images).

2.3 Face Models and Portrait Rendering

Manually created face rigs [Seymour et al. 2017] are stunningly realistic, but require skilled visual effects artists working long hours. 3D morphable models (3DMMs) [Blanz and Vetter 1999; Egger et al. 2020] are generative models that allow fine-grained control over shape and texture of an object, often used for faces. Such models can be augmented with deformation-dependent texture maps [Matthews and Baker 2004] and photometric capture to also model appearance variation [Smith et al. 2020]. 3DMMs offer maximal control over shape, facial expression, texture, etc., but tend to lack photo-realism and typically only model the face region (hair, neck, etc. are not included) even though recent advances [Yenamandra et al. 2021] are starting to consider entire heads.

Another approach to capturing faces involves complex multi-view setups [Beeler et al. 2010; Bi et al. 2021; Ghosh et al. 2011; Lombardi et al. 2018; Wei et al. 2019]. In contrast to these methods, we only need a few casually captured images as input and can perform arbitrary semantic edits.

Face re-enactment is an active area of research. Kim et al. [2018] transfer head pose and facial expressions from a source video to target video. Single [Geng et al. 2018; Siarohin et al. 2019; Zakharov et al. 2020] or multiple [Wang et al. 2019; Zakharov et al. 2019] source images can be used in conjunction with facial feature point extraction of a target sequence to hallucinate novel views and facial expressions. Most of these methods have only been demonstrated with very limited pose variation. Our approach allows true free-viewpoint navigation while reasonably preserving identiy.

Rao et al. [2020] shares some similarities with our approach, since they also process facial animations in a canonical frame, and lift the face to 3D. However, they use StyleGAN to synthesize only the mouth, and require dense capture.

Face rotation using a single image [Nagano et al. 2019; Xu et al. 2020; Zhou et al. 2020] usually proceeds by first estimating geometry, followed by a neural rendering step. While results are impressive given that only a single image is used, the variation of attainable poses tends to be limited. Subtle perspective corrections of portraits – including stereo image synthesis – can be addressed using image warping [Fried et al. 2016; Zhao et al. 2019]. We also use warping, but mostly for global transformations, while perspective effects are generated by manipulating latent codes, successfully overcoming typical problems arising from disocclusions, hair, and the lack of global consistency.

Specialized variants of neural radiance fields for portrait synthesis [Gafni et al. 2021; Wang et al. 2021] produce free-viewpoint results of high visual quality and can handle facial animations, but inherit the problems of neural radiance fields: dense capture setup and long rendering times. Gao et al. [2020] combine neural radiance fields with strong priors to allow portrait rendering from a single view, achieving some effects we handle (e.g., the Vertigo-effect), but the range of achievable poses and visual quality are limited.

2.4 Constrained Cameras

Camera control in interactive applications is important for viewpoint computation, motion planning and editing. See Christie et al. [2008] for an in-depth survey. Our camera manifold builds on fixed projected locations, related to 2-object composition problems [Blinn 1988]: How to place a camera such that two objects are at prescribed locations in the image plane? Lino and Christie [2012] observed that the solution to this problem is a manifold surface in camera parameter space; they later generalized this to the *Toric space* [Lino and Christie 2015], allowing interactive exploration and optimization of viewpoints with multiple geometric constraints. Similarly, we parameterize the 7 DoF camera space using a 3D manifold. However, their approach is more general, with a more complex solution space,

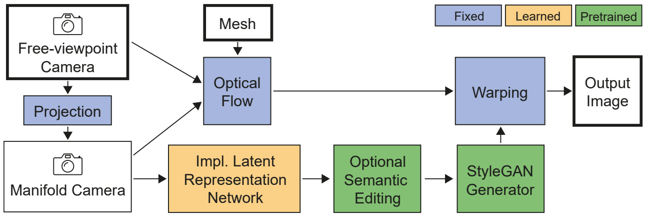

Fig. 2. Overview of our method.

while our simpler problem has more constraints and is amenable to efficient convex optimization.

3 OVERVIEW

Our method takes as input 10-25 photographs of a person and allows free 3D viewpoint rendering of the face with arbitrary camera models based on a pre-trained StyleGAN2 portrait model [Karras et al. 2020b]. An overview of our method is shown in Fig. 2.

3.1 Method

Our key observation is that the StyleGAN portrait model can synthesize a quantifiable range of pose variations: All generated images have the property that the 2D locations of eyes and mouth are severely constrained. Based on this observation, we define a space of camera parameters we call the *camera manifold*, which results in images in the canonical configuration that respects this constraint (Sec. 4). We then analyze and delimit the *range* of the manifold, i.e., the subspace of camera poses *within* our manifold parameterization that StyleGAN can synthesize successfully.

We start from a few photos and use standard 3D reconstruction to obtain calibrated cameras and mesh of the face, as described below. To render the face from an unconstrained novel view $\mathbf{V}$, we use the steps described next.

We first project $\mathbf{V}$ to the closest camera $\hat{\mathbf{V}}$ on the manifold, thereby reducing the problem to an in-domain rendering task. We then use an implicit latent representation network, which maps physically meaningful coordinates on the camera manifold to a StyleGAN latent code (Sec. 5). The network functions as a parameterized embedding of the face and is trained in a supervised fashion using aligned input views and simple IBR renderings [Buehler et al. 2001] in a progressive training schedule. The entire pre-processing pipeline is face-specific and takes 45 minutes.

The final step is to move from the StyleGAN-generated manifold view $\hat{\mathbf{V}}$ to free view $\mathbf{V}$ (Sec. 6). Given that both views correspond to physically meaningful cameras, we use the face mesh to establish a dense flow field which describes how the image corresponding to the manifold view $\hat{\mathbf{V}}$ needs to be deformed to obtain the image corresponding to (unconstrained) novel view $\mathbf{V}$.

3.2 Preprocessing

To incorporate 3D awareness, our method relies on calibration of the input cameras and an approximate geometric representation of the face to be rendered. Neither calibration nor geometry need to be accurate for our method to produce convincing free-viewpoint images.

We calibrate cameras and create the geometric proxy using off-the-shelf software [CapturingReality 2016], resulting in a reconstructed triangle mesh, the calibrated cameras and undistorted input images (see supplemental for details). Quality is satisfactory using 10-25 photos, despite casual sequential capture without a rig.

4 THE CAMERA MANIFOLD

To allow free-viewpoint rendering of captured faces, we first need to understand the native capabilities of StyleGAN in terms of viewpoint synthesis. To this end, we define the subset of cameras that allows the generation of valid StyleGAN2 images. The portrait model published by the authors [Karras et al. 2020b] is trained on the FFHQ dataset [Karras et al. 2019]. The alignment of this dataset is a key component of the model, greatly improving quality but limiting the variety of images that can be synthesized. This is because the photos lie on a relatively narrow image manifold [Jahanian et al. 2020], implicitly defined by this alignment process based on facial features. We define our *camera manifold* based on the FFHQ alignment procedure (Sec. 4.1), estimate the range of the manifold in which StyleGAN can generate realistic images (Sec. 4.2), explain how to project a free-viewpoint camera to the closest manifold camera (Sec. 4.3), and finally present the process of aligning a captured face to the canonical coordinate system we use (Sec. 4.4).

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. 3a). 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. 3a). From these three points a square crop window is computed by determining its center $\mathbf{c}$ (also using the eye midpoint $\mathbf{x}_c$) as well as its orientation and size $\mathbf{s}$, all illustrated in Fig. 3b. Exact formulas are given in the supplemental. Given the crop window geometry, the original image is re-sampled to obtain the final aligned output image (Fig. 3c).

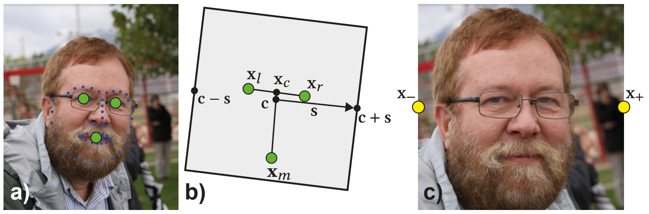

Fig. 3. The 2D alignment performed in the FFHQ dataset. *a)* Raw facial feature points (blue dots) are detected and aggregated to obtain representative eye and mouth positions (green dots). *b)* Geometric features are used to determine the square crop window (grey, not shown to scale) with center $\mathbf{c}$ and vector $\mathbf{s}$ giving orientation and size. *c)* The resulting aligned image.

4.1 Defining the Camera Manifold

The alignment procedure described above guides the definition of our *camera manifold*. We first write the alignment as a mapping from facial feature positions to a 2D similarity transform: $\mathbf{F}(\mathbf{x}_l, \mathbf{x}_r, \mathbf{x}_m) \in \mathbb{R}^6 \to (\mathbb{R}^2 \to \mathbb{R}^2)$. The similarity transform maps pixel locations from the unaligned to the aligned image. We observe that $\mathbf{F}$ is deterministic and C^0-continuous. Since $\mathbf{F}$ is an alignment

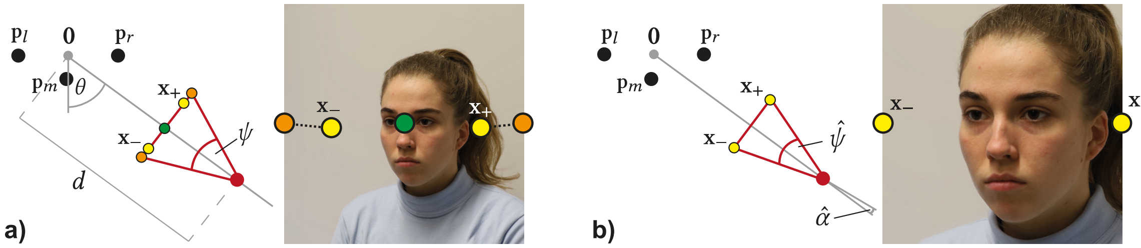

Fig. 4. The construction of the camera manifold. 2D illustrations capture the setup from the top. *a)* Our setup consists of 3D eye and mouth positions $\mathbf{p}_l$, $\mathbf{p}_r$, and $\mathbf{p}_m$, and a perspective camera (red) parameterized by spherical coordinates (only θ and d are shown in this 2D illustration). The trackball parameterization in Eq. 2 keeps the eye midpoint (green point) fixed to the image center, but fails to move $\mathbf{x}_-$ and $\mathbf{x}_+$ (yellow points) to their 2D target positions (orange points). *b)* Our manifold camera model (Eq. 3) optimizes for an additional 3D camera rotation (only $\hat{\alpha}$ is shown in the 2D illustration) and the field of view $\hat{\psi}$ to ensure that projected locations are fixed to their targets. The resulting image lies in the span of the StyleGAN portrait model.

procedure, it exhibits an infinite number of invariances. In our model, we require two invariances for an unambiguous mapping and we consider the two (arbitrary) points (Fig. 3b)

$$\mathbf{x}_- = [0, 0.5]^T = \mathbf{F}(\mathbf{x}_l, \mathbf{x}_r, \mathbf{x}_m)(\mathbf{c} - \mathbf{s}) =: \mathbf{F}_-(\mathbf{x}_l, \mathbf{x}_r, \mathbf{x}_m)$$

and

$$\mathbf{x}_+ = [1, 0.5]^T = \mathbf{F}(\mathbf{x}_l, \mathbf{x}_r, \mathbf{x}_m)(\mathbf{c} + \mathbf{s}) =: \mathbf{F}_+(\mathbf{x}_l, \mathbf{x}_r, \mathbf{x}_m),$$

where the definitions of $\mathbf{F}_-$ and $\mathbf{F}_+ \in \mathbb{R}^6 \to \mathbb{R}^2$ exploit the fact that $\mathbf{c}$ and $\mathbf{s}$ are themselves functions of the facial feature positions.

