



CubeGAN: Omnidirectional Image Synthesis Using Generative Adversarial Networks

C. May  and D. Aliaga 

Purdue University, USA



Figure 1: Unwrapped cube-maps produced by our network.

Abstract

We propose a framework to create projectively-correct and seam-free cube-map images using generative adversarial learning. Deep generation of cube-maps that contain the correct projection of the environment onto its faces is not straightforward as has been recognized in prior work. Our approach extends an existing framework, StyleGAN3, to produce cube-maps instead of planar images. In addition to reshaping the output, we include a cube-specific volumetric initialization component, a projective resampling component, and a modification of augmentation operations to the spherical domain. Our results demonstrate the network's generation capabilities trained on imagery from various 3D environments. Additionally, we show the power and quality of our GAN design in an inversion task, combined with navigation capabilities, to perform novel view synthesis.

CCS Concepts

• **Computing methodologies** → **Computer graphics; Rendering; Neural networks;**

1. Introduction

Recently there have been vast strides in image generation using deep generative networks. Applications include content creation, dataset production, scene understanding, neural rendering, and more. However, most works related to image-generation networks have focused on generating planar perspective images. In this work, we aim to extend such a network to produce omnidirectional images.

Previous works have addressed the different challenges and uses of omnidirectional images such as scene reconstruction, super-resolution, and novel view synthesis (see survey by [XLZC20]). However, while image synthesis GANs have progressed significantly, few works have centered on deep synthesis of true omnidirectional images. An ideal omnidirectional image is an image sampling the environment fully surrounding a viewpoint. At one design extreme, a spherical projection surface can be placed around

the viewpoint. This surface permits uniform sampling (i.e., equiangular sampling) and does not inherently have any seams or artifacts. However, while some deep spherical convolution methods have been proposed (e.g., [CGKW18; CCG18]) in general this representation is difficult to map to a raster grid, GANs, CNNs, and other image-based deep learning structures. At the other design extreme are equirectangular projections (and related projections such as Mercator and gnomonic) which define a rectangular projection surface. But, this representation yields very non-uniform sampling (especially at the "top" and "bottom" of the image) which leads to such portions often being omitted (hence not allowing for true omnidirectional viewing) (e.g., [LCC*19]). Between these two design extremes there is a continuum of multi-plane solutions which include, for example, convex icosahedral, cubical, and tetrahedral projection surfaces. Such multi-plane, or tiled, solutions, which are the inspiration for many omnidirectional camera-cluster designs (e.g., [LKK*16]) provide straightforward support for raster grids and yield improved sampling uniformity, but introduce the issue of seams between the imaging tiles. In fact, as the number of tiles increases, sampling uniformity improves but more seams are introduced. Cube-maps are one common flavor of such omnidirectional representations as they produce exactly six square tiles, reasonable sampling uniformity, and amity to raster processing.

However, deep generation of cube-maps that contain the correct projection of the environment onto its faces is not straightforward (see [EPV*19]). Some prior methods have attempted to address this issue by constraining the generation of the content at the edge of a face (of the cube-map) with that of the corresponding edge of another face or extending the constraint to a narrow region near each edge (e.g., [CCD*18]), but this still does not produce a geometrically correct projection. In the case of GANs, the result is the uncoordinated generation of omnidirectional cube-map content that yields distortions and artifacts, and thus reduces the quality of the produced image. This improper projective constraint at the boundary is not only problematic for cube-maps but for any multi-plane approximation to an omnidirectional image.

Our approach enables projectively-correct and seam-free cube-map creation using generative adversarial learning, resulting in better omnidirectional images. We explicitly embed the particular characteristics of cube-map imaging into a generative adversarial process. In particular, i) we extend the already impressive StyleGAN framework [KAL*21] to represent a continuous spherical signal in the initial layers of the network. Then, ii) we extend the architecture to include a component that uses a ray-casting based solution to correctly re-sample the extended boundary of each cube-map face to yield a projectively correct mapping of the pixels of the corresponding face. Finally, iii) we alter the data augmentation process to further encourage seam-free image generation. Collectively these components, together with the practicality of cube-maps, produce seamless omnidirectional imagery for any viewing direction.

Our results show the improved ability of our methodology. Using five datasets created from 3D environments, we compare to the base StyleGAN3 network trained on equirectangular images, conduct an ablation study of the components of our work, and demonstrate a simple novel omnidirectional view synthesis application. We also show and/or discuss comparisons to several alternative image gen-

eration networks, novel view synthesis (e.g., [XZX*21]) and neural rendering. While we demonstrate our solution using the very successful StyleGAN3, there is room for further improvement. GANs in general often suffer from blob-shaped artifacts when trained on diverse, multi-modal datasets (e.g., [SSG22] and Figure 6), but do perform well on highly structured datasets such as faces [KLA19] and with sufficient training time. Nonetheless, we anticipate our methodology can also be adapted to other generative frameworks.

Our main contributions are:

- a generative adversarial network architecture designed to synthesize omnidirectional cube-maps,
- a ray-casting based method to correctly resample the boundaries between cube-map faces during training, and
- a method for simple navigation and novel view synthesis within a generated scene.

2. Related Work

Omnidirectional images, panoramas, and 360-degree imagery are often used for environment mapping in video games, remote tourism, and virtual reality (e.g., [XLZC20]). They also have applications in atmospheric and planetary sciences, astronomy, and cartography. Recently, they have been the focus of deep learning tasks such as reconstruction, super resolution, and image generation.

One avenue of research on omnidirectional imagery is to perform scene reconstruction tasks. For example, omnidirectional video [JMK*22] and multiple depth-enhanced RGBA panoramas [LXM*20] have been used to perform scene reconstruction and synthesis. Other works have focused on reconstruction from a single panorama image, such as indoor room layouts [YWP*19; YJL*18].

Another research goal has been improving the resolution of omnidirectional imagery to bring it closer to that of standard planar perspective images. Solutions use adversarial learning (e.g., [ZZL*20]) and some consider the different sampling properties at different latitudes within the image (e.g., [DWX*21; NIA21; KKL21; You22]).

