EpipolarGAN: Omnidirectional Image Synthesis with Explicit Camera Control

Christopher May and Daniel Aliaga

Purdue University, West Lafayette IN 47907, USA
may5@purdue.edu and aliaga@cs.purdue.edu

Abstract. In recent years, generative networks have achieved high quality results in 3D-aware image synthesis. However, most prior approaches focus on outside-in generation of a single object or face, as opposed to full inside-looking-out scenes. Those that do generate scenes typically require depth/pose information, or do not provide camera positioning control. We introduce EpipolarGAN, an omnidirectional Generative Adversarial Network for interior scene synthesis that does not need depth information, yet allows for direct control over the camera viewpoint. Rather than conditioning on an input position, we directly resample the input features to simulate a change of perspective. To reinforce consistency between viewpoints, we introduce an epipolar loss term that employs feature matching along epipolar arcs in the feature-rich intermediate layers of the network. We validate our results with comparisons to recent methods, and we formulate a generative reconstruction metric to evaluate multi-view consistency.

Keywords: Image generation· GAN· Interior scenes

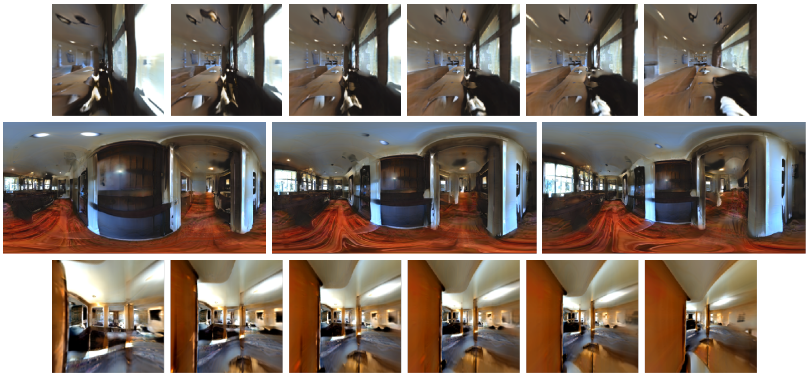

Fig. 1: Epipolar GAN. Our approach generates perspective and equirectangular images undergoing translation. Our network is trained without depth or camera pose data yet produces omnidirectional scenes with full control of camera position and with structural consistency between views.

1 Introduction

Recent advances in generative deep learning have enabled synthesis of 3D-aware imagery, where different views of the same object exhibit structural coherence. Most approaches are limited to single objects or faces, where the camera is typically constrained to a view facing the object, at a fixed distance. However, the task of generating full immersive environments remains a difficult problem. In this setting, the camera is no longer constrained but instead can move freely throughout the scene. This expanded domain, along with the large diversity of interior environments, presents a challenge for the representational capacity of existing networks. The problem is alleviated somewhat by incorporating knowledge of camera poses and depth during training [4, 11]. In the absence of such information, the task becomes more difficult.

We introduce a Generative Adversarial Network (GAN) that attempts to model the space of interior scenes with omnidirectional imagery, without requiring depth and/or camera pose information per image (e.g., unlabeled, unposed RGB images) (Fig. 1). Our network does not explicitly or implicitly represent the scene geometry or volume, yet allows for full camera movement within the generated environment. We achieve this with a set of input volumetric Fourier features which are projected onto a spherical sampling surface, which in turn enables a simulated shift in perspective in the output layer. To further reinforce multi-view consistency, we develop an additional loss term that exploits epipolar geometry constraints. We evaluate the structural coherence of our output with a new generative reconstruction metric and compare related prior methods.

Our contributions can be summarized as follows:

- we introduce a generative adversarial network with omnidirectional output and full camera control without requiring depth and/or camera pose information;
- we describe a loss term that utilizes epipolar geometry to encourage multi-view consistency;
- we define a polar filtering method that improves generated image quality near the poles (e.g., when looking upwards or downwards from the current viewpoint); and
- we measure the consistency of generated images with a new reconstruction metric.

2 Related Work

The past few years have seen a dramatic increase in the quality of generative AI imagery. Advancements in GANs [13] have resulted in photorealistic image synthesis [19, 21], with controllable attributes [37], and on large multi-modal datasets [5, 12]. Recently, diffusion models [17, 31, 35, 42] have also found success in synthesizing high quality 2D images. In parallel, the development of neural radiance fields (NeRFs) [3, 28] has driven rapid progress in 3D representations within a network. Although the original NeRF formulation is not generative (*i.e.*

it learns a single volume from a collection of images), there have been recent works [6, 7, 14, 32, 34, 36] that seek to combine NeRF with generative models in order to synthesize volumes for 3D-aware and multi-view consistent imagery.

Table 1: Summary of comparisons to related work. **Camera control:** whether the network or output enables explicit control over camera position. **Omnidirectional:** whether the network produces omnidirectional image content. **Without depth/pose:** whether depth or annotated camera pose is not required at training time.

	Camera control	Omnidirectional	Without depth	Without pose
BIPS [33]	✓	✓	✗	✓
GSN [11]	✓	✗	✗	✓
GAUDI [4]	✓	✗	✗	✗
DiffCollage [45]	✗	✓	✓	✓
CubeGAN [27]	✗	✓	✓	✓
EpipolarGAN (ours)	✓	✓	✓	✓

While many such works focus on single object or face synthesis from an "outside-in" pose distribution, typically over $SO(3)$, comparatively few methods have approached the task of full scene "inside-out" synthesis. BIPS [33] generates omnidirectional RGB-D images from partial depth input, which can be unprojected into a 3D mesh of an interior scene. Besides utilizing depth information, they also require prior knowledge of room layouts during training, limiting the ability to train on real datasets without detailed annotations. GSN [11] proposes a GAN architecture that models a scene as a set of local radiance fields, controlled by a hierarchy of global/local latent codes. Although the network has the capacity to generate large and complex scenes, the results lack consistency between views due to non-ideal upsampling. Further, the generated scenes closely resemble the scenes in the training dataset, bringing into question the generalizing power of the network – nonetheless, we compare to this method in the results section. Another recent work in this domain is GAUDI [4], which optimizes an empirical latent distribution to minimize reconstruction error of a generated radiance field across a set of camera trajectories. The empirical distribution is then sampled using a diffusion model. However, inference is slow due to the diffusion network, and training requires annotated camera pose trajectories and per-frame depth. Our method generates omnidirectional imagery without the use of depth or camera pose information.