Our method relates the fixed positions $\mathbf{x}_-$ and $\mathbf{x}_+$ to feasible camera parameters, thus connecting 2D image alignment to 3D-aware image formation. To map a 3D camera pose to aligned image space, we solve the following problem: Assuming fixed eye and mouth positions in *3D space*, which combinations of intrinsic and extrinsic camera parameters yield the prescribed projection locations $\mathbf{x}_-$ and $\mathbf{x}_+$ in image space?

For the purpose of defining our camera manifold, we assume that StyleGAN-generated images are modeled as being captured with a perspective pinhole camera. Note however, that we can generate images using arbitrary camera models. We further assume, without loss of generality, that the left and right 3D eye positions are

$$\mathbf{p}_l = [-1, 0, 0]^T \quad \text{and} \quad \mathbf{p}_r = [1, 0, 0]^T, \tag{1}$$

respectively. We further assume we have access to the 3D mouth position $\mathbf{p}_m$ (see supplemental for our exact definition of a frontal pose). In Sec. 4.4 we show how to align a general pose of a face capture to this canonical setting. Note that this alignment does not impose any restrictions on face shape.

A full camera model with 7 DoF gives too much freedom for our manifold, as many camera parameters correspond to an image with the face lying outside the frame. We therefore, as a first step, restrict ourselves to a trackball camera model:

$$\bar{\mathbf{A}}(\theta, \phi, d, \psi) = \mathbf{P}(\psi)\mathbf{T}(0, 0, d)\mathbf{R}(\theta, \phi, 0), \tag{2}$$

where $\mathbf{R} \in \mathbb{R}^3 \to \mathbb{R}^3$ is a rotation parameterized by three Euler angles, $\mathbf{T} \in \mathbb{R}^3 \to \mathbb{R}^3$ is a translation parameterized by three offset coordinates, and $\mathbf{P} \in \mathbb{R}^3 \to \mathbb{R}^2$ is a perspective projection with field of view ψ. The free parameters θ, ϕ, d, and ψ correspond to horizontal and vertical rotations around the eye midpoint, distance to the eye midpoint, and field of view, respectively (Fig. 4a). $\bar{\mathbf{A}}$ corresponds to an un-aligned image in the spirit of Fig. 3a: The face lies within the frame, but is not aligned in general.

In a second step, we enable fixed projection locations as follows:

$$\hat{\mathbf{A}}(\theta, \phi, d) = \mathbf{P}(\hat{\psi})\mathbf{R}(\hat{\alpha}, \hat{\beta}, \hat{\gamma})\mathbf{T}(0, 0, d)\mathbf{R}(\theta, \phi, 0). \tag{3}$$

Here, we add an additional rotation before the projection and exclude ψ from the list of free parameters. We refer to the free parameters $\mathbf{m} = [\theta, \phi, d]^T \in \mathbb{M} \subset \mathbb{R}^3$ as *manifold coordinates*. We define the space of $\mathbb{M}$ in Sec. 4.2. In terms of the camera position, $\mathbf{m}$ corresponds to spherical coordinates. Intuitively, the coefficients $\hat{\mathbf{c}} = \left[\hat{\alpha}, \hat{\beta}, \hat{\gamma}, \hat{\psi}\right]$ steer a rotation (via $\hat{\alpha}$, $\hat{\beta}$, $\hat{\gamma}$) and scaling (via $\hat{\psi}$) without changing the camera position. Excluding degenerate cases, which we avoid as described in Sec. 4.2, for any fixed set of manifold coordinates, there exists a coefficient vector $\hat{\mathbf{c}}(\mathbf{m})$ to position two 3D points to arbitrary 2D projected image locations.

We now have all the machinery in place to solve our alignment problem (Fig. 4b), i.e., given a manifold coordinate $\mathbf{m}$, find a coefficient vector $\hat{\mathbf{c}}(\mathbf{m})$ that simultaneously satisfies the four equations:

$$\begin{cases} \mathbf{F}_-\left(\hat{\mathbf{A}}(\mathbf{m})\mathbf{p}_l, \hat{\mathbf{A}}(\mathbf{m})\mathbf{p}_r, \hat{\mathbf{A}}(\mathbf{m})\mathbf{p}_m\right) = \mathbf{x}_- \\ \mathbf{F}_+\left(\hat{\mathbf{A}}(\mathbf{m})\mathbf{p}_l, \hat{\mathbf{A}}(\mathbf{m})\mathbf{p}_r, \hat{\mathbf{A}}(\mathbf{m})\mathbf{p}_m\right) = \mathbf{x}_+ \end{cases} \tag{4}$$

where $\hat{\mathbf{A}}(\mathbf{m})\mathbf{p}$ denotes the projection of 3D point $\mathbf{p}$ to the screen using camera $\hat{\mathbf{A}}(\mathbf{m})$. Due to the nonlinear nature of the equations, obtaining an analytic solution is challenging. Instead, we solve numerically for the least-squares solution using the Levenberg-Marquardt algorithm, which usually converges after 35 iterations. The optimization takes about 10 milliseconds in our Python implementation and produces temporally stable results. We therefore solve the equations on the fly.

Our model allows free positioning of the camera via $\mathbf{m}$. However, all rotational DoF as well as the field of view are automatically adjusted to ensure aligned portrait images. Our cameras therefore lie on a 3D manifold in the 7D parameter space. The coordinates θ and ϕ resemble yaw/pitch parameterizations of face pose in previous work, but by construction also include translational and scaling components which previous approaches do not model. Manipulation of d results in the Vertigo effect: Increasing d, i.e., moving the camera

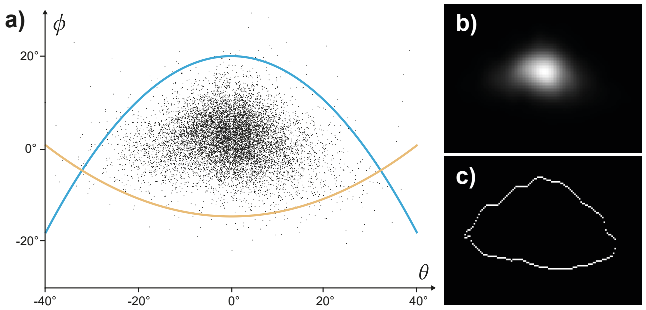

Fig. 5. Camera manifold range analysis. Sampling and analyzing StyleGAN-generated images in terms of camera pose (dots in *a*) reveals an anisotropic distribution, which we bound using two parabolas (curves in *a*). Bounding proceeds by first converting the samples into a probability distribution (*b*), followed by iso-line extraction (*c*) and curve fitting.

away from the face, is compensated by a corresponding decrease in the field of view, and vice versa.

4.2 Range

The above definition does not restrict camera position $\mathbf{m}$, allowing arbitrary views of the head, e.g., profile or rear views. We observe, however, that StyleGAN-generated images have smaller variations in camera pose, consistently showing front or moderate oblique angle views. We next determine the boundaries or *range* of our camera manifold. This mainly concerns the rotation parameters θ and ϕ; the Vertigo-type variations induced by d are small enough to ignore in this context, when avoiding very small d (and resulting large $\hat{\psi}$). Therefore, we simply enforce $d \geq 10$, i.e., camera positions at least five inter-ocular distances away from the face.

To quantify the range of camera positions in the $\theta\phi$-plane, we first produce 10k random facial images using the StyleGAN2 generator with default truncation parameter $\psi = 0.5$. We then use the method of Bulat and Tzimiropoulos [2017] to infer estimates of 3D eye and mouth positions per image, which we convert to corresponding $\theta\phi$-tuples (details on calibration in supplemental). When plotting the resulting distribution (dots in Fig. 5a), we see that the images indeed concentrate around the frontal pose $\theta = \phi = 0$, but show a strong eye-shaped anisotropy, while being close to symmetric around the axis $\theta = 0$. These observations suggest that we can model the boundary of this distribution using two parabolas of the form $\phi = a\theta^2 + b$. To obtain the coefficients a and b, we first convert the samples into a probability distribution $p_{\theta\phi}$ (Fig. 5b). We use kernel density estimation on a regular grid of size 128×96 capturing $\theta \in [-40°, 40°]$ and $\phi \in [-30°, 30°]$, utilizing a Gaussian kernel with a standard deviation of three grid cells. We then determine the iso-lines $p_{\theta\phi} = 0.01 \max(p_{\theta\phi})$ (Fig. 5c) and perform two least-squares parabola fits. Below, we give the result of our fit for the upper and lower boundary curves (parabolas in Fig. 5a):

$$c_u(\theta) = -0.024\theta^2 + 20.00 \quad \text{and} \quad c_l(\theta) = 0.010\theta^2 - 14.60.$$

A valid coordinate in our model satisfies $c_l(\theta) \leq \phi \leq c_u(\theta)$.