2.1. Deep Generative Models

Beginning with Goodfellow's seminal paper [GPM*14], GANs have become a powerful image generation tool with numerous subsequent papers (see survey [PYY*19]). For example, Pix2Pix [IZZE17] and CycleGAN [ZPIE17] generate impressive image-to-image translations, BigGAN [BDS19] produces impressive high-resolution content, and StyleGAN [KLA19] enables controlling the output using concepts from style transfer literature.

However, most prior GAN papers have not focused on producing projectively-correct and seam-free fully omnidirectional images, such as cube-maps. [LCC*19] extrapolate a learned coordinate manifold and produce seam-free extended images; they demonstrate cylindrical projections which produce highly varying sampling densities, but omit the problematic north and south pole of the omnidirectional image content. The more recent [LCL*22] can

generate arbitrary size images, but they do not produce omnidirectional views. [Kei20] and [XZX*21] extend a single image to be omnidirectional (e.g., equirectangular). The former does not produce good agreement at the edges of the extended image boundary (up to about 20% misalignment) and the latter uses RGBD images, assumes a typical indoor room geometry, and employs 13000 to 17600 images to train an image inpainting engine that assists the synthesis task. In particular, they re-sample the single input image to many locations and use [NNJ*19] to inpaint the re-sampled images. Then, from this collection of images, produce a NeRF-style solution [MST*21] in order to perform the image synthesis.

In contrast, our approach embeds into the GAN-based process projectively-correct and seam-free cube-map generation to produce fully omnidirectional images, improving upon StyleGAN3, for example. Moreover, our approach can be used to assist with the inpainting component of novel view synthesis by generating plausible scene content. Our results show quantitatively and qualitatively similar image quality to StyleGAN3, but without exhibiting seams or polar distortions.

Recently, diffusion models have shown promising results in both conditional and unconditional image generation settings [RDN*22; RBL*22], in some cases surpassing GANs in terms of perceptual quality [DN21]. Unlike GANs, these networks synthesize images by repeated denoising operations within a Markov chain. To our knowledge, none of these works focus specifically on omnidirectional images.

Text2Light [CWL22] is another recent work that synthesizes high dynamic range (HDR) panoramas from text descriptions, making use of CLIP embeddings [RKH*21] and spherical positional encoding. Despite high visual and semantic quality of their generated panoramas, we show in Figure 6 that the images are not true equirectangular projections, and suffer from seams and polar distortions. Our method explicitly addresses both shortcomings.

3. CubeGAN

We describe our cube-map generation network based on StyleGAN3, followed by our improvements: initialization, projective-resampling, and augmentation.

3.1. GAN Framework

We define a GAN framework that builds upon the StyleGAN3 system. Briefly, the original system involves a *latent mapping module* consisting of a number of fully connected layers, followed by a *synthesis module*. The mapping module M transforms the input latent z into the mapped latent $w = M(z)$. The synthesis module S begins with a set of fixed input features y_0 . This input content is subjected to n synthesis layer applications $y_k = S_k(w, y_{k-1})$ where $k \in [1, n]$, each of which performs a modulated convolution, followed by a combined upsample, activation, and downsample operation. Afterwards, both the image dataset x and the synthetic image content y_n are subjected to an augmentations module A , producing $x' = A(x)$ and $y'_n = A(y_n)$, which are used to train the discriminator D .

We extend the system to support cube-map output. First, we modify the shape of the tensors that each layer expects and outputs

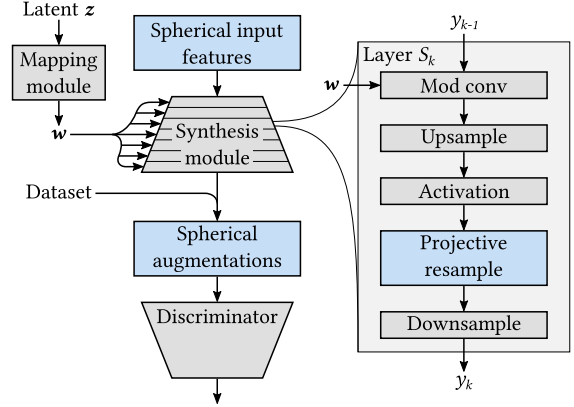


Figure 2: *CubeGAN* architecture: we show a visual summary of our cube-map generation modules added to a StyleGAN framework.

within the synthesis module. We represent a cube-map as a stack of six images, thus each image sample y_k produced by the network is a 4-dimensional tensor with shape $[6, c, h, w]$, where c is the number of channels (e.g., 3), and h and w are the height and width of a single cube face, respectively. As training occurs in batches, an additional batch dimension is prepended to the image shape, thus the network ultimately works with 5 dimensional tensors. Certain operations within the network expect 4 dimensional tensors, such as 2D convolution (i.e., the cube face index is unexpected). Prior to such operations, we reshape the tensor from 5 dimensions to 4 dimensions by interleaving the batch indices with the cube face indices. Thus, convolution is performed separately on each cube face, as though each were a separate sample in the batch.

We specifically choose the StyleGAN3-R configuration described in [KAL*21] to ensure rotational equivariance throughout the network, allowing for arbitrarily oriented cube-maps. While an equirectangular projection would allow for Y axis rotations in a translationally equivariant network, it would not allow for any other rotation axis, even under rotational equivariance, due to both the extreme disparity in sampling rates near the poles and the non-linear mapping of spherical content to the image plane. In other words, a 3D rotation of an omnidirectional image does not correspond to a 2D rotation of its equirectangular projection. A cube-map, on the other hand, is a piece-wise linear projection, and has a more uniform sampling rate across the cube faces. A 3D rotation about any axis thus more closely resembles a 2D rotation of image features about the axis intersection with the cube. Note that unlike for equirectangular projections, translational equivariance for cube-maps does not permit rotations about any axis, justifying our choice of the StyleGAN3-R configuration.

Our extensions to the StyleGAN3 framework are highlighted with blue boxes in Figure 2. We modify the *input features* to produce a projectively-correct and seamless initial image. Then, a *projective-resampling module* is executed during each synthesis layer application. Outputs from our cube-map specific *augmentations module* are used to train the discriminator. The final output is a 6-stack of images representing a cube-map.