Orthogonal to 3D-aware scene generation is the task of omnidirectional or panoramic image generation. This has been extensively studied, however many methods focus on horizontal panoramas [23, 24, 40, 44] or infinite image generation [2, 10, 25, 38] but do not consider true omnidirectional output that includes the polar regions. Some works focus on omnidirectional high dynamic range (HDR) image generation [9, 43] for use in lighting tasks. While these HDR methods output full omnidirectional images, they are not intended for direct visual-

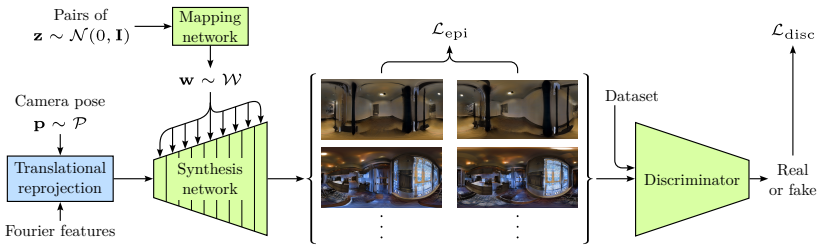

Fig. 2: Network Layout. Camera poses $\mathbf{p}$ are used to reproject the Fourier features prior to synthesis. Latents $\mathbf{z}$ are sampled and duplicated into pairs, producing pairs of identical scenes with different camera positions. Epipolar loss $\mathcal{L}_{\text{epi}}$ is evaluated on the pairs.

ization and thus they do not avoid acyclic discontinuities and polar distortions. Other works turn to a cube-map representation; e.g., CubeGAN [27] employs volumetric Fourier features and incorporates a padding resampling operation to maintain continuity between cube faces. However, the resampling operation is expensive and leads to slow training time. CubeGAN also demonstrates camera movement within a scene, but only when trained with a densely sampled image set of a single scene. Further, control of the camera is embedded in the latent space and difficult to map to spatial coordinates. Recently, DiffCollage [45] was proposed as a diffusion model that takes into account graph topologies during image generation. Among the demonstrated configurations is a cube-map representation, which is conditioned on a semantic segmentation map. Both CubeGAN and DiffCollage produce seamless, distortionless panoramic images, but neither method allows for controlling the viewpoint.

Our work in relation to others can be summarized by Table 1. We propose an approach that generates full, seamless and distortionless omnidirectional images, allowing for direct control over the camera position, and without requiring depth and/or camera pose information at training time.

3 Methodology

We first describe our base network including our modifications to support equirectangular imagery. Next, we incorporate a camera position as an input to the initial network layer, and explain the resampling process to simulate perspective change. We also address the issue of polar distortion with additional frequency filtering near the image boundaries. Finally, we introduce our epipolar loss term to further reinforce consistency between generated viewpoints (Fig. 2).

3.1 Network Layout

We begin by building from the StyleGAN3 [19] network, for which we give a brief overview. StyleGAN3 consists of a mapping network which transforms input latent vectors into an intermediate representation, and a synthesis network whose

individual layers generate image features at progressively higher resolutions. It is distinguished from its predecessors [18,20,21] by its strict control over aliasing and allowed frequencies within the network. This is accomplished with specifically designed filters for upsampling and downsampling, which are additionally used to band-limit the nonlinear activation function in each layer. The input to the first layer is replaced by features generated from a fixed set of 2D frequencies and phases, which enable explicit control over the orientation and offset of the generated images. The resulting network is equivariant to 2D translation (StyleGAN3-T configuration) or rotation (StyleGAN3-R configuration).

In modifying this network for equirectangular image generation, we specifically choose the StyleGAN3-T configuration. This selection is because for an equirectangular projection, a horizontal translation in image space is equivalent to a 3D rotation about the Y axis, so translational equivariance allows the same scene to be generated regardless of its equatorial orientation. Neither vertical image translation nor planar rotation are applicable to equirectangular projections, so we do not consider the rotationally equivariant StyleGAN3-R configuration.

Before we begin to modify the network, we must first consider the implications of generating equirectangular images. Our first observation is that the generated signal is no longer planar, but spherical. As such, we must redefine the frequencies that make up the signal to be angular frequencies. In an equirectangular projection, each column of pixels subtends an angle of 180°, and each row 360°. This requires us to support non-square aspect ratios. Specifically, we force the width of the generated images to be twice the height. For the purposes of determining frequency cutoffs for each layer, we consider the height to be the angular sampling rate.

Our use of angular frequencies must now reshape the construction of the input layer to the synthesis module. To this end we extend the volumetric frequency scheme of [27]. Specifically, we represent the input signal as a spherical manifold of unit radius, along which we sample the values of a fixed set of 3D frequencies and phases. The resultant signal is projected onto our equirectangular image plane and subsequently used as the input to the first layer of the network. Similar to the base StyleGAN3 input layer, we can directly scale and rotate the frequencies in 3D, correspondingly transforming the output image in a fully equivariant network.[1]

Our second observation is that a spherical manifold has finite area, as opposed to a plane. As a consequence, the image content at the borders of our equirectangular representation must seamlessly align with content at the opposing edges. While the input layer is seamless by construction, repeated convolutions in subsequent layers quickly accumulate discontinuities. Ignoring image borders [25] or expensive image resampling operations [27] are typical options. However, for our equirectangular projection, we only need to copy pixel values into the respective padding regions, without resampling, to ensure spherical agreement. Specifically, the equirectangular image is horizontally cyclic and vertically reflective with a

[1] Since our network is only planar-translationally equivariant, only 3D rotations about the Y axis purely change the orientation of the output.