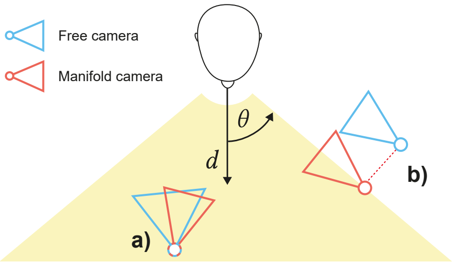

Fig. 6. Projecting cameras to the camera manifold (only θ and d are shown in this top view): *a)* If the position of the free-viewpoint camera (blue) is in the valid range (orange area, corresponds to a ϕ-slice of Fig. 5a), the closest manifold camera (pink) only differs by in-place rotation and field of view. *b)* If the free-viewpoint camera lies outside the valid range, the camera position is also affected by the manifold projection. Consequently, parallax occurs only in case *b)*.

4.3 Projecting a Camera to the Manifold

Finding the closest camera on the manifold for a free-viewpoint camera is easy in our model (Fig. 6): We take the camera position in spherical coordinates as manifold coordinate $\mathbf{m}$ and solve Eq. 4.

However, arbitrary camera positions do not necessarily correspond to valid manifold coordinates as defined in the previous section. In these cases, we find the closest valid manifold coordinate (Fig. 6b), using an analytic solution for this projection to the valid range (details in supplemental). Note that only in this case the camera position changes and parallax is induced in the image.

4.4 Alignment to Canonical Coordinates

Our head reconstruction can have arbitrary global 3D scene scale, positioning and orientation. Since our camera manifold assumes the eyes to be at defined 3D positions $\mathbf{p}_l$ and $\mathbf{p}_r$, we perform 3D alignment to the canonical configuration (Eq. 1) [Gao et al. 2020; Rao et al. 2020], making our method independent of the reconstruction algorithm.

The locations of the eyes and mouth on the mesh are required for alignment. While 3D facial feature location algorithms exist [Bowyer et al. 2006], our multi-view data provided stable results despite possibly strong reconstruction noise: Similar to the FFHQ alignment, we use the method of Kezemi and Sullivan [2014] to obtain facial feature points on the face for each input view and compute representative 2D eye and mouth locations (Fig. 3a). We then re-project the 2D feature points onto the mesh and average the re-projected features of all input views to obtain a robust estimate of the eye and the mouth positions in 3D.

We now seek to find a similarity transform that maps the estimated 3D eye positions to their targets $\mathbf{p}_l$ and $\mathbf{p}_r$. To obtain a unique solution and remain consistent with the definition of the camera manifold, we further enforce a frontal pose in terms of vertical orientation, as detailed in the supplemental. The transformation matrix is obtained using the Levenberg-Marquardt algorithm and then used to transform both the mesh and the input camera parameters.

We also process the input images: We observe that the embedding of images into the GAN latent space (Sec. 5) gives results of higher visual quality when the background is smooth. Further, such "simple" backgrounds are easier to integrate into a synthetic scene. We therefore blur the background of the input images. We use the method of Ke et al. [2020] to extract a foreground matte. Then we apply a strong Gaussian filter (we use a σ of one tenth the image width) to the background region [Knutsson and Westin 1993] and compose the foreground on top. Finally, we 2D-align all input images using the FFHQ alignment.

Recall, that in the canonical coordinate system, any camera's manifold coordinate $\mathbf{m}$ corresponds to the camera's position in spherical coordinates. Consequently, the manifold coordinates of the 3D-aligned input cameras – provided they are in the valid range – correspond to the 2D-aligned images in the sense of our model.

5 MAPPING MANIFOLD COORDINATES TO STYLEGAN LATENT CODES

We now have the manifold coordinates that correspond to a given camera pose. We next find the StyleGAN latent code that corresponds to a given manifold coordinate, by training a small per-face implicit latent representation network. This network allows us to move on the camera manifold.

5.1 StyleGAN Terminology

StyleGAN [Karras et al. 2019] maps normally distributed random samples $\mathbf{z} \in \mathbb{R}^{512}$ to an intermediate latent vector $\mathbf{w} \in \mathbb{R}^{512}$ using a learned mapping $\mathbf{w} = H(\mathbf{z})$. The space of $\mathbf{w}$'s is commonly referred to as W. The vector $\mathbf{w}$ controls feature normalizations in 18 layers of the generator network G, which produces the final image $I = G(\mathbf{w}) = G(H(\mathbf{z}))$. It has been observed [Karras et al. 2019][Abdal et al. 2019] that the expressivity of the generator is much higher, when different $\mathbf{w}$ are fed to the generator layers. In the general case, 18 different sets of latents $\mathbf{w}^+ \in \mathbb{R}^{18\times 512} = W^+$ can be used.

5.2 Method

By construction, all views $\hat{\mathbf{V}}$ on the camera manifold can be rendered by finding a corresponding latent vector $\mathbf{w_m} \in W^+$. We therefore seek a mapping $M \in \mathbb{M} \to W^+$ from manifold coordinates to latents, such that, given manifold coordinates $\mathbf{m}$, we obtain $\hat{\mathbf{V}}$:

$$\hat{\mathbf{V}} = G\left(\mathbf{w_m}\right) = G\left(M\left(\mathbf{m}\right)\right).$$

Recall that, in contrast to all previous methods, $\mathbf{m}$ is an exact meaningful physical 3D quantity.

We found a face-specific mapping M to provide highest-quality results. Therefore, in analogy to recent works on implict neural representations [Genova et al. 2019; Sitzmann et al. 2020, 2019], we refer to M as an implicit latent representation network. It has been observed that the latents fed to the earlier StyleGAN layers correspond to coarse-scale image properties including face pose, while later layers add medium- to small-scale features to the images [Karras et al. 2019]. We therefore define a constrained architecture: For layers 0-5 we parameterize M with a small multi-layer perceptron (MLP) per layer, each mapping raw manifold coordinates $\mathbf{m}$ to a 512-D latent vector. Due to the low-frequency behaviour of pose changes, we do not use Fourier features [Tancik et al. 2020]. For the MLPs, we found two hidden layers with 32 features and leaky ReLU activation functions to be sufficient for our application. For the remaining layers 6-17 we directly optimize for static latents, which ensures a consistent output for different $\mathbf{m}$ (Fig. 7).

5.3 Training Data

Thanks to our multi-view setup and the geometric reconstruction, we can train M in a supervised fashion using training tuples $\left\{\left(\mathbf{m}_k, \hat{\mathbf{V}}_k\right)\right\}_k$. We employ two complementary sources of information as training data: First, we use the aligned input images from Sec. 4.4. The images constitute a sparse set of high-quality training samples on the camera manifold. Second, we use the input views together with the face mesh to create manifold views using image-based rendering. Specifically, we use unstructured lumigraph rendering [Buehler et al. 2001] with cameras corresponding to random $\mathbf{m}$ on the camera manifold. We can generate any of these renderings on the fly covering the entire manifold. In the supplemental we give details on how we sample the valid manifold range. The geometric reconstruction and calibration have uncertainty, producing rendering artifacts that reduce overall image quality. We can compute a per-pixel estimate of this uncertainty by computing the color variance $\boldsymbol{\sigma}_c^2$ when blending the different images; we will use this estimate to reduce the influence of incorrect IBR pixels.

5.4 Loss

Our renderings should: *a*) be close to the training images, *b*) preserve the identity of the depicted person from all views, and *c*) look photorealistic. We use the following loss to achieve these goals:

$$\mathcal{L} = \mathcal{L}_{\ell_1} + \lambda_{\text{LPIPS}}\mathcal{L}_{\text{LPIPS}} + \lambda_{\text{id}}\mathcal{L}_{\text{id}} + \lambda_{\text{prior}}\mathcal{L}_{\text{prior}}. \tag{5}$$

The first two terms address goal *a*). Here, $\mathcal{L}_{\ell_1}$ penalizes pixel differences in terms of the ℓ_1-norm, weighted by per-pixel IBR confidence:

$$\mathcal{L}_{\ell_1} = \left\|\exp\left(-\eta\boldsymbol{\sigma}_c^2\right)\left(\hat{\mathbf{V}} - \hat{\mathbf{V}}_k\right)\right\|_1, \tag{6}$$

with the falloff factor $\eta = 100$ in all our experiments and $\boldsymbol{\sigma}_c^2 = \mathbf{0}$ if $\hat{\mathbf{V}}_k$ is an input view. $\mathcal{L}_{\text{LPIPS}}$ is the LPIPS [Zhang et al. 2018] distance which we apply to a downsampled version of the images to a resolution of 256 × 256 pixels.

The identity loss $\mathcal{L}_{\text{id}}$ addresses goal *b*) and makes use of the pretrained VGG-face [Parkhi et al. 2015] network Ψ, which converts an image into face recognition features:

$$\mathcal{L}_{\text{id}} = 1 - \langle\Psi(\hat{\mathbf{V}}), \overline{\Psi}\rangle,$$

where $\overline{\Psi}$ denotes the normalized mean face recognition features of all input views.

Finally, we address goal *c*) by recognizing that StyleGAN renderings of highest quality and stability are usually obtained when the latents $\mathbf{w}$ follow the distribution dictated by the mapping network H. Specifically, we address the problem that the extended space W^+, where the styles of different StyleGAN layers are decoupled from each other, leads to an underconstrained problem and consequently images of lower realism. We enforce a certain amount of coherence between the styles using

$$\mathcal{L}_{\text{prior}} = \frac{1}{18}\sum_{i=1}^{18}\left\|\mathbf{w}_i - \overline{\mathbf{w}}\right\|_1,$$

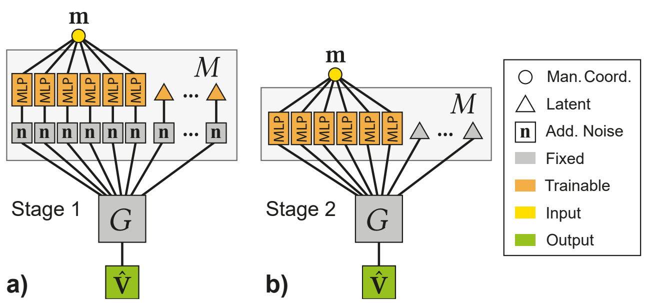

Fig. 7. We employ a progressive training schedule. *a)* In the first stage, we only use the input images as training data. We train MLPs that map manifold coordinates (yellow dot) to the first 6 StyleGAN latents, and we directly optimize for the remaining static latents (orange triangles). All latents fed to G are subject to random perturbations (boxes labelled $\mathbf{n}$) 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).

where $\mathbf{w}_i$ refers to the latent fed into the i-th generator layer, and $\overline{\mathbf{w}} = \frac{1}{18}\sum_{i=1}^{18}\mathbf{w}_i$. This prior encourages the network to produce latents which are similar for all generator layers, i.e., closer to the original StyleGAN latent space W.