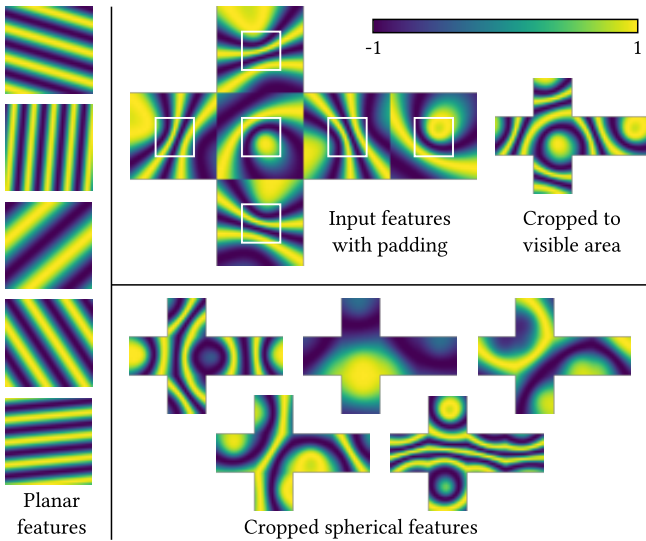


Figure 3: Input features. *Left:* examples of planar Fourier features, from StyleGAN3. *Top right:* an example of our spherical Fourier features, projected onto a cube map whose faces include padding regions. The areas within the white squares denote the visible regions of each cube face. To the right is the cropped cube map showing only the visible regions, where the signal is continuous. *Bottom right:* more examples of cropped spherical Fourier features.

3.2. Initialization

The initial image content y_0 for the synthesis module should be a valid and continuous image. Building off the StyleGAN3 architecture, the subsequent synthesis layer applications ensure translational and rotational equivariance, and prevent aliasing from propagating through the network (see [KAL*21]). As a result, the initial input becomes a continuous bandlimited signal that can be subjected to affine transformations. Such transformations are apparent in the output synthesized images. However, in the case of a cube-map, a naïve use of the image initialization scheme of StyleGAN3 (i.e., a set of planar frequencies sampled across the image, as in Figure 3, left) results in discontinuities at the seams between cube faces. Instead, the input signal must be defined to be continuous across the multiple faces of the cube-map.

We accomplish this by replacing the 2D planar frequencies with 3D volumetric frequencies. The resulting signal is sampled across the surface of the unit sphere, and then projected onto the cube faces. Additionally, the cube faces are created with padding regions in order to reduce the effects of boundary artifacts during convolution. Image content is created in these regions during synthesis, but they are ultimately cropped from the final output and will not be visible. Hence, each of the six input cube face images, including these padded areas, has an effective field of view > 90 degrees. The aforementioned signal projection accounts for this, thus the initial image content will be correctly continuous across the visible portions of the cube-map faces, as shown in Figure 3.

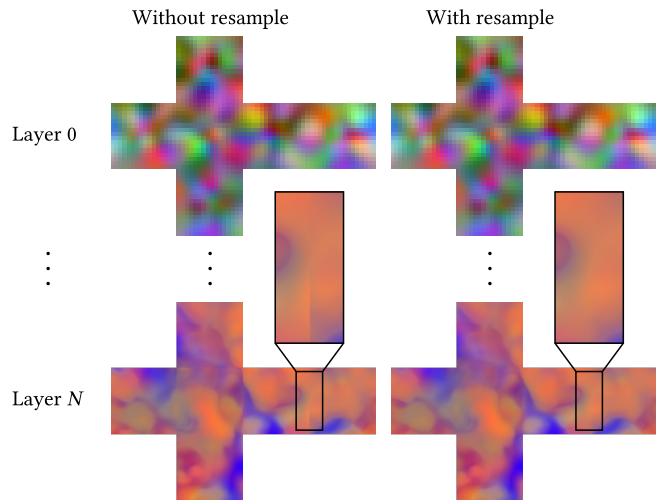


Figure 4: Effects of input features and projective resampling. The first three channels of cropped layer outputs are visualized as RGB colors. The first layer is consistent by construction, so no seams are visible. By the last layer of the network, discontinuities between adjacent faces become apparent in the absence of projective resampling.

3.3. Projective Resampling

In order to create perspective-correct and seam-free cube-maps, each cube face is padded with an additional border area. This padding enables convolution operations to sample from content beyond the visible boundaries of the cube face. The intent is to allow the synthesis module to generate content in one face that matches up to the content in an adjacent face, thus producing a seamless image transition. From the previous section, the overlapping regions of the input feature map y_0 are consistent by construction. However, successive convolutions throughout the network cause the overlapping regions to become progressively more desynchronized due to boundary effects and different projections of the overlapping content. The effects of said desynchronization are particularly apparent in the later layers of the network, producing visible seams between cube faces, as shown in Figure 4. Thus, we must perform additional operations throughout the network to enforce consistency between the cube faces.

The naïve solution of exactly replicating one face’s visible area into the padding regions of adjacent faces is not correct for non-coplanar faces. Figure 5 illustrates the problem. Padding the cube faces in this way has two consequences: the corner regions are undefined, and the content in the padded regions has an incorrect projection. In other words, straight lines appear to have a kink across the face boundary. To produce the correct behavior, we must perform a projective resampling of the padding areas. Thus, for each pixel in each padding area, a ray is cast from the origin through that pixel, and intersected with the visible regions of the other faces. The value of the padding pixel is replaced by the result of an interpolation of the four nearest pixels surrounding the ray intersection point. In our system, we assume a bilinear interpolation model. This operation has the additional benefit of redirect-