180° longitudinal shift. We perform this border fix at each layer before the upsampled activation.

StyleGAN3 applies adaptive augmentation [18] to both generated and dataset images prior to discrimination. Augmentations include both geometric and color-space transformations, and serve to strengthen the discriminator's ability to generalize. We extend the set of geometric augmentations to account for the spherical topology of our images. Instead of transforming the image as a 2D plane, we consider that each input pixel samples a 3D direction and accordingly convert from spherical coordinates $\mathbf{x}_i \in \mathbb{R}^2$ to Cartesian coordinates $\mathbf{y}_i \in \mathbb{R}^3$. Transformations $\mathbf{M} \in \mathbb{R}^{3\times3}$ are directly applied to these 3D vectors, which are then normalized and converted back into spherical coordinates $\bar{\mathbf{x}}_i \in \mathbb{R}^2$ before resampling the image:

$$\mathbf{y}_i = \begin{bmatrix} \cos(\mathbf{x}_{i,\phi})\cos(\mathbf{x}_{i,\theta}) \\ \sin(\mathbf{x}_{i,\phi}) \\ \cos(\mathbf{x}_{i,\phi})\sin(\mathbf{x}_{i,\theta}) \end{bmatrix}, \tag{1}$$

$$\bar{\mathbf{y}}_i = \frac{\mathbf{M} \times \mathbf{y}_i}{\|\mathbf{M} \times \mathbf{y}_i\|}, \tag{2}$$

$$\bar{\mathbf{x}}_i = \begin{bmatrix} \arctan(\bar{\mathbf{y}}_{i,z}/\bar{\mathbf{y}}_{i,x}) \\ \arcsin(\bar{\mathbf{y}}_{i,y}) \end{bmatrix}. \tag{3}$$

The resulting output is still an equirectangular projection, but the omnidirectional content is scaled and rotated with respect to the input. Note that these transformations exclude 3D translation, for several reasons. We do not have depth information, so we cannot accurately translate in 3D. Even if we did have depth, resampling would cause issues with occlusions and disocclusions in the image. In the absence of depth, we consider the omnidirectional content to exist infinitely far away, and thus any 3D translations would not affect the output.

With these modifications, the network can train on and generate equirectangular projections of omnidirectional images. In the following sections we will discuss our additions to enable 3D translation of the camera while maintaining consistency between viewpoints.

3.2 Translational Reprojection

Camera movement in the output scene is enabled by reprojection of the features at the input layer. The features are defined as a fixed set of frequencies and phases in 3D (i.e., planar sine waves), which are sampled along the surface of a unit sphere centered at the origin. Given a camera position $\mathbf{p} \in \mathbb{R}^3$, we build the initial equirectangular image by casting rays $\hat{\mathbf{u}}$ through each pixel and intersecting with the sampling sphere. The value of the sinusoid at the intersection point $\mathbf{r}$ becomes the pixel value. Given that the sphere has unit radius and $\mathbf{p}$ is within the sphere, the intersection equation reduces to:

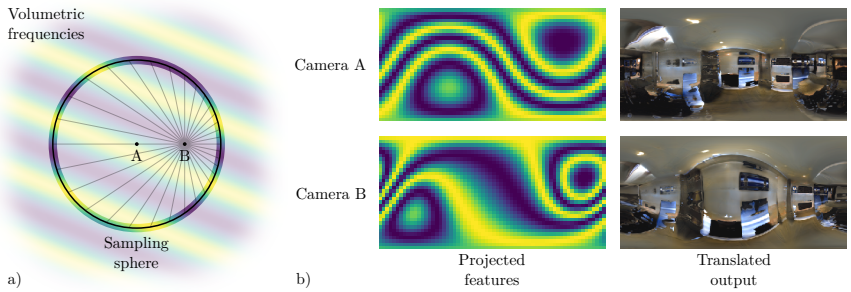

Fig. 3: Translational reprojection. a) Top-down view of two camera viewpoints within the sampling sphere, with A at the center and rays projected from B. b) Input features projected onto the views of both cameras, and the corresponding projected features and exemplary translated generated output.

$$\mathbf{r} = \mathbf{p} + t\,\hat{\mathbf{u}}, \qquad t = -\hat{\mathbf{u}} \cdot \mathbf{p} + \sqrt{(\hat{\mathbf{u}} \cdot \mathbf{p})^2 - \mathbf{p} \cdot \mathbf{p} + 1}. \tag{4}$$

Figure 3 demonstrates the effect of translational reprojection in the input layer. As the input signal is warped according to the camera's position, the perspective is similarly translated in the output image. We sample camera positions with a truncated Gaussian distribution $\mathcal{P} = \mathcal{N}(0, \sigma^2\mathbf{I})$; $\|\mathbf{p} \sim \mathcal{P}\| \leq 0.75$ such that any sampled position is guaranteed to be within the sampling sphere, with distance at least 0.25 from the sphere's surface. In practice we set $\sigma^2 = [0.125, 0.00625, 0.125]$ as an ad-hoc estimation of the dataset camera distribution, as the perspective is typically in human height range, and not very close to the walls of the scene.