5.5 Training

We found that a progressive training schedule, which splits training into two stages, produces results of highest quality. Fig. 7 summarizes our method; details and exact parameters are given in supplemental.

In the first stage, we only use the aligned input views as training data and optimize all trainable parameters (Fig. 7a). 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. This stage trains in 35 min on an NVIDIA RTX6000.

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%). We fix the latents of the static layers to prevent high-frequency IBR artifacts from impacting them (Fig. 7b). This stage fills in the pose gaps between the input views and trains in 4 min.

6 FREE-VIEWPOINT RENDERING

We have so far established a method to render a view $\hat{\mathbf{V}}$ corresponding to a pinhole camera $\hat{\mathbf{A}}$ on the camera manifold using StyleGAN. We now seek to move away from the manifold to synthesize free-viewpoint images $\mathbf{V}$ with an arbitrary camera model $\mathbf{A}$. We achieve this goal using a simple image warping strategy as follows.

First we project $\mathbf{A}$ to the closest manifold camera $\hat{\mathbf{A}}$ (Sec. 4.3, first column in Fig. 8). Recall that both $\mathbf{A}$ and $\hat{\mathbf{A}}$ correspond to physically meaningful cameras. Therefore, since we have the face geometry at our disposal, we can render an inverse flow field [Mark et al. 1997; Yang et al. 2011], which for each pixel of $\mathbf{V}$ indicates where to lookup $\hat{\mathbf{V}}$ (second column in Fig. 8). The desired viewpoint $\mathbf{V}$ is then obtained by warping $\hat{\mathbf{V}}$ according to the flow field (third column in Fig. 8). This procedure is compatible with any camera model $\mathbf{A}$ for $\mathbf{V}$, including physical lenses, stereoscopic setups, etc. Multi-sampled effects require multiple entries per flow field pixel [Yu et al. 2010]. We use multi-sampling by default to anti-alias occlusion boundaries arising from parallax.

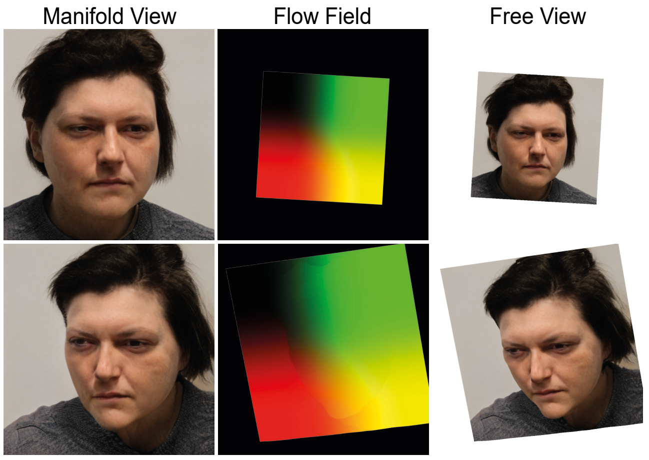

Fig. 8. Free-view generation in our approach: We use StyleGAN to generate the closest view on the camera manifold (left), which is warped using a flow field (center) to obtain the final result (right). The top row shows an example of a parallax-free flow field, arising from a free camera in the valid manifold range. The configuration in the bottom row requires parallax in the flow field, as the free camera lies outside the valid manifold range.

Note that if the position of $\mathbf{A}$ corresponds to a manifold coordinate $\mathbf{m}$ in the valid range (Sec. 4.2), the projection to $\hat{\mathbf{A}}$ does not change the camera's position (Fig. 6a). Therefore, the flow field is parallax-free and corresponds to a continuous remapping (top row in Fig. 8). If $\mathbf{A}$ is a perspective pinhole camera, the warp even reduces to a simple global affine image transformation. Parallax occurs in the flow field only if $\mathbf{A}$ moves outside the valid manfold range (Fig. 6b; bottom row in Fig. 8). This operation is akin to view-dependent texture mapping [Debevec et al. 1996], with StyleGAN as a texture generator.

The final image $\mathbf{V}$ is always synthesized using a coordinated interplay between latent manipulations and image warping. Even a seemingly simple lateral camera motion cannot be created by only shifting the image: The new viewpoint results in a slight variation of viewing angle that in turn leads to a change of manifold coordinates. A re-evaluation of the networks for synthesizing correct perspective is therefore required.

7 RESULTS AND EVALUATION

We implemented our method based on the original StyleGAN2 code [Karras et al. 2020b] in Python, complemented by a custom OpenGL rendering framework, including an interactive viewer. We provide all source code of our method here: https://repo-sam.inria.fr/fungraph/freestylegan.

To acquire faces we requested authorization from our institutional ethics committee, and due to restrictions imposed by their decision concerning privacy and security, we are only able to show a small number of images per subject as illustration in this paper, but cannot distribute the full datasets. We provide instructions on how to capture a face with a smartphone in the supplemental.

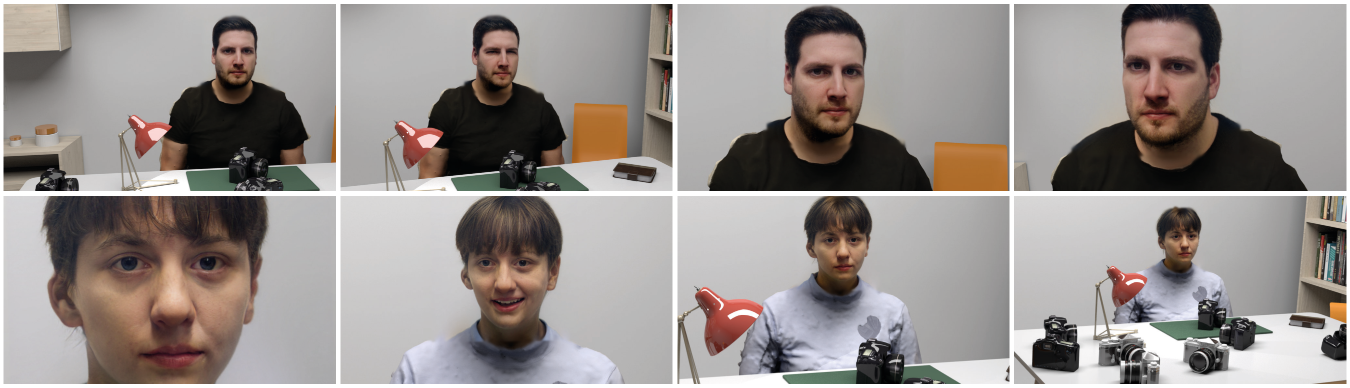

Fig. 9. Compositing of our portrait renderings into a synthetic scene with a moving camera. Note the semantic edits (eyes, smile) in the second column.

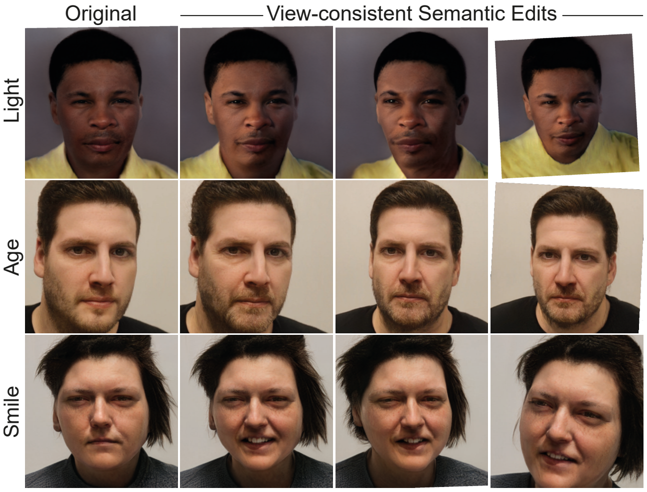

Fig. 10. Semantic editing results: The original multi-view embedding (col. 1) is modified (col. 2) and stays consistent across novel views (col. 3+4).

7.1 Results

To evaluate our approach, we use our own captures. We asked participants for a series of 10-25 photographs of their head and upper body. The participants were asked to sit still, while non-professional photographers with no background in visual computing sequentially captured images from different viewing angles, using a digital (smartphone) camera. None of the authors (or their collaborators) were present during capture and no training was conducted.

To ensure both successful geometry reconstruction and disentanglement of the camera pose in the StyleGAN latents, we found it crucial to eliminate all other sources of variation (lighting, facial expression, etc.) across the input views. Yet, our technique is robust enough to allow lightweight sequential capture and does not require a multi-camera setup.

To demonstrate free-viewpoint rendering and the accuracy of our camera model, we composite our results on top of a rendering of a synthetic scene in Fig. 9 with a moving camera. We perform compositing with color correction membranes [Farbman et al. 2009] using convolution pyramids [Farbman et al. 2011]. As can also be seen from our supplemental video, our approach for the first time allows to successfully integrate a 2D GAN portrait rendering into a synthetic 3D scene with free-viewpoint control.

In Fig. 10 we show semantic editing results. We apply the method of Härkönen et al. [2020] to the output latents from our implicit representation network. We observe that the modifications stay view-consistent in our renderings (see supplemental video for interactive editing sessions). Note that the method used to perform semantic editing is independent of our approach.

In Fig. 11 we demonstrate face renderings with different camera models. The Vertigo effect (Fig. 11b) can be achieved without leaving the camera manifold, i.e., without warping, by simply varying the d-component of the manifold coordinate. We see that our latent representation network is able to synthesize the shift in perspective when moving towards the face while opening the field of view. In Fig. 11c we demonstrate spherical lens distortions, while Fig. 11d showcases a stereoscopic rendering result. Our solution generates the two images from the correct viewpoints, also allowing fine-grained control over interocular distance and screen depth.

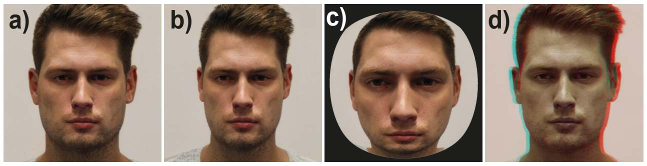

Fig. 11. Different camera models: Perspective pinhole camera (*a*), Vertigo-effect (*b*), non-linear lenses (*c*), and stereo image pairs (*d*, use anaglyph glasses for stereo 3D impression).