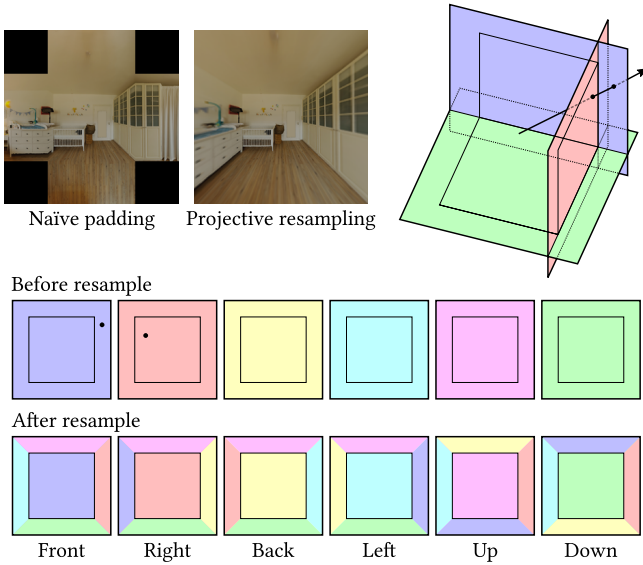


Figure 5: Projective resampling operation. **Top left:** naïve padding vs. projective resampling. Replicating adjacent face pixels into the padding regions introduces incorrect projections and undefined corners. Our resampling approach addresses both issues. **Top right:** visual depiction of ray-face intersection for resampling. For clarity, the back, left, and up faces are not shown. The intersection points are marked by circles, also shown on the individual faces below. **Bottom:** cube face contents before and after resampling the padding regions. The colors indicate from which face the pixel values originate. The inset squares outline the visible region of each cube face. After resampling, the padding regions are colored according to the neighboring faces that were sampled from.

ing the gradients through the visible regions of the cube faces during back-propagation, thereby allowing adjacent faces to influence each other.

For the StyleGAN3-R configuration, we only need to perform the projective resampling in certain places throughout the pipeline. The learned kernel for the modulated convolution operations has size 1×1 , thus this operation will not introduce any inconsistency between the overlapping regions, and we do not need to apply projective resampling afterwards. However, the upsampling and downsampling operations use filters with a larger receptive field, and as such are susceptible to desynchronization. We note however that bilinear interpolation introduces arbitrarily high frequencies, which we want to avoid for the bandlimited signal of each synthesis layer application. Therefore, we only apply projective resampling on the higher resolution image after the upsample operation. The subsequent downsample operation removes any high frequencies introduced by the resampling.

3.4. Spherical Augmentations

During training, adaptive augmentation is in general beneficial to improve image quality. The initial network employs the adaptive augmentation scheme of Karras et al. [KAH*20]. However, the ge-

ometric transformations are designed for planar images. We extend the geometric augmentations to the spherical domain by enabling anisotropic scaling in three dimensions as well as rotation about an arbitrary axis. Since the spherical image content is mapped to a cube centered at the origin, isotropic scaling has no effect, so it is disabled. Similar adaptations are made to the pixel blitting operations: 90 degree rotations about X, Y, and Z are supported, as well as mirroring, via flipping, transposing, and swapping the individual cube face images. Integer pixel translations are omitted since they would require re-sampling adjacent faces.

Dataset augmentations, as opposed to adaptive augmentations, are applied once to the dataset at the start of training to inflate the number of samples to train with. For the initial (planar) network, horizontal flips are implemented, resulting in a 2x dataset size increase. Vertical flips are not typically used when training on, e.g. a faces dataset, since the network should not learn to generate vertically flipped faces. For our network, we extend these dataset augmentations for cube maps. In addition to mirroring, we also enable 90 degree rotations, but constrained to the Y axis only, because most omnidirectional images have a clearly defined Y axis (e.g., sky vs. ground), whereas the orientation about the Y axis is typically arbitrary. Having both mirrors and rotations enabled results in an 8x dataset size increase. Note that these dataset augmentations are disabled when training on a single 3D scene, as in Section 4.4, since the orientation is fixed and learned by the network.

4. Results and Evaluation

4.1. Datasets and Training

We have trained our model on the Pano3D dataset [AZD*21], using both the Matterport3D [CDF*17] and GibsonV2 [XRH*18] splits, converted from equirectangular projections to cube maps, with a cube face resolution of 128×128 . Additionally, we trained on four static scene datasets created from 3D models obtained from Sketchfab [Ske22], specifically St. Thomas [art21], St. Giles [art19], Khayiminga Temple [Ban18], and Bedroom [fhe17]. Each scene dataset consists of 10,000 cube maps rendered at randomly selected viewpoints throughout the scene, with each cube face having a resolution of 128×128 . The orientation is fixed for all cube maps in the dataset, i.e. the front face always faces the -Z axis, etc. We used default configuration values from StyleGAN3, with the exception of the number of channels per layer, which was reduced to accommodate the increased number of pixels per sample. We also tuned the R1 regularization weight on a per-dataset basis, as recommended by the authors. The models trained at an average speed of 185 seconds per 1000 images (kimgs), on a compute cluster with a variety of A100 and A30 NVIDIA GPUs. Hand-picked perspective renderings of our outputs trained on Pano3D and Sketchfab datasets can be seen in Figures 10 and 11 respectively, and uncurated cube-maps trained on Pano3D are shown in Figure 12.

4.2. Comparisons

We evaluate our network's improvements towards omnidirectional image generation by comparing to the base StyleGAN3 network [KAL*21]. The Pano3D dataset is used for both networks, with our CubeGAN trained on cube maps, and StyleGAN3 trained on