3.3 Epipolar Loss

In reprojecting the input features according to the camera position, we establish a strong foundation for the network to build view-dependent imagery upon, while retaining the same overall scene characteristics. However, the network is not equivariant to changes in camera position. This can be demonstrated by considering two viewpoints given the same latent vector, as shown in Figure 3. Camera B is closer to the surface of the sampling sphere than camera A, so the sampled input features appear larger from B's perspective than from A's. In other words, the same input features have lower angular frequency in one view than the other, causing successive bandlimited layers to filter the signals differently. On one hand, this is actually a desirable property. Camera movement should incur parallax, which in a completely equivariant system would not occur. On the other hand, without strict handling of the synthesized content, the generator has too much freedom and develops view-inconsistent imagery.

We design an additional loss term to incentivize learning structurally-consistent generation in a stochastic manner, without sabotaging the network's ability to

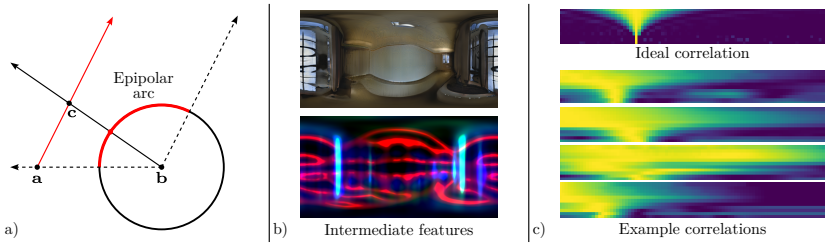

Fig. 4: Epipolar loss. a) Epipolar arc formed by projecting ray $\overrightarrow{\mathbf{ac}}$ onto the sphere centered at **b**. The feature **c** appears at the extremes of the arc when it is located at **a** or infinitely far from **a**, respectively. b) Despite the center wall having relatively little textures in the RGB output (top), the intermediate layers contain rich features suitable for feature matching (bottom). c) The ideal correlation (top) along an arc (horizontal axis) has a funnel shape. The earlier layers (vertical axis) contain lower frequencies than later layers, thus the correlation width progressively shrinks. Example correlations found during training (bottom) also have a rough funnel shape.

introduce multi-view disparity. Our formulation has its roots in epipolar geometry [15], specifically the notion that an observed point **c** of unknown depth in one camera **a** can be seen in another camera **b** along the ray $\overrightarrow{\mathbf{ac}}$ projected onto **b**'s image plane. In the omnidirectional setting (Fig. 4a), this projected line becomes an arc: the projection of $\overrightarrow{\mathbf{ac}}$ onto the sphere centered at **b**.

We use this concept to design our loss term. If an image feature is visible from one view, we should find it along an arc in a second view. During training, we sample $\frac{n}{2}$ pairs of identical latents $\mathbf{z} \sim \mathcal{N}(0, \mathbf{I})$ and n camera positions $\mathbf{p} \sim \mathcal{P}$, resulting in $\frac{n}{2}$ pairs of the same scene from two different viewpoints. From each view, we sample k random features $\hat{\mathbf{u}} \in S^2$ and their corresponding epipolar arcs in the opposite view, and attempt to find the features in their arcs via cross-correlation feature matching. A strong set of correlations indicates a high degree of structural coherence, because it means that the same image features are present in both views in locations predicted by the relative camera displacement.

In feature matching literature, it's important to choose strong features in the first place (*e.g.*, SIFT features [26]), because not all image patches will produce strong correlations between images. A blank, featureless wall for example would be a poor choice. While such walls do appear in our dataset and generated images, we note that they are only "featureless" in the output RGB layer. The intermediate layers have higher channel depth and much richer features (Fig. 4b), thus we perform cross-correlational feature matching within the internal layers of the synthesis network to avoid sampling bad features.

Let $\mathbf{C} \in \mathbb{R}^{l \times m}$ be a set of correlations for a single feature along an arc of length m in l intermediate layer images. The loss function consists of two parts. First, we desire a strong correlation in a single location along the arc, and weak correlations elsewhere. Thus, we normalize $\mathbf{C}$ to the $[0 \ldots 1]$ range independently per layer and penalize the mean value:

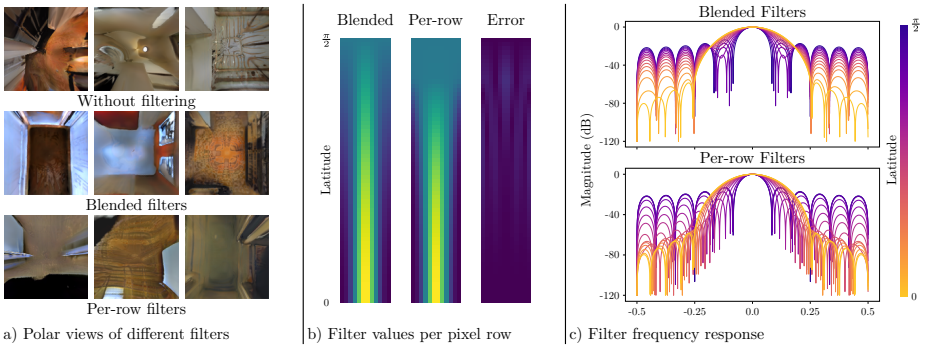

Fig. 5: Polar Filtering. a) Polar views when trained without filtering, and with blended and per-row filtering. No filtering causes "pinching" near the poles, while blended filters and per-row filters alleviate the effect to a degree. b) Visualization of blended and per-row upsampling filters, and the absolute error between them (max error is 0.027). c) Frequency response of the filters from b). Per-row filters have more aggressive attenuation in the stopband than blended filters.

$$\bar{\mathbf{C}} = \frac{\mathbf{C} - \min_m(\mathbf{C})}{\max_m(\mathbf{C}) - \min_m(\mathbf{C})}, \qquad \mathcal{L}_{\text{mean}} = \frac{1}{l \cdot m} \sum_{l,m} \bar{\mathbf{C}}. \tag{5}$$