7.2 Comparisons

We compare our algorithm to state-of-the-art StyleGAN-based rendering techniques, portrait-specific approaches, and for completeness to general-purpose free-view IBR methods. A full set of videos can be found in our supplemental material.

First we compare our approach to two state-of-the-art StyleGAN-based portrait rendering approaches: PIE [Tewari et al. 2020b] and StyleFlow [Abdal et al. 2021]. While both methods perform novel-view portrait rendering using StyleGAN, they differ from ours in two fundamental ways: First, they rely solely on manipulations of

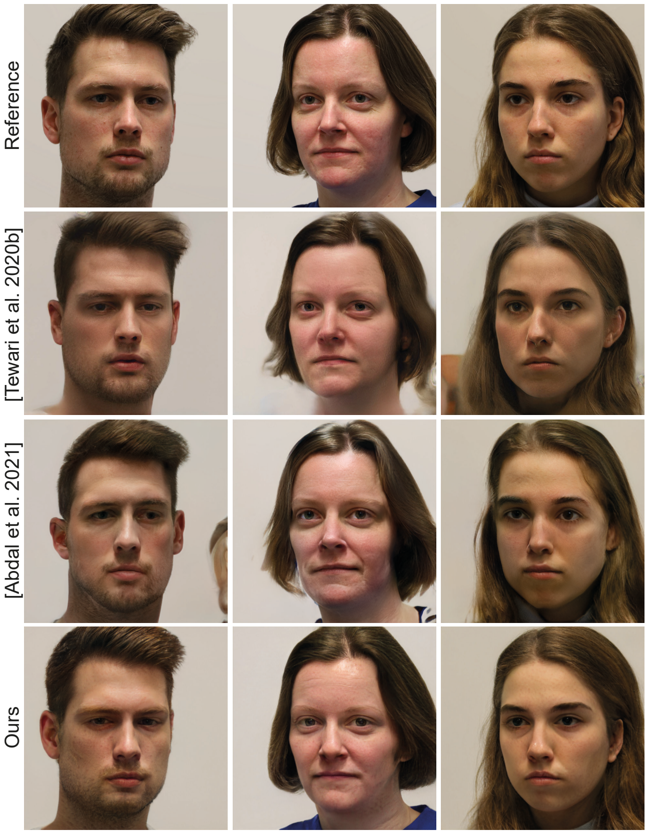

Fig. 12. Comparison to state-of-the-art StyleGAN editing approaches.

StyleGAN latent code $\mathbf{w}^+$, restricting novel views to the camera manifold, i.e., a very small fraction of what our method allows, prohibiting free-viewpoint navigation. Second, their view parameterization does not correspond to a physically meaningful quantity: Latent code-induced viewpoint manipulations are not simply a camera rotation around a fixed 3D point (Sec. 4), but induce non-linear dependencies between camera position, orientation and field of view. This is in contrast to our camera manifold formulation which translates latent codes into physically meaningful quantities. To allow a fair comparison, we thus restrict our output to the camera manifold, omitting our free-viewpoint warping. We determine image quality by re-synthesizing held-out input views. For both PIE and StyleFlow, we use a frontal view as input. Since there is no way to directly find the held-out camera with these methods, we densely sample the yaw-pitch-space for both methods and select the view minimizing mean ℓ_2-distance of facial landmarks [Kazemi and Sullivan 2014] to the requested input view. Our approach generates the required views directly. Our method outperforms the others both numerically (Tbl. 1) and, arguably, in subjective image quality (Fig. 12).

We also compare against the facial re-enactment technique of Siarohin et al. [2019] and the morphable face albedo model of Smith et al. [2020] in Fig. 13. For Siarohin et al. we use their pre-trained model based on the VoxCeleb dataset [Nagrani et al. 2017], which allows a reasonable variety of poses due to a conservative crop window. We use a textured mesh [CapturingReality 2016] rendered from free viewpoints (unless stated otherwise) to drive an aligned frontal-pose view. We observe that their result quality highly depends on the viewpoint: While frame-filling portrait views are handled well, viewpoint-induced scaling of the head tends to result in distortions or severe identity shifts. Additionally, their spatial output resolution is 256 × 256, in contrast to our resolution of 1024 × 1024. For the method of Smith et al. we use their inverse rendering pipeline based on a frontal view. Their method is designed for maximum control over face shape, facial expression, and surface properties far beyond what the StyleGAN latent space captures, but it suffers from a lack of photo-realism.

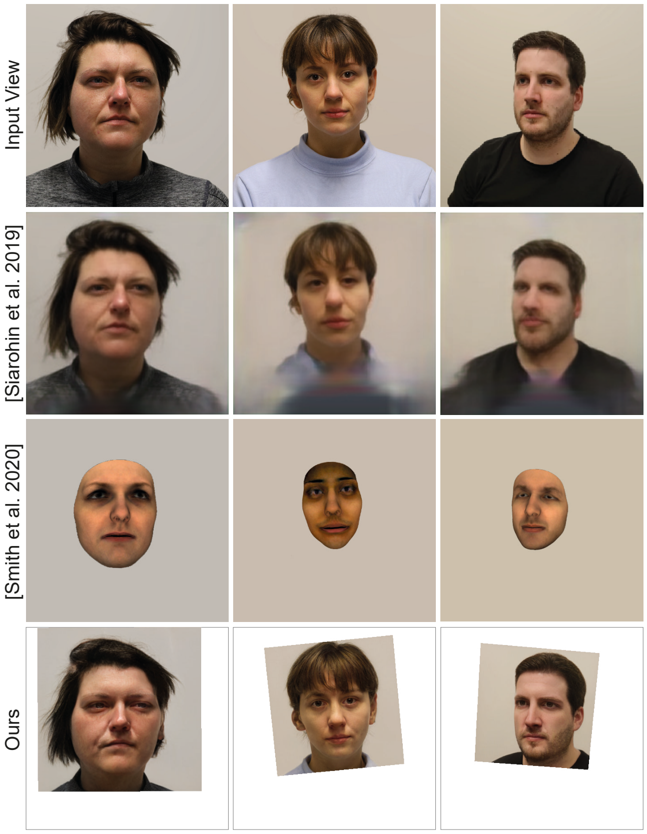

Fig. 13. Comparison to editable portrait rendering methods.

Table 1. Image quality comparison of our method against state-of-the-art StyleGAN-based approaches.

Method	PSNR↑	SSIM↑	E-LPIPS↓
[Tewari et al. 2020b]	17.8	.711	.027
[Abdal et al. 2021]	17.2	.717	.028
Ours	20.8	.758	.020

Our technique allows to change the distance between face and camera by manipulating StyleGAN latent codes using our camera manifold formulation and the trained network M. In Fig. 14, we compare this latent-driven shift of perspective to the single-image approach of Fried et al. [2016]. Given a shot captured from a close distance (Fig. 14, left), their method uses warping to simulate a long-distance shot with the corresponding decrease in field of view

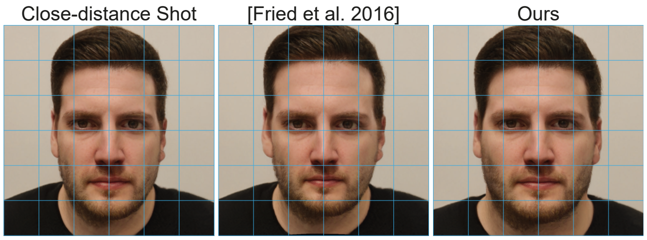

Fig. 14. Manipulation of camera distance and focal length (Vertigo effect) using a warping technique (center) and our latent-driven approach (right). Use the grid overlay to assess changes of proportions.

(Fig. 14, center). In contrast, our method (Fig. 14, right) requires only a manipulation of manifold coordinates and no warping to synthesize this Vertigo effect. We observe that, while both techniques exhibit similar characteristics in the central face region, our method produces globally more consistent results, e.g., neck and shoulders are also correctly influenced by the shift of perspective.

We quantitatively evaluate camera accuracy, image quality, and identity preservation in Tbl. 2. For completeness, we include the multi-view IBR methods DeepBlending [Hedman et al. 2018], Free View Synthesis [Riegler and Koltun 2020], and NeRF++ [Zhang et al. 2020]. For the first two approaches, we use the pre-trained models from the authors, which have not been trained on faces. Note that, in contrast to ours, none of these methods allows semantic editing.

To estimate camera accuracy, we re-synthesize all input views for six subjects resulting in 132 images total, while holding out the input views to compare against for the approaches of Hedman et al. [2018], Riegler and Koltun [2020], and ours. For the method of Siarohin et al. [2019] we use the input views as the driving source. We then determine the ℓ_2-distance of 2D facial landmarks [Kazemi and Sullivan 2014] to those of the ground truth images. To compensate for different image resolutions, we normalize this error by the estimated interocular distance per view. We additionally report the success rate of the landmark detector, providing an indication of how realistic the generated faces are [Tewari et al. 2020b]. To analyze the capability of identity preservation, we employ the method of Schroff et al. [2015] to extract face recognition features for 280 free-viewpoint video frames across different subjects and measure the cosine distance to the normalized mean recognition features of the respective input views (last column in Tbl. 2). In the supplemental we also present visual comparisons and image error metrics. Since most free-viewpoint IBR methods reproject images they achieve better quality; recall however that they *do not allow any semantic editing*. In contrast, 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.

7.3 Ablations

We analyze the effectiveness of individual components of our algorithm by ablation. We consider distribution and number of input views, alternatives to our camera manifold formulation, loss terms and training procedure, and our background blur approach.

7.3.1 Input Views. In Fig. 15a we show a typical distribution of input views in our manifold parameterization. The distributions of multiple datasets can be found in our supplemental video. We observe that our casual capture tends to result in an uneven view distribution and a significant fraction of the input views lying outside the manifold boundaries. These views are vital for successful geometry reconstruction (Sec. 3.2), as they cover the subject from a wider range of directions, but these views cannot be used in the first training stage (Sec. 5.5). We did not observe a strong effect of the exact view distribution when input views are reasonably stratified.

Table 2. Comparison of camera accuracy (measured by facial landmarks: Alignment & Detection Rate), and face Recognition Error.