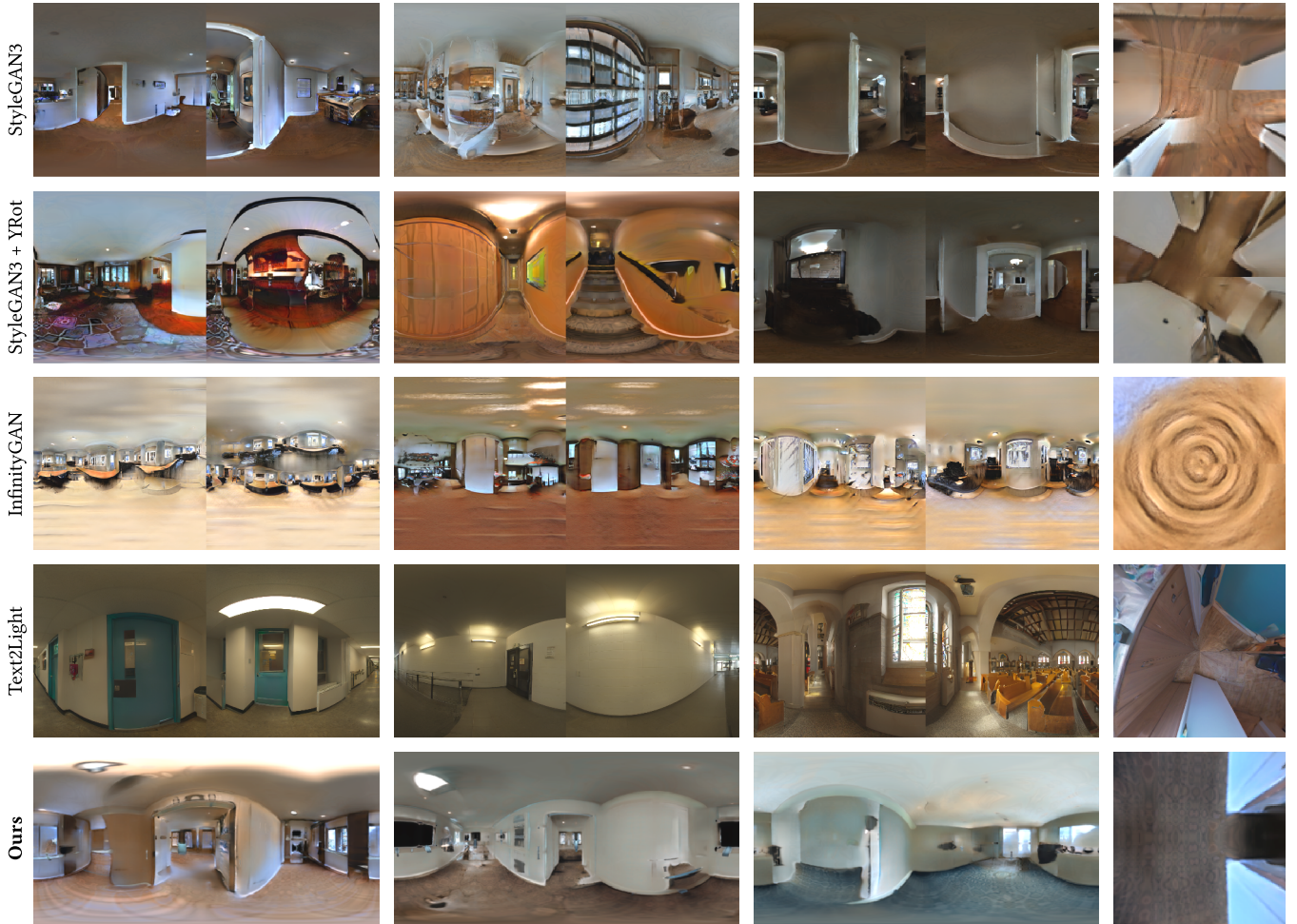


Figure 6: Example outputs from StyleGAN3, StyleGAN3 with Y-rotational augmentations, InfinityGAN trained on Pano3D, Text2Light, and our method (reprojected to equirectangular images). The left three columns are rotated by 180 degrees such that the seams are visible. The right-most column is a perspective projection facing the south pole where polar distortions are clearly visible. Our method shows neither seams nor polar distortions.

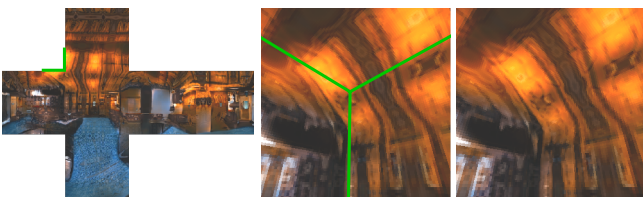


Figure 7: Example cube-map generated from our network (left), projected to a perspective viewpoint facing a corner of the cube, shown with and without a wireframe overlay. The image content is continuous across all cube face boundaries, and produces no visible seams.

Table 1: FID and seam measurements of our network compared to StyleGAN3 with and without Y-rotational augmentations. FID-180 measures the FID after horizontally circular-shifting the images by half the width.

Network	FID	FID-180	Seams
StyleGAN3	8.098	9.256	2.58×10^{-1}
StyleGAN3 + YRot	6.539	7.256	2.24×10^{-1}
CubeGAN (ours)	9.401	9.449	2.15×10^{-3}

equirectangular images. StyleGAN requires a square aspect ratio, thus we use 512×512 equirectangular images. This results in equal horizontal resolution to the 128×128 cube maps, and twice the vertical resolution. Each network is trained to 10,000 kimg, and then evaluated by measuring the Fréchet Inception Distance (FID)

[HRU*17]. To measure the FID of our cube-map-producing generator, we first reproject all faces of the cube map onto an equirectangular image, for both generated and dataset cube maps, and then measure FID as normal. Table 1 lists individual FID measurements.

An important aspect of our network is that it produces a seam-free omnidirectional image. StyleGAN however does not have any mechanism to prevent seams, and the equirectangular outputs in fact contain a vertical seam at the meridian, located at the left and right image boundary. As a result of the seam only appearing on the boundary, the FID measurement is not affected by its presence, despite it having an obvious negative impact on image quality when viewed under rectilinear projection. In general, FID is not very sensitive to the appearance of seams. Nonetheless, to help compensate for this, we also compute a variant of the FID wherein the equirectangular images are circularly shifted horizontally by half their width, corresponding to a rotation about the Y axis of 180 degrees. As a result, the seam appears in the center of the image. We apply this rotation to both generated and dataset images prior to FID measurement. As reported in Table 1, the rotated FID of our seam-free solution is very close to the standard FID (i.e., <1%), whereas it increases by 14% and 11% of the standard FID for base StyleGAN3 variants where a seam is clearly present.