Additionally, we want the strongest correlations along the arcs to line up between the layers. We calculate the mean of $\bar{\mathbf{C}}$ across all layers, representing the alignment of strongest correlations along the arc. A high value at a particular location indicates that multiple layers had strong correlations in the same spot. Then we take the maximum along the arc and penalize values less than 1:

$$\mathcal{L}_{\text{align}} = 1 - \max_m \left(\frac{1}{l} \sum_l \bar{\mathbf{C}} \right). \tag{6}$$

We combine these two terms to arrive at our epipolar loss:

$$\mathcal{L}_{\text{epi}} = \mathcal{L}_{\text{mean}} + \beta \mathcal{L}_{\text{align}}. \tag{7}$$

In practice, we set $\beta = 10$ and $k = 16$. l is the number of layers in the synthesis network, and m depends on the sampling rate and angle between the feature and the epipole. Figure 4c shows the ideal correlation shape incentivized by our loss, along with example correlations found during training. More details on epipolar arc sampling and loss weight can be found in the supplementary material.

3.4 Polar Filtering

One challenge of generating equirectangular imagery is in preventing distortions or "pinching" effects due to the extreme disparity in the angular sampling rate

near the poles versus at the equator. We observe that each row of pixels projects a circle onto the unit sphere of circumference C as a function of the latitude ϕ:

$$C = 2\pi\cos(\phi). \tag{8}$$

As a result, pixel rows near the poles have a much shorter arc length than those near the equator, yet our bandlimiting filters do not make any distinction between them. Allowing the same frequencies in the top row of pixels as in the middle results in undesired higher angular frequencies near the poles in the output, causing the aforementioned distortions. We address this issue with spatially varying filters.

The ideal spatially varying filters to use would need $\frac{h}{2} + 1$ filters instead of just 1 and it would be quite time consuming computationally. As a more efficient alternative, we experiment with an approximation of per-row filters by blending between the initial "equatorial" filter and the extreme "polar" filter, using $\cos(\phi)$ as the blend weight. This yields a roughly $4\times$ increase in speed as compared to a different filter per row. Figure 5b) shows a visual representation of the blended and per-row filters from the zenith to the equator, along with the absolute difference between them, and Figure 5c) plots the frequency response for both variants. While the per-row filters exhibit stronger attenuation near the equator, the attenuation drops off near the poles in both methods. Larger filters would help here, but training time becomes prohibitively slow.

Figure 5a) shows example outputs of our network when trained without polar filtering and with per-row and blended filtering enabled. Due to the long training time, filtering was enabled on a model trained without filtering and retrained for a small number of iterations. While polar filtering visually reduces the amount of distortion, the effect is not completely absent. We suspect that training for a longer time would help this further.

4 Evaluation

We evaluate our network with an ablation study for each of our additions. We also compare to other related works both qualitatively and quantitatively. Finally, to evaluate reconstruction quality, we introduce a new metric based on structure-from-motion.

4.1 Training Configuration

We run all our experiments on RGB images extracted from the Pano3D dataset [1], in total consisting of 35k equirectangular images of interior spaces, at a resolution of 512×256 pixels. Aside from our previously described modifications, we reduce the number of synthesis layers from 14 to 12, which made training more stable. Otherwise, we train with the default configuration settings of StyleGAN3, using a batch size of 32. Tuning the R1 regularization weight to a value of 16 yielded the best results. Models trained on a machine with two NVIDIA

A100 GPUs at an average speed of 52 seconds per 1000 images (s/kimg) without any polar filtering (see Sec. 3.4), at 236 s/kimg using our blended filter approximation, and at 983 s/kimg using approximately ideal per-row filters.

4.2 Reconstruction

To evaluate the effectiveness of our epipolar loss term (Section 3.3), we describe a metric based on global structure from motion (SfM) [29] with the goal of measuring the multi-view coherence of our generated scenes. Intuitively, a low reconstruction error across many scenes indicates a tendency to synthesize high quality image-space correspondences between positions. We do not have ground truth scene geometry to compare to, so we rely on measuring the error of reconstructed camera poses, for which we do have ground truth.

Let S be a set of scenes represented by latent vectors. For a given scene $s \in S$, we generate n views with camera positions $\mathbf{p}_{i\in\{1\ldots n\},s} \sim \mathcal{P}$, and then recover a subset of m_s reconstructed camera positions $\mathbf{q}_{j\subseteq\{1\ldots n\},s}$ using OpenMVG's [30] global SfM solver with the equirectangular projection model. The reconstructed camera positions are fit to the ground truth camera positions with a homogeneous rigid transformation $\mathbf{M}_s \in \mathbb{R}^{3\times4}$ obtained from the Kabsch-Umeyama algorithm [41]. We calculate the mean over all scenes and the RMSE between the fit reconstructed camera positions $\hat{\mathbf{q}}_{j,s} = \mathbf{M}_s \times [\mathbf{q}_{j,s}^T, 1]^T$ and ground truth positions:

$$\text{SfM RMSE} = \frac{1}{|S|}\sum_{s\in S}\text{RMSE}(\mathbf{p}_{j,s}, \hat{\mathbf{q}}_{j,s}) \tag{9}$$

The subset of recovered poses is also an indicator of reconstruction quality, and is obtained by calculating the mean fraction of recovered poses:

$$\text{SfM NP} = \frac{1}{|S|}\sum_{s\in S}\frac{m_s}{n} \tag{10}$$

In practice, we let $|S| = 1000$ and $n = 16$ to balance compute time with generalization over a large number of scenes. We report both RMSE and NP in our ablation study in Table 2.