Method	Semantic	Facial Landmarks		Recog.
	Editing	Align.↓	Det. Rate↑	Error↓
[Hedman et al. 2018]	✗	.023	99%	.07
[Riegler and Koltun 2020]	✗	.027	100%	.08
[Zhang et al. 2020][1]	✗	.018	100%	.24
[Siarohin et al. 2019]	✓[2]	.254	42%	.23
Ours	✓	.068	100%	.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.

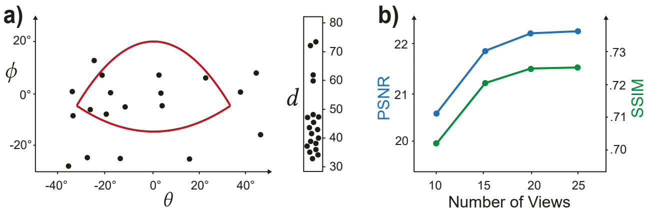

Fig. 15. *a)* A typical input view distribution (black dots) in our manifold parameterization. A large fraction of views lies outside the manifold boundary (red shape). *b)* Image quality as a function of input view count.

To investigate how the number of input views influences result quality, we progressively reduce the views used in our pipeline. We then perform an exhaustive leave-one-out image quality analysis (following the same protocol as in Sec. 7.2) for each configuration. The results are given in Fig. 15b. We observe that, unsurprisingly, image quality improves as the number of input views increases, but tends to saturate at about 20 views. Using less than 15 views has a stronger negative effect on image quality. For less than 10 views camera calibration and geometry reconstruction are unreliable.

7.3.2 Camera Manifold. We analyze three alternatives to our camera manifold formulation. In the upper row of Fig. 16, we show a result obtained when omitting the manifold completely, i.e., using the trackball camera model from Eq. 2. We feed the four camera parameters to M and train it to produce latents for free-viewpoint images directly. As this task is harder, we give M more capacity by doubling the number of both the hidden layers and feature channels. The resulting images are severely distorted, while our approach matches the reference - a ULR rendering of the desired pose - well.

The lower row of Fig. 16 shows a result obtained with a naive manifold: We use the frontal pose to optimize for a coefficent vector $\hat{\mathbf{c}}$ and fix it while rotating the camera around the head. This naturally results in in-plane rotations and therefore out-of-distribution images

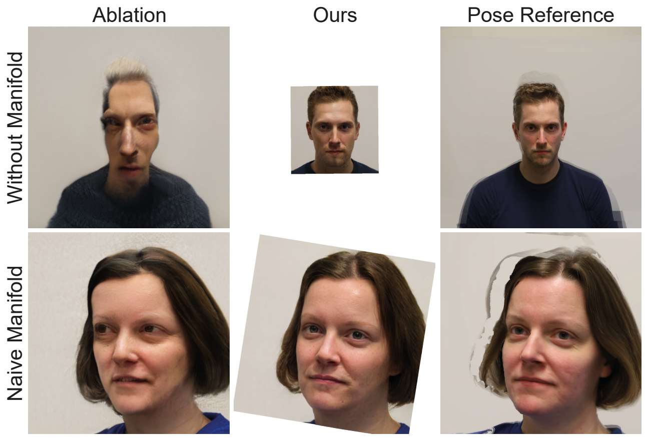

Fig. 16. Manifold ablations: Training a mapping from free cameras to latents gives heavily distorted results (upper left). A naive mainfold is better, but still does not respect training data alignment and therefore results in distortions (lower left). In contrast, our solution (middle column) is distortion-free and matches the pose reference (right column) well.

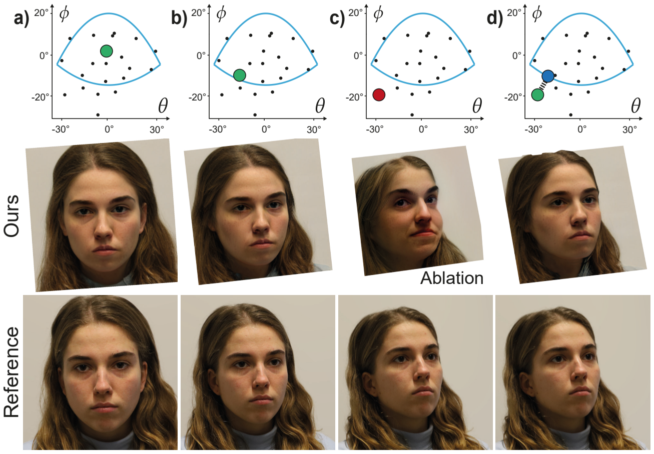

Fig. 17. Manifold boundary analysis: *a*) and *b*) Within the manifold boundaries our results are of consistently high quality. *c*) Directly generating a view outside the manifold (red point) using StyleGAN results in low quality compared to the held-out input view. *d*) We first render the closest view in the valid range (blue point) with StyleGAN. The final image (green point) is then produced using warping. Black dots denote input views.

for StyleGAN and we again observe strong distortions in the results, while our full approach handles these configurations well.

Finally, we analyze the effect of our manifold boundaries in Fig. 17. Result quality is consistently high when moving within the boundaries (Fig. 17a and b; see also supplemental videos). When we omit the manifold boundaries and use M to directly generate a view outside the valid range (Fig. 17c), image quality suffers. Our full approach (Fig. 17d) first generates the closest valid StyleGAN image, and then warps it to the desired view, resulting in superior quality.

7.3.3 Loss and Training. We analyze our two-stage training procedure in Fig. 18. When only using the first stage for training, our method relies on the sparse input views only. It therefore fails to generalize to the entire camera manifold, either by generating distorted images (first row) or false perspectives (second row). If we only use the second training stage, we get blurry results with identity shifts due to IBR artifacts in the training data and the omitted LPIPS loss.

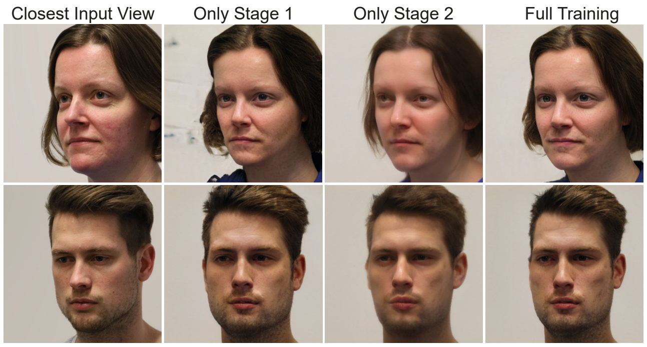

Fig. 18. Effect of the training stages: Only running stage 1 results in poor generalization to the entire manifold. Stage 2 alone gives overall lower quality results, as the high-quality information from the input views is missing. Our two-stage training procedure provides highest-quality results.

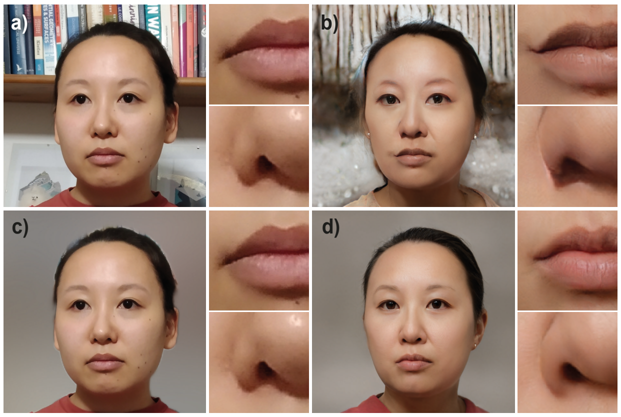

Fig. 19. If the background in the input images is not smooth (*a*), the embedding does not only fail to match the background (*b*), but also distorts facial features, such as mouth and nostrils (insets). We blur the background before the embedding (*c*) to achieve higher-quality results (*d*).

In the supplemental we also show that excluding the LPIPS term reduces image sharpness, while the identity loss preserves slight face identity shifts. The prior loss increases photo-realism.

7.3.4 Background Blur. Fig. 19 demonstrates the effect of blurring the background of the input images. We observe that a blurred background has an influence not only on the background region, but also helps to preserve the identity and increase photorealism.

8 DISCUSSION, FUTURE WORK, AND CONCLUSION

Our approach enables high-quality free-viewpoint synthesis of faces using StyleGAN with casually captured multi-view data as input. While our capture is lightweight, previous work on GAN embeddings require only a single image; at the expense of significantly less variation in pose and no direct way to connect to precise camera

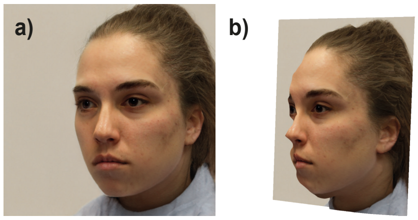

Fig. 20. Limitations: *a*) If the camera position is in the valid manifold range, no parallax occurs and rendering quality is not impacted by geometric fidelity. *b*) When leaving the valid manifold range, inaccuracies in the proxy geometry can become visible at extreme angles.

parameters. A possible extension of our method towards single-view inputs would require performing GAN inversion, geometry estimation, and camera calibration – likely using a full camera model – from a single image. This could be done by harnessing the capabilities of StyleGAN as a multi-view generator [Zhang et al. 2021a] and to fold geometry estimation into the training loop [Pan et al. 2021].

We allow free positioning of the virtual camera, beyond the distribution of the StyleGAN training corpus. If the camera position lies within the valid range of our camera manifold, the image quality of our method does not depend on the quality of the geometric reconstruction, as no parallax occurs (Fig. 20a); if it leaves the valid range, our warping scheme generates the required parallax. This can create artifacts if the proxy is incorrect or too coarse (Fig. 20b). Further, all semantic manipulations that change geometric features of the face are naturally not mirrored in the geometry, leading to projective texturing artifacts. A possible avenue of future work would be to synchronize geometric changes using a 3DMM.

StyleGAN embeddings need to find a trade-off between high-fidelity image reconstruction and high-quality editing capabilities [Blau and Michaeli 2018; Tov et al. 2021]. Our approach seeks to find a sweet spot on this spectrum, but sometimes slight shifts in identity or other image attributes can occur. A related issue is a "showerdoor effect", resulting in flickering and texture details sticking to the screen rather than the face. We share these problems with previous work on this topic, as they stem to a large extent from inherent limitations of the generator [Karras et al. 2021]. In an orthogonal line of research, 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 supplemental).