We also consider the effect of Y-rotational adaptive augmentations on the seam presence. We modify the adaptive augmentations of the base StyleGAN3 network to include random horizontal circular shifts, in addition to the standard planar geometric augmentations. This network is trained to 10,000 kimgs, resuming from the base StyleGAN3 model (without Y-rotational augmentations) at 5,000 kimgs. From Table 1, we can see that the additional augmentation mildly helps the overall quality of generation, but does not completely resolve the seams as evidenced visually in Figure 6, and numerically by the gap between FID and FID-180 measurements. Our network, on the other hand, shows neither meridian seams (bottom row of Figure 6), nor seams across the cube face boundaries (Figure 7).

We further demonstrate our method’s lack of seams by directly measuring pixel differences across the image boundaries, as listed in Table 1. These values are calculated from the mean absolute difference between pixel pairs across the seams for 50k generated samples and the full dataset. Specifically,

$$s = \left| \frac{\|G_i - G_j\|_1}{n_G} - \frac{\|D_i - D_j\|_1}{n_D} \right|, \quad (1)$$

where G and D are generated and dataset images, i and j represent the sets of pixels on either side of the seam, and n_G , n_D are the total number of pixel pairs. For the equirectangular models we measure the seam at the left and right image boundaries, whereas for the cube-map model we measure 12 total seams at each cube edge. From the table, it is clear that our model results in smaller pixel differences than the base StyleGAN3 variants, indicating continuous image content.

Few other methods have attempted to generate true omnidirectional images in the GAN setting. COCO-GAN [LCC*19], as a result of conditioning on image patch coordinates, has been shown to produce seam-free cylindrical panoramas by using a horizontally cyclic coordinate system. However, this method avoids generating

Table 2: Ablation study, displaying the FID of combinations of enabled network extensions, denoted by the first three columns: IF (input features), PR (perspective resampling), and SA (spherical augmentations). The rows marked with * include a manual orientation correction before measuring FID. Recall that these models are only partially-trained (to 1000 kimgs) and thus the final FID measurements are large compared to our other trained models – moreover, an example cube-map for each configuration is shown in Figure 8.

	IF	PR	SA	FID
	N	N	N	119.26
	Y	N	N	111.47
	N	Y	N	112.58
	N	N	Y	107.18
	N	Y	Y	93.171
	Y	Y	Y	89.346
*	N	N	Y	105.13
*	N	Y	Y	82.945
*	Y	Y	Y	74.471

content at the poles by training with equirectangular images whose polar regions have been cropped out. When training their network with uncropped equirectangular images, we were unable to produce meaningful results in similar training times (e.g., several days). InfinityGAN [LCL*22] demonstrates horizontally-extendable image generation and can support cyclic panorama generation by inpainting content between provided end images, but it does not train with equirectangular images and does not attempt polar content generation. We trained InfinityGAN with the default settings on the Pano3D dataset, but without a spherical coordinate system it produces meridian seams and incoherent polar content. Recently, Text2Light [CWL22] has shown impressive results in spherical panorama generation from text input using a spherical positional encoding. However, while the generated images have the general appearance of equirectangular panoramas, they are not cyclic and often exhibit severe polar distortions. Figure 6 shows equirectangular outputs from both base StyleGAN3 variants, InfinityGAN, and Text2Light, alongside our cubemap outputs after equirectangular reprojection. All images are rotated by 180 degrees to clearly show the meridian seams. The rightmost column shows a perspective projection of the south pole, equivalently the bottom face of the cube-map, displaying polar distortions. Our cube-map generator network produces neither seams nor polar distortions.

4.3. Ablation study

We examine the effect of enabling and disabling different components of our approach. This is done by measuring the FID of models trained with several configurations of components: input features (IF), projective resampling (PR), and spherical augmentations (SA). Because of the many variations and large compute times, the networks are trained on the St. Thomas dataset for only 1000 kimgs. The resulting FID values are listed in Table 2. We note that none of the models trained without spherical augmentations converged or produced meaningful output and thus had high

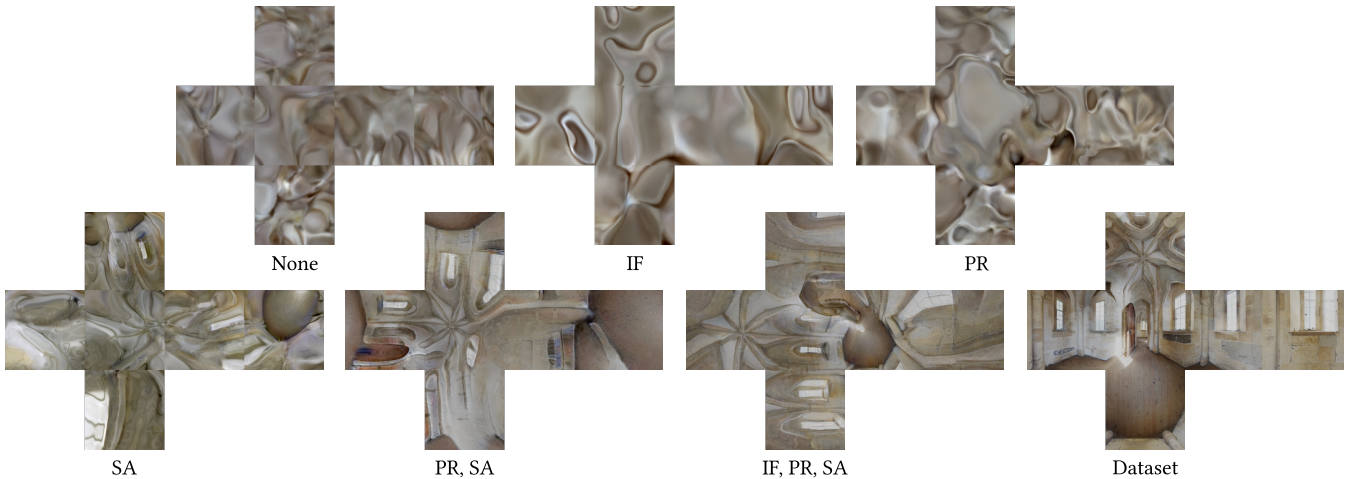


Figure 8: Example outputs from each of the network combinations in the ablation study, labeled by the enabled extensions *IF* (input features), *PR* (perspective resampling), and *SA* (spherical augmentations), without any manual orientation correction applied. The right-most image is an example from the dataset.