4.3 Ablation Study

We study the effect of each of our modifications to the network in Table 2. Each configuration described below is trained to only 2000 kimgs (because of the lengthy training time) and we measure SfM reconstruction error and pose fraction. Configuration A is our initial baseline of StyleGAN3 with translational equivariance. The only change we have made here is to enforce 2:1 rectangular outputs. It does not support camera motion. Configuration B enables the set of modifications to support spherical imagery described in Section 3.1 and Section 3.2, namely: spherical input Fourier features with translational reprojection,

equirectangular border fixing via pixel blitting, and spherical augmentation operations. Configuration C applies additional filtering near the polar regions, using approximate ideal per-row filters (Sec. 3.4). Due to the lengthy training time, this configuration was resumed from configuration B at 1900 kimgs and trained for an additional 100 kimgs. Finally, configuration D includes our epipolar loss term (Sec. 3.3) also trained to 2000 kimgs (the results of it being trained for more time is in Table 3). As in configuration C, this configuration was trained without polar filtering for 1900 kimgs, and then resumed with polar filtering enabled for 100 additional kimgs.

Table 2: Ablation study

Configuration @ 2000 kimgs	SfM RMSE $\times 10^{-3} \downarrow$	SfM NP $\uparrow$
A: StyleGAN3-T [19]	–	–
B: + Trans. reproj.	7.47 ± 8.45	0.731 ± 0.391
C: + Polar filtering*	7.36 ± 6.88	0.75 ± 0.381
D: + **Epipolar loss**	**7.25** ± 7.15	**0.775** ± 0.368

For the first (A) and last (D) configuration, we also measure the FID [16] as well a custom variant of the FID, FID-Poles, that specifically looks at the polar regions of the generated images to quantify polar distortion effects. This is done by reprojecting the equirectangular images to two 90° field of view perspective images of size 128×128 pixels, facing up and down respectively, before computing the FID as normal. StyleGAN3-T has an FID of 16.7 and FID-Poles of 73.0. Our configuration D has an FID of 15.0 and a FID-Poles of 58.6. This implies that at this abbreviated training stage, our method is able to produce imagery of similar to (though slightly better than) StyleGAN3 but with camera control ability. Further, our method's improved FID-Poles measure reflects the betterment provided by our pipeline, especially the polar filtering component, for observing the poles.

4.4 Comparisons

We compare to related works both visually and quantitatively. Figure 6 shows visual comparisons of equirectangular and bottom-facing views from our method as well as that of StyleGAN3 [19], GSN [11], and CubeGAN [27]. Since StyleGAN3 is designed for 2D planar images, it does not model the cyclic topology of omnidirectional images, causing seams and polar distortions at the boundary. GSN does not natively output equirectangular images, but we produce such images by stitching together perspective outputs in a cube layout. Due to non-ideal upsampling, artifacts at the borders of each image are visible in the combined output. CubeGAN is specifically designed for omnidirectional imagery, and does not produce any seams or distortions in its output. However, we note that Cube-

Fig. 6: Visual comparisons to other works. StyleGAN3 does not specifically handle equirectangular images, so the output is not cyclic and meridian seams are visible in the center and bottom views. Bottom views also show polar distortion. GSN equirectangular images are created by stitching together 90° field of view perspective outputs, which exhibit seams at each image's boundary. Both CubeGAN and EpipolarGAN's outputs have neither seams nor distortions, but CubeGAN does not enable camera control.

Table 3: FID and SfM of related works.

	FID ↓	SfM RMSE $\times 10^{-3}$ ↓	SfM NP ↑
GSN [11]	43.32	7.80	**0.77**
GAUDI [4]	18.52	–	–
EpipolarGAN (ours)	**10.78**	**6.64**	0.716

GAN does not allow for camera movement. Our approach also does not contain seams, while also allowing for positional control over the camera.

Table 3 has quantitative comparisons between our method and GSN [11] and GAUDI [4]. We reproduce FID values from [4] on the VLN-CE [22] dataset, which uses Matterport3D [8] environments, a subset of the Pano3D [1] dataset we train on. Our network is trained on Pano3D for 20k kimgs. We note that because of the different datasets, the FID values are not directly comparable. We also measure the SfM reconstruction metric on GSN using the pretrained weights provided by the authors on the Replica [39] dataset. At the time of writing GAUDI did not have available source code.

5 Limitations & Conclusion

We have presented a network for learning to synthesize omnidirectional images of interior scenes. By resampling input features, we enable explicit camera motion within a generated scene. The network is incentivized to produce geometrically correct and multi-view consistent imagery by use of an additional epipolar loss term, which matches features along epipolar arcs in the intermediate layers of the network. Our results are validated with an ablation study, by measuring the reconstruction error of generated scenes, and in comparisons to related works.

One particularly overt limitation is that we do not actually have full range of motion with the camera. Since the translational reprojection step essentially offsets the sampling sphere in space, we cannot move the camera outside of its radius. Alternatively, instead of geometrically translating the sampling sphere, we could offset the phase of the sampled input frequencies along the direction of motion, which would allow for unbounded translation. However, doing so significantly alters the image content between views and in our experiments this has led to unstable training. Stabilizing this approach is an avenue of future work.

Although we focus on multi-view consistent output, our approach does not inherently model 3D scene geometry. As a result, complex environments are not easily represented. In particular, occlusions pose a challenge since the appearance and disappearance of objects is at odds with our epipolar loss. Our primary advantage over radiance field representations is facilitating the use of un-annotated images without depth or camera pose information, and training on single-image scenes (*e.g.*, without image pairs of the same scene). Under these conditions, the task of full-scene omnidirectional synthesis remains an unsolved problem.

Acknowledgements

This project was funded in part by NSF Grant #2107096 and NSF Grant #1835739.