It is potentially possible to train StyleGAN on a more diverse set of images such that free viewpoint capabilities naturally arise. We believe that our approach of extending the capabilities of a constrained model in a post-process is a viable solution presenting a good compromise, given the resources needed to train a GAN from scratch, and the quality of current methods achieved by careful alignment. An exciting avenue of future work would be to jointly train our embedding network and fine-tune the GAN to lift some of its limitations.

In conclusion, we have presented a method that allows the generation of a StyleGAN image from a free-viewpoint 3D camera, enabling these stunningly realistic images to be used in 3D applications, together with the semantic editing capabilities of previous methods. We believe that this is an important step forward in bridging the gap between powerful 2D image-processing learning solutions and the generation of fully editable 3D content for general use, with minimal content creation effort.

ACKNOWLEDGMENTS

This research was funded by the ERC Advanced grant FUNGRAPH No 788065 (http://fungraph.inria.fr). The authors are grateful to the OPAL infrastructure from Université Côte d'Azur for providing resources and support. The authors thank Ayush Tewari, Ohad Fried, and Siddhant Prakash for help with comparisons, Adrien Bousseau, Ayush Tewari, Julien Philip, Miika Aittala, and Stavros Diolatzis for proofreading earlier drafts, the anonymous reviewers for their valuable feedback, and all participants who helped capture the face datasets.

REFERENCES

Rameen Abdal, Yipeng Qin, and Peter Wonka. 2019. Image2StyleGAN: How to Embed Images Into the StyleGAN Latent Space?. In *ICCV*. 4432–4441.

Rameen Abdal, Yipeng Qin, and Peter Wonka. 2020. Image2StyleGAN++: How to Edit the Embedded Images?. In *CVPR*. 8296–8305.

Rameen Abdal, Peihao Zhu, Niloy Mitra, and Peter Wonka. 2021. Styleflow: Attribute-conditioned exploration of stylegan-generated images using conditional continuous normalizing flows. *ACM Transactions on Graphics (TOG)* (2021).

Mallikarjun B R, Ayush Tewari, Abdallah Dib, Tim Weyrich, Bernd Bickel, Hans-Peter Seidel, Hanspeter Pfister, Wojciech Matusik, Louis Chevallier, Mohamed Elgharib, and Christian Theobalt. 2021. PhotoApp: Photorealistic Appearance Editing of Head Portraits. In *ACM Transactions on Graphics (TOG, Proc. SIGGRAPH)*.

Thabo Beeler, Bernd Bickel, Paul Beardsley, Bob Sumner, and Markus Gross. 2010. High-quality single-shot capture of facial geometry. In *ACM Transactions on Graphics (TOG, Proc. SIGGRAPH)*. 1–9.

Sai Bi, Stephen Lombardi, Shunsuke Saito, Tomas Simon, Shih-En Wei, Kevyn Mcphail, Ravi Ramamoorthi, Yaser Sheikh, and Jason Saragih. 2021. Deep Relightable Appearance Models for Animatable Faces. *ACM Transactions on Graphics (TOG, Proc. SIGGRAPH)* (2021).

Volker Blanz and Thomas Vetter. 1999. A morphable model for the synthesis of 3D faces. In *Proc. SIGGRAPH*. 187–194.

Yochai Blau and Tomer Michaeli. 2018. The perception-distortion tradeoff. In *CVPR*. 6228–6237.

J. Blinn. 1988. Where am I? What am I looking at? (cinematography). *IEEE Computer Graphics and Applications* 8, 4 (1988), 76–81. https://doi.org/10.1109/38.7751

Kevin W Bowyer, Kyong Chang, and Patrick Flynn. 2006. A survey of approaches and challenges in 3D and multi-modal 3D+ 2D face recognition. *Computer vision and image understanding* 101, 1 (2006), 1–15.

Chris Buehler, Michael Bosse, Leonard McMillan, Steven Gortler, and Michael Cohen. 2001. Unstructured lumigraph rendering. In *Proc. SIGGRAPH*. 425–432.

Adrian Bulat and Georgios Tzimiropoulos. 2017. How far are we from solving the 2D & 3D Face Alignment problem? (and a dataset of 230,000 3D facial landmarks). In *ICCV*.

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

Eric R Chan, Marco Monteiro, Petr Kellnhofer, Jiajun Wu, and Gordon Wetzstein. 2021. pi-GAN: Periodic Implicit Generative Adversarial Networks for 3D-Aware Image Synthesis. In *CVPR*.

Anpei Chen, Ruiyang Liu, Ling Xie, Zhang Chen, Hao Su, and Yu Jingyi. 2021. SofGAN: A Portrait Image Generator with Dynamic Styling. *ACM Transactions on Graphics (TOG)* 41, 1, Article 1 (July 2021), 26 pages.

Marc Christie, Patrick Olivier, and Jean-Marie Normand. 2008. Camera control in computer graphics. In *Computer Graphics Forum*, Vol. 27. Wiley Online Library, 2197–2218.

Paul Debevec, Tim Hawkins, Chris Tchou, Haarm-Pieter Duiker, Westley Sarokin, and Mark Sagar. 2000. Acquiring the reflectance field of a human face. In *Proc. SIGGRAPH*. 145–156.

Paul Debevec, Camillo Taylor, and Jitendra Malik. 1996. Modeling and rendering architecture from photographs: A hybrid geometry-and image-based approach. In *Proc. SIGGRAPH*. 11–20.

Yu Deng, Jiaolong Yang, Dong Chen, Fang Wen, and Xin Tong. 2020. Disentangled and Controllable Face Image Generation via 3D Imitative-Contrastive Learning. In *CVPR*. 5154–5163.

Bernhard Egger, William AP Smith, Ayush Tewari, Stefanie Wuhrer, Michael Zollhoefer, Thabo Beeler, Florian Bernard, Timo Bolkart, Adam Kortylewski, Sami Romdhani, et al. 2020. 3d morphable face models - past, present, and future. *ACM Transactions on Graphics (TOG)* 39, 5 (2020), 1–38.

Zeev Farbman, Raanan Fattal, and Dani Lischinski. 2011. Convolution pyramids. *ACM Transactions on Graphics (TOG)* 30, 6 (2011), 175.

Zeev Farbman, Gil Hoffer, Yaron Lipman, Daniel Cohen-Or, and Dani Lischinski. 2009. Coordinates for instant image cloning. *ACM Transactions on Graphics (TOG)* 28, 3 (2009), 1–9.

Ohad Fried, Eli Shechtman, Dan B Goldman, and Adam Finkelstein. 2016. Perspective-aware Manipulation of Portrait Photos. *ACM Transactions on Graphics (TOG, Proc. SIGGRAPH)* (2016).

Guy Gafni, Justus Thies, Michael Zollhöfer, and Matthias Nießner. 2021. Dynamic Neural Radiance Fields for Monocular 4D Facial Avatar Reconstruction. In *CVPR*. 8649–8658.

Chen Gao, Yichang Shih, Wei-Sheng Lai, Chia-Kai Liang, and Jia-Bin Huang. 2020. Portrait Neural Radiance Fields from a Single Image. *arXiv preprint arXiv:2012.05903* (2020).

Jiahao Geng, Tianjia Shao, Youyi Zheng, Yanlin Weng, and Kun Zhou. 2018. Warp-guided gans for single-photo facial animation. *ACM Transactions on Graphics (TOG)* 37, 6 (2018), 1–12.

Kyle Genova, Forrester Cole, Daniel Vlasic, Aaron Sarna, William T Freeman, and Thomas Funkhouser. 2019. Learning shape templates with structured implicit functions. In *ICCV*. 7154–7164.

Abhijeet Ghosh, Graham Fyffe, Borom Tunwattanapong, Jay Busch, Xueming Yu, and Paul Debevec. 2011. Multiview face capture using polarized spherical gradient illumination. In *ACM Transactions on Graphics (TOG, Proc. SIGGRAPH Asia)*. 1–10.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in neural information processing systems*. 2672–2680.

Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. 2020. GANSpace: Discovering Interpretable GAN Controls. In *Proc. NeurIPS*.

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.

Ali Jahanian, Lucy Chai, and Phillip Isola. 2020. On the "steerability" of generative adversarial networks. In *ICLR*.

Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. 2020a. Training Generative Adversarial Networks with Limited Data. In *Proc. NeurIPS*.

Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. 2021. Alias-Free Generative Adversarial Networks. *CoRR* abs/2106.12423 (2021).

Tero Karras, Samuli Laine, and Timo Aila. 2019. A style-based generator architecture for generative adversarial networks. In *CVPR*. 4401–4410.

Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. 2020b. 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.

Zhanghan Ke, Kaican Li, Yurou Zhou, Qiuhua Wu, Xiangyu Mao, Qiong Yan, and Rynson WH Lau. 2020. Is a Green Screen Really Necessary for Real-Time Human Matting? *arXiv preprint arXiv:2011.11961* (2020).

Hyeongwoo Kim, Pablo Garrido, Ayush Tewari, Weipeng Xu, Justus Thies, Matthias Nießner, Patrick Pérez, Christian Richardt, Michael Zollöfer, and Christian Theobalt. 2018. Deep Video Portraits. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 163.

Hans Knutsson and C-F Westin. 1993. Normalized and differential convolution. In *CVPR*. 515–523.

Zhengqi Li, Wenqi Xian, Abe Davis, and Noah Snavely. 2020. Crowdsampling the Plenoptic Function. In *ECCV*.

Christophe Lino and Marc Christie. 2012. Efficient Composition for Virtual Camera Control. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Lausanne, Switzerland) *(SCA '12)*. Eurographics Association, 65–70.

Christophe Lino and Marc Christie. 2015. Intuitive and efficient camera control with the toric space. *ACM Transactions on Graphics (TOG)* 34, 4 (2015), 1–12.

Stephen Lombardi, Jason Saragih, Tomas Simon, and Yaser Sheikh. 2018. Deep appearance models for face rendering. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 1–13.

Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. 2019. Neural Volumes: Learning Dynamic Renderable Volumes from Images. *ACM Transactions on Graphics (TOG)* 38, 4, Article 65 (July 2019), 14 pages.

William R Mark, Leonard McMillan, and Gary Bishop. 1997. Post-rendering 3D warping. In *Proceedings of the 1997 symposium on Interactive 3D graphics*. 7–ff.