Table 3: FID measurements of models trained on 3D scenes, along with the training amount (in kimgs) for each.

Dataset	Training amount	FID
St. Thomas	2760	54.562
St. Giles	4320	20.767
Khayiminga	5560	25.231
Bedroom	2160	22.796

FID values. Additionally, some models learned rotated orientations with respect to the dataset, due to augmentations leaking into the generator (described in detail in [KAH*20]). The FID values are highly dependent on the learned orientation of the network. If the network produces high quality images that are rotated with respect to the dataset, the FID will be large and not representative of the image quality. Thus, for such models, we display the measurements with and without a manual rotation to correct for the orientation. We have found that further training and tuning of the γ hyperparameter usually results in the network learning the correct orientation. Example outputs from each of the trained models are shown in Figure 8.

4.4. View Synthesis

As another demonstration of the ability of our network, we use the framework to produce user-controlled novel view synthesis. In particular, we train our network on several datasets of 3D interiors, described in Section 4.1. FID measurements for each of these trained models can be found in Table 3. GANs are typically trained on diverse datasets with many different examples, but in these cases all examples in the dataset are from the same scene, thus the network learns to generate viewpoints specific to that scene. Different input latent codes cause the generator to produce different views of the

same interior. As a result, the latent space can serve as a proxy for the position of the viewpoint.

By interpolating through the latent space, we enable scene traversal along a path in 3D. At each viewpoint along a given path, we find an enclosing simplex of dataset images (i.e., a tetrahedron), annotated by their 3D positions. Each of these dataset images is projected into the intermediate latent space W following a GAN inversion procedure [XZY*21] such as that of [AQW19]. The projected latents are weighed by the path point's barycentric coordinates with respect to the enclosing simplex, producing an interpolated latent, which is used to synthesize the images along the path.

We compare the produced video to the ground truth rendering by measuring the per-frame similarity in terms of SSIM [WBSS04], PSNR, and LPIPS [ZIE*18]. Each metric is evaluated per cube face, and then averaged. Additionally, we perform the same measurements on a video consisting of alpha-blended dataset images, in which the weights of the blended images correspond to the barycentric coordinates of the interpolated latent. The resulting video follows the same path as the network-generated walkthrough, but each frame exhibits "ghosting" artifacts due to blending viewpoints that are some distance from the desired position. Examples of such walkthroughs can be seen in the supplementary video.

We plot these metrics for walkthroughs of two different datasets in the top row of Figure 9. While the generated paths have relatively smooth plots, the dataset interpolated paths exhibit many spikes. This is due to the proximity of some dataset points along the path, which dominate the alpha-blended frames, thus increasing similarity to the ground truth. The Bedroom dataset is very dense relative to St. Thomas, so we restrict the allowed dataset points to a random subset of 150 points. In the bottom row, we measure the windowed variance of the St. Thomas plots using a window size of 10 frames. The dataset interpolations exhibit much higher variation in quality, whereas the generated walkthroughs are much more consistent.

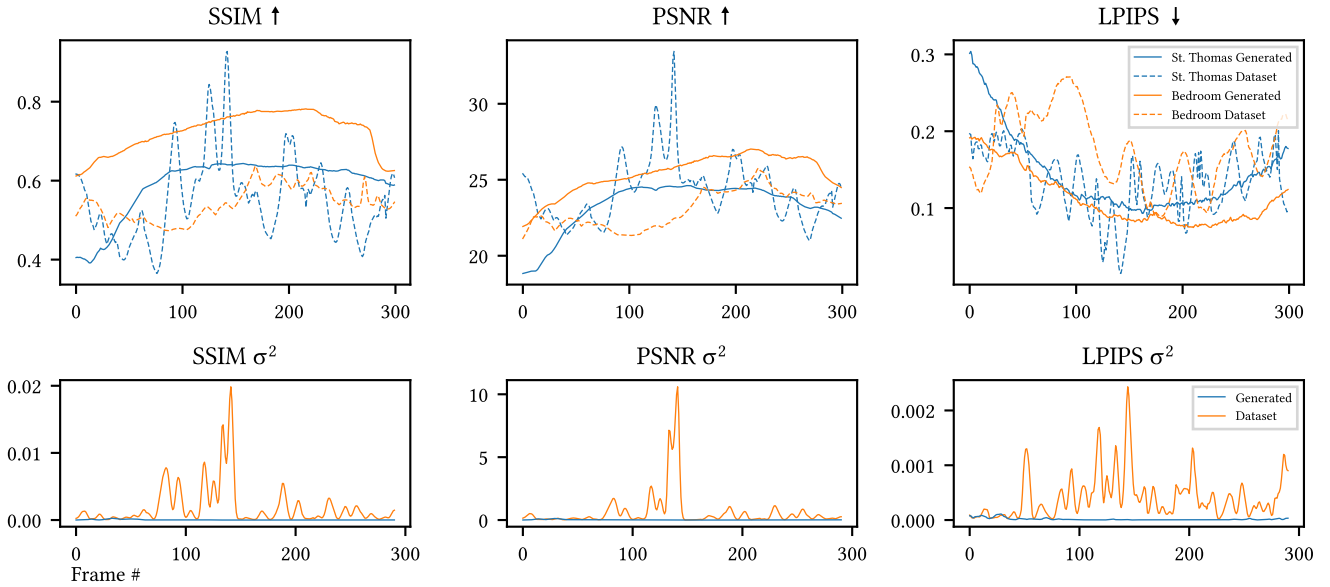


Figure 9: Scene navigation. **Top row:** per-frame evaluation of a GAN-generated walkthrough of the St. Thomas and Bedroom scenes, compared to the ground truth rendering, versus interpolation of nearby dataset images. For the Bedroom scene, 150 randomly chosen dataset points (out of 10,000) were used to interpolate projected latent codes along the traversed path. The St. Thomas interpolation uses all 10,000 points in the dataset. **Bottom row:** windowed variance of the St. Thomas walkthrough, with a window size of 10 frames.