References

1. Albanis, G., Zioulis, N., Drakoulis, P., Gkitsas, V., Sterzentsenko, V., Alvarez, F., Zarpalas, D., Daras, P.: Pano3D: A holistic benchmark and a solid baseline for 360° depth estimation. In: CVPR. pp. 3727–3737 (2021). https://doi.org/10.1109/CVPRW53098.2021.00413 10, 14
2. Bar-Tal, O., Yariv, L., Lipman, Y., Dekel, T.: MultiDiffusion: Fusing diffusion paths for controlled image generation. In: ICML (2023). https://doi.org/10.5555/3618408.3618482 3
3. Barron, J.T., Mildenhall, B., Tancik, M., Hedman, P., Martin-Brualla, R., Srinivasan, P.P.: Mip-NeRF: A multiscale representation for anti-aliasing neural radiance fields. In: ICCV. pp. 5855–5864 (2021). https://doi.org/10.1109/ICCV48922.2021.00580 2
4. Bautista, M.A., Guo, P., Abnar, S., Talbott, W., Toshev, A., Chen, Z., Dinh, L., Zhai, S., Goh, H., Ulbricht, D., Dehghan, A., Susskind, J.: GAUDI: A neural architect for immersive 3d scene generation. In: NeurIPS. pp. 25102–25116 (2022). https://doi.org/10.48550/arXiv.2207.13751 2, 3, 14
5. Brock, A., Donahue, J., Simonyan, K.: Large scale GAN training for high fidelity natural image synthesis. In: ICLR (2019). https://doi.org/10.48550/arXiv.1809.11096 2
6. Chan, E.R., Lin, C.Z., Chan, M.A., Nagano, K., Pan, B., Mello, S.D., Gallo, O., Guibas, L.J., Tremblay, J., Khamis, S., Karras, T., Wetzstein, G.: Efficient geometry-aware 3D generative adversarial networks. In: CVPR. pp. 16123–16133 (2022). https://doi.org/10.1109/CVPR52688.2022.01565 3
7. Chan, E.R., Monteiro, M., Kellnhofer, P., Wu, J., Wetzstein, G.: pi-GAN: Periodic implicit generative adversarial networks for 3D-aware image synthesis. In: CVPR. pp. 5799–5809 (2021). https://doi.org/10.1109/CVPR46437.2021.00574 3
8. Chang, A., Dai, A., Funkhouser, T., Halber, M., Niebner, M., Savva, M., Song, S., Zeng, A., Zhang, Y.: Matterport3d: Learning from rgb-d data in indoor environments. In: 2017 International Conference on 3D Vision (3DV). pp. 667–676 (2017). https://doi.org/10.1109/3DV.2017.00081 14
9. Chen, Z., Wang, G., Liu, Z.: Text2light: Zero-shot text-driven hdr panorama generation. ACM Trans. Graph. **41**(6) (2022). https://doi.org/10.1145/3550454.3555447 3
10. Cheng, Y.C., Lin, C.H., Lee, H.Y., Ren, J., Tulyakov, S., Yang, M.H.: In&Out: Diverse image outpainting via GAN inversion. In: CVPR. pp. 11431–11440 (2022). https://doi.org/10.1109/CVPR52688.2022.01114 3
11. DeVries, T., Bautista, M.A., Srivastava, N., Taylor, G.W., Susskind, J.M.: Unconstrained scene generation with locally conditioned radiance fields. In: ICCV. pp. 14304–14313 (2021). https://doi.org/10.1109/ICCV48922.2021.01404 2, 3, 12, 14
12. Esser, P., Rombach, R., Ommer, B.: Taming transformers for high-resolution image synthesis. In: CVPR. pp. 12873–12883 (2021). https://doi.org/10.1109/CVPR46437.2021.01268 2

13. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: NeurIPS. vol. 27 (2014). https://doi.org/10.48550/arXiv.1406.2661 2
14. Gu, J., Liu, L., Wang, P., Theobalt, C.: StyleNeRF: A style-based 3D aware generator for high-resolution image synthesis. In: ICLR (2022). https://doi.org/10.48550/arXiv.2110.08985 3
15. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press, ISBN: 0521540518, second edn. (2004) 8
16. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: GANs trained by a two time-scale update rule converge to a local nash equilibrium. In: NeurIPS. vol. 30 (2017). https://doi.org/10.48550/arXiv.1706.08500 12
17. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. In: NeurIPS (2020). https://doi.org/10.48550/arXiv.2006.11239 2
18. Karras, T., Aittala, M., Hellsten, J., Laine, S., Lehtinen, J., Aila, T.: Training generative adversarial networks with limited data. In: NeurIPS. vol. 33, pp. 12104–12114 (2020). https://doi.org/10.48550/arXiv.2006.06676 5, 6
19. Karras, T., Aittala, M., Laine, S., Härkönen, E., Hellsten, J., Lehtinen, J., Aila, T.: Alias-free generative adversarial networks. In: NeurIPS. vol. 34, pp. 852–863 (2021). https://doi.org/10.48550/arXiv.2106.12423 2, 4, 12
20. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. In: CVPR. pp. 4396–4405 (2018). https://doi.org/10.1109/CVPR.2019.00453 5
21. Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., Aila, T.: Analyzing and improving the image quality of StyleGAN. In: CVPR. pp. 8107–8116 (2020). https://doi.org/10.1109/CVPR42600.2020.00813 2, 5
22. Krantz, J., Wijmans, E., Majumdar, A., Batra, D., Lee, S.: Beyond the nav-graph: Vision-and-language navigation in continuous environments. In: ECCV. pp. 104–120 (2020). https://doi.org/10.1007/978-3-030-58604-1_7 14
23. Li, J., Bansal, M.: Panogen: Text-conditioned panoramic environment generation for vision-and-language navigation. In: NeurIPS. vol. 36, pp. 21878–21894 (2023). https://doi.org/10.48550/arXiv.2305.19195 3
24. Lin, C.H., Chang, C.C., Chen, Y.S., Juan, D.C., Wei, W., Chen, H.T.: COCO-GAN: Generation by parts via conditional coordinating. In: ICCV. pp. 4512–4521 (2019). https://doi.org/10.1109/ICCV.2019.00461 3
25. Lin, C.H., Cheng, Y.C., Lee, H.Y., Tulyakov, S., Yang, M.H.: InfinityGAN: Towards infinite-pixel image synthesis. In: ICLR (2022). https://doi.org/10.48550/arXiv.2104.03963 3, 5
26. Lowe, D.G.: Object recognition from local scale-invariant features. In: ICCV. vol. 2, pp. 1150–1157 (1999). https://doi.org/10.1109/ICCV.1999.790410 8
27. May, C., Aliaga, D.: CubeGAN: Omnidirectional image synthesis using generative adversarial networks. In: Eurographics. vol. 42, pp. 213–224 (2023). https://doi.org/10.1111/CGF.14755 3, 4, 5, 12
28. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: NeRF: Representing scenes as neural radiance fields for view synthesis. In: ECCV (2020). https://doi.org/10.1007/978-3-030-58452-8_24 2
29. Moulon, P., Monasse, P., Marlet, R.: Global fusion of relative motions for robust, accurate and scalable structure from motion. In: ICCV. pp. 3248–3255 (2013). https://doi.org/10.1109/ICCV.2013.403 11
30. Moulon, P., Monasse, P., Perrot, R., Marlet, R.: OpenMVG: Open multiple view geometry. In: RRPR. pp. 60–74 (2016). https://doi.org/10.1007/978-3-319-56414-2_5 11