Ricardo Martin-Brualla, Noha Radwan, Mehdi S. M. Sajjadi, Jonathan T. Barron, Alexey Dosovitskiy, and Daniel Duckworth. 2021. NeRF in the Wild: Neural Radiance Fields for Unconstrained Photo Collections. In *CVPR*.

Iain Matthews and Simon Baker. 2004. Active appearance models revisited. *International journal of computer vision* 60, 2 (2004), 135–164.

Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. 2020. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In *ECCV*.

Koki Nagano, Huiwen Luo, Zejian Wang, Jaewoo Seo, Jun Xing, Liwen Hu, Lingyu Wei, and Hao Li. 2019. Deep face normalization. *ACM Transactions on Graphics (TOG)* 38, 6 (2019), 1–16.

A. Nagrani, J. S. Chung, and A. Zisserman. 2017. VoxCeleb: a large-scale speaker identification dataset. In *INTERSPEECH*.

Thu Nguyen-Phuoc, Chuan Li, Lucas Theis, Christian Richardt, and Yong-Liang Yang. 2019. Hologan: Unsupervised learning of 3d representations from natural images. In *ICCV*. 7588–7597.

Michael Niemeyer and Andreas Geiger. 2021. CAMPARI: Camera-Aware Decomposed Generative Neural Radiance Fields. arXiv:2103.17269

Xingang Pan, Bo Dai, Ziwei Liu, Chen Change Loy, and Ping Luo. 2021. Do 2D GANs Know 3D Shape? Unsupervised 3D Shape Reconstruction from 2D Image GANs. In *ICLR*.

Keunhong Park, Utkarsh Sinha, Jonathan T. Barron, Sofien Bouaziz, Dan B Goldman, Steven M. Seitz, and Ricardo Martin-Brualla. 2021. Nerfies: Deformable Neural Radiance Fields. *ICCV* (2021).

Omkar M. Parkhi, Andrea Vedaldi, and Andrew Zisserman. 2015. Deep Face Recognition. In *BMVC*. BMVA Press, Article 41, 41.1-41.12 pages.

Alec Radford, Luke Metz, and Soumith Chintala. 2016. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. In *ICLR 2016*.

Srinivas Rao, Rodrigo Ortiz-Cayon, Matteo Munaro, Aidas Liaudanskas, Krunal Chande, Tobias Bertel, Christian Richardt, Alexander JB, Stefan Holzer, and Abhishek Kar. 2020. Free-Viewpoint Facial Re-Enactment from a Casual Capture. In *SIGGRAPH Asia 2020 Posters*. 1–2.

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.

Johannes L Schonberger and Jan-Michael Frahm. 2016. Structure-from-motion revisited. In *CVPR*. 4104–4113.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. Facenet: A unified embedding for face recognition and clustering. In *CVPR*. 815–823.

Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. 2020. GRAF: Generative Radiance Fields for 3D-Aware Image Synthesis. In *NeurIPS*.

Mike Seymour, Chris Evans, and Kim Libreri. 2017. Meet mike: epic avatars. In *ACM SIGGRAPH 2017 VR Village*. 1–2.

Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou. 2020. Interpreting the latent space of gans for semantic face editing. In *CVPR*. 9243–9252.

Yichun Shi, Divyansh Aggarwal, and Anil K Jain. 2021. Lifting 2D StyleGAN for 3D-Aware Face Generation. In *CVPR*. 6258–6266.

Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. 2019. First Order Motion Model for Image Animation. In *NeurIPS*.

Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. 2020. Implicit neural representations with periodic activation functions. *NeurIPS* 33 (2020).

Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. 2019. Scene Representation Networks: Continuous 3D-Structure-Aware Neural Scene Representations. In *NeurIPS*.

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

Pratul P. Srinivasan, Boyang Deng, Xiuming Zhang, Matthew Tancik, Ben Mildenhall, and Jonathan T. Barron. 2021. NeRV: Neural Reflectance and Visibility Fields for Relighting and View Synthesis. In *CVPR*.

Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T. Barron, and Ren Ng. 2020. Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains. *NeurIPS* (2020).

Ayush Tewari, Mohamed Elgharib, Gaurav Bharaj, Florian Bernard, Hans-Peter Seidel, Patrick Pérez, Michael Zöllhofer, and Christian Theobalt. 2020a. StyleRig: Rigging StyleGAN for 3D Control over Portrait Images. In *CVPR*.

Ayush Tewari, Mohamed Elgharib, Mallikarjun BR, Florian Bernard, Hans-Peter Seidel, Patrick Pérez, Michael Zöllhofer, and Christian Theobalt. 2020b. PIE: Portrait Image Embedding for Semantic Control. *ACM Transactions on Graphics (TOG, Proc. SIGGRAPH Asia)* 39, 6.

A. Tewari, O. Fried, J. Thies, V. Sitzmann, S. Lombardi, K. Sunkavalli, R. Martin-Brualla, T. Simon, J. Saragih, M. Nießner, R. Pandey, S. Fanello, G. Wetzstein, J.-Y. Zhu, C. Theobalt, M. Agrawala, E. Shechtman, D. B Goldman, and M. Zollhöfer. 2020c. State of the Art on Neural Rendering. *Computer Graphics Forum (EG STAR 2020)* (2020).

Justus Thies, Michael Zollhöfer, and Matthias Nießner. 2019a. Deferred Neural Rendering: Image Synthesis using Neural Textures. *ACM Transactions on Graphics (TOG)* (2019).

Justus Thies, Michael Zollhöfer, Christian Theobalt, Marc Stamminger, and Matthias Nießner. 2019b. Image-guided neural object rendering. In *ICLR*.

Omer Tov, Yuval Alaluf, Yotam Nitzan, Or Patashnik, and Daniel Cohen-Or. 2021. Designing an Encoder for StyleGAN Image Manipulation. In *ACM Transactions on Graphics (TOG, Proc. SIGGRAPH)*.

Paul Upchurch, Jacob Gardner, Geoff Pleiss, Robert Pless, Noah Snavely, Kavita Bala, and Kilian Weinberger. 2017. Deep feature interpolation for image content changes. In *CVPR*. 7064–7073.

Ting-Chun Wang, Ming-Yu Liu, Andrew Tao, Guilin Liu, Jan Kautz, and Bryan Catanzaro. 2019. Few-shot Video-to-Video Synthesis. In *NeurIPS*.

Ziyan Wang, Timur Bagautdinov, Stephen Lombardi, Tomas Simon, Jason Saragih, Jessica Hodgins, and Michael Zollhofer. 2021. Learning Compositional Radiance Fields of Dynamic Human Heads. In *CVPR*. 5704–5713.

Shih-En Wei, Jason Saragih, Tomas Simon, Adam W Harley, Stephen Lombardi, Michal Perdoch, Alexander Hypes, Dawei Wang, Hernan Badino, and Yaser Sheikh. 2019. VR facial animation via multiview image translation. *ACM Transactions on Graphics (TOG)* 38, 4 (2019), 1–16.

Jonas Wulff and Antonio Torralba. 2020. Improving inversion and generation diversity in stylegan using a Gaussianized latent space. *arXiv preprint arXiv:2009.06529* (2020).

Sicheng Xu, Jiaolong Yang, Dong Chen, Fang Wen, Yu Deng, Yunde Jia, and Xin Tong. 2020. Deep 3D Portrait from a Single Image. In *CVPR*. 7710–7720.

Lei Yang, Yu-Chiu Tse, Pedro V Sander, Jason Lawrence, Diego Nehab, Hugues Hoppe, and Clara L Wilkins. 2011. Image-based bidirectional scene reprojection. In *ACM Transactions on Graphics (TOG, Proc. SIGGRAPH Asia)*. 1–10.

Tarun Yenamandra, Ayush Tewari, Florian Bernard, Hans-Peter Seidel, Mohamed Elgharib, Daniel Cremers, and Christian Theobalt. 2021. i3DMM: Deep Implicit 3D Morphable Model of Human Heads. In *CVPR*.

Xuan Yu, Rui Wang, and Jingyi Yu. 2010. Real-time depth of field rendering via dynamic light field generation and filtering. In *Computer Graphics Forum*, Vol. 29. 2099–2107.

Egor Zakharov, Aleksei Ivakhnenko, Aliaksandra Shysheya, and Victor Lempitsky. 2020. Fast Bi-layer Neural Synthesis of One-Shot Realistic Head Avatars. In *ECCV*.

Egor Zakharov, Aliaksandra Shysheya, Egor Burkov, and Victor Lempitsky. 2019. Few-shot adversarial learning of realistic neural talking head models. In *ICCV*. 9459–9468.

Jiakai Zhang, Xinhang Liu, Xinyi Ye, Fuqiang Zhao, Yanshun Zhang, Minye Wu, Yingliang Zhang, Lan Xu, and Jingyi Yu. 2021b. Editable Free-viewpoint Video Using a Layered Neural Representation. In *ACM Transactions on Graphics (TOG, Proc. SIGGRAPH)*.

Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. 2020. NeRF++: Analyzing and Improving Neural Radiance Fields. arXiv:2010.07492 [cs.CV]

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. 2018. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*. 586–595.

Yuxuan Zhang, Wenzheng Chen, Huan Ling, Jun Gao, Yinan Zhang, Antonio Torralba, and Sanja Fidler. 2021a. Image GANs meet Differentiable Rendering for Inverse Graphics and Interpretable 3D Neural Rendering. In *ICLR*.

Yajie Zhao, Zeng Huang, Tianye Li, Weikai Chen, Chloe LeGendre, Xinglei Ren, Ari Shapiro, and Hao Li. 2019. Learning perspective undistortion of portraits. In *ICCV*. 7849–7859.

Hang Zhou, Jihao Liu, Ziwei Liu, Yu Liu, and Xiaogang Wang. 2020. Rotate-and-Render: Unsupervised Photorealistic Face Rotation from Single-View Images. In *CVPR*. 5911–5920.

Jiapeng Zhu, Yujun Shen, Deli Zhao, and Bolei Zhou. 2020. In-domain GAN Inversion for Real Image Editing. In *ECCV*.

Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman, and Alexei A Efros. 2016. Generative visual manipulation on the natural image manifold. In *ECCV*. Springer, 597–613.

Jun-Yan Zhu, Zhoutong Zhang, Chengkai Zhang, Jiajun Wu, Antonio Torralba, Josh Tenenbaum, and Bill Freeman. 2018. Visual object networks: Image generation with disentangled 3D representations. *NeurIPS* 31 (2018), 118–129.