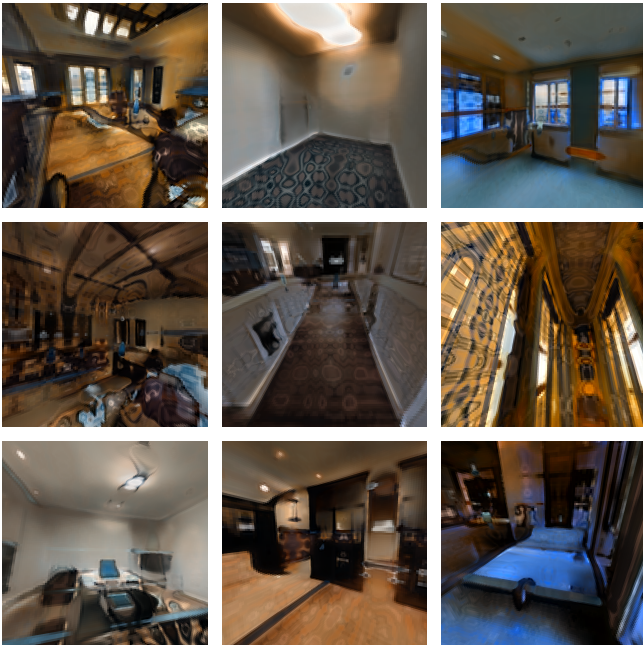


Figure 10: Example perspective views of generated cube-maps from the Pano3D dataset.



Figure 11: Example perspective views of generated cube-maps from the four Sketchfab datasets.

[XZX*21] performs novel view synthesis in similar environments. While they use one input image with RGBD data, they also use image inpainting to complete the reprojections of the provided input image to nearby locations. Said image completion makes use of a dense sampling of images (13000 to 17600 images). The dense sampling they employ is of similar spacing to three of our datasets (i.e., Bedroom has a finer sampling). The PSNR and LPIPS ranges they report are very similar to ours in Figure 9, while their SSIM range is slightly better than ours. Thus, our generative method produces almost similar image quality but without the depth data requirement and with slightly fewer images.

5. Conclusions

We have presented a method to generate projectively-correct and seam-free cube-maps using GANs. Our approach includes continuous spherical input features, a projective re-sampling process, and spherical augmentations. We build upon the StyleGAN3 framework, though our contributions can likely be applied to other generative models as well. Collectively, our method produces an improved generation process for omnidirectional content, which results in seamless imagery without polar distortions. We test with several datasets, perform an ablation study, and report other analyses including comparisons to StyleGAN3, InfinityGAN, and Text2Light. In addition, our generative ability enables us to assist with novel view synthesis.

With regards to future work, there are several directions. Image quality could be further improved by exploring additional improvements and training as well as other generative frameworks. Another issue is that there is no geometric knowledge involved in the generation process. Given that the domain of environment cube-maps is large and diverse, and generally under-sampled, it is possible that additional geometric information may assist in the generation process. We intend to explore incorporating NeRF-like [MST*20] or other geometric constraints into our network.

This work is supported in part by funds from the US National Science Foundation (NSF) Grant #1835739, US NSF Grant #1816514, US NSF Grant #2106717, and US NSF #2032770.

References

- [AQW19] ABDAL, RAMEEN, QIN, YIPENG, and WONKA, PETER. “Image2stylegan: How to embed images into the stylegan latent space?”. *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, 4432–4441 8.
- [art19] ARTFLETCH. *St Giles Cripplegate*. <https://sketchfab.com/3d-models/st-giles-cripplegate-b92917ff83914adc8bc93959ba8b4399>. Last retrieved 2022-05-20. Oct. 2019 5.
- [art21] ARTFLETCH. *St Thomas's Tower Corner Chamber*. <https://sketchfab.com/3d-models/st-thomass-tower-corner-chamber-fe2d1ca21f2a4e8e941c81012e2abf4b>. Last retrieved 2022-05-20. Oct. 2021 5.
- [AZD*21] ALBANIS, GEORGIOS, ZIOULIS, NIKOLAOS, DRAKOULIS, PETROS, et al. “Pano3d: A holistic benchmark and a solid baseline for 360deg depth estimation”. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, 3727–3737 5.
- [Ban18] BANDERA, MIGUEL. *Bagan - Khayiminga temple interior*. <https://sketchfab.com/3d-models/bagan-khayiminga-temple-interior-4c02614c50c14b00a04367ae6b6e55ad>. Last retrieved 2022-05-20. May 2018 5.
- [BDS19] BROCK, ANDREW, DONAHUE, JEFF, and SIMONYAN, KAREN. “Large Scale GAN Training for High Fidelity Natural Image Synthesis”. *International Conference on Learning Representations*. 2019. URL: <https://openreview.net/forum?id=Blxssqj09Fm2>.
- [CCD*18] CHENG, HSIEN-TZU, CHAO, CHUN-HUNG, DONG, JIN-DONG, et al. “Cube Padding for Weakly-Supervised Saliency Prediction in 360° Videos”. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, 1420–1429. DOI: 10.1109/CVPR.2018.00154 2.
- [CCG18] COORS, BENJAMIN, CONDURACHE, ALEXANDRU PAUL, and GEIGER, ANDREAS. “SphereNet: Learning Spherical Representations for Detection and Classification in Omnidirectional Images”. *Computer Vision – ECCV 2018*. Ed. by FERRARI, VITTORIO, HEBERT, MARTIAL, SMINCHISESCU, CRISTIAN, and WEISS, YAIR. Cham: Springer International Publishing, 2018, 525–541. ISBN: 978-3-030-01240-3 2.
- [CDF*17] CHANG, ANGEL, DAI, ANGELA, FUNKHOUSER, THOMAS, et al. “Matterport3D: Learning from RGB-D Data in Indoor Environments”. *International Conference on 3D Vision (3DV)* (2017) 5.
- [CGKW18] COHEN, TACO S., GEIGER, MARIO, KÖHLER, JONAS, and WELLING, MAX. “Spherical CNNs”. *International Conference on Learning Representations*. 2018, 1–15 2.
- [CWL22] CHEN, ZHAOXI, WANG, GUANGCONG, and LIU, ZIWEI. “Text2Light: Zero-Shot Text-Driven HDR Panorama Generation”. *ACM