31. Nichol, A.Q., Dhariwal, P.: Improved denoising diffusion probabilistic models. In: ICML. pp. 8162–8171 (2021). https://doi.org/10.48550/arXiv.2102.09672 2
32. Niemeyer, M., Geiger, A.: GIRAFFE: Representing scenes as compositional generative neural feature fields. In: CVPR (2021). https://doi.org/10.1109/CVPR46437.2021.01129 3
33. Oh, C., Cho, W., Chae, Y., Park, D., Wang, L., Yoon, K.J.: BIPS: Bi-modal indoor panorama synthesis via residual depth-aided adversarial learning. In: ECCV. pp. 352–371 (2022). https://doi.org/10.1007/978-3-031-19787-1_20 3
34. Rebain, D., Matthews, M., Yi, K.M., Lagun, D., Tagliasacchi, A.: LOLNeRF: Learn from one look. In: CVPR. pp. 1558–1567 (2022). https://doi.org/10.1109/CVPR52688.2022.00161 3
35. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: CVPR. pp. 10684–10695 (2022). https://doi.org/10.1109/CVPR52688.2022.01042 2
36. Schwarz, K., Liao, Y., Niemeyer, M., Geiger, A.: GRAF: Generative radiance fields for 3D-aware image synthesis. In: NeurIPS. pp. 20154–20166 (2020). https://doi.org/10.48550/arXiv.2007.02442 3
37. Shoshan, A., Bhonker, N., Kviatkovsky, I., Medioni, G.: GAN-Control: Explicitly controllable GANs. In: ICCV. pp. 14083–14093 (2021). https://doi.org/10.1109/ICCV48922.2021.01382 2
38. Skorokhodov, I., Sotnikov, G., Elhoseiny, M.: Aligning latent and image spaces to connect the unconnectable. In: ICCV. pp. 14144–14153 (2021). https://doi.org/10.1109/ICCV48922.2021.01388 3
39. Straub, J., Whelan, T., Ma, L., Chen, Y., Wijmans, E., Green, S., Engel, J.J., Mur-Artal, R., Ren, C., Verma, S., Clarkson, A., Yan, M., Budge, B., Yan, Y., Pan, X., Yon, J., Zou, Y., Leon, K., Carter, N., Briales, J., Gillingham, T., Mueggler, E., Pesqueira, L., Savva, M., Batra, D., Strasdat, H.M., Nardi, R.D., Goesele, M., Lovegrove, S., Newcombe, R.: The Replica dataset: A digital replica of indoor spaces. arXiv preprint arXiv:1906.05797 (2019). https://doi.org/10.48550/arXiv.1906.05797 14
40. Tang, S., Zhang, F., Chen, J., Wang, P., Furukawa, Y.: MVDiffusion: Enabling holistic multi-view image generation with correspondence-aware diffusion. In: NeurIPS. vol. 36 (2023). https://doi.org/10.48550/arXiv.2307.01097 3
41. Umeyama, S.: Least-squares estimation of transformation parameters between two point patterns. IEEE TPAMI **13**(04), 376–380 (1991). https://doi.org/10.1109/34.88573 11
42. Vahdat, A., Kreis, K., Kautz, J.: Score-based generative modeling in latent space. In: NeurIPS (2021). https://doi.org/10.48550/arXiv.2106.05931 2
43. Wang, G., Yang, Y., Loy, C.C., Liu, Z.: StyleLight: HDR panorama generation for lighting estimation and editing. In: ECCV (2022). https://doi.org/10.1007/978-3-031-19784-0_28 3
44. Wu, S., Tang, H., Jing, X.Y., Zhao, H., Qian, J., Sebe, N., Yan, Y.: Cross-view panorama image synthesis. PR **131**(C) (2022). https://doi.org/10.1016/j.patcog.2022.108884 3
45. Zhang, Q., Song, J., Huang, X., Chen, Y., yu Liu, M.: DiffCollage: Parallel generation of large content with diffusion models. In: CVPR. pp. 10188–10198 (2023). https://doi.org/10.1109/CVPR52729.2023.00982 3, 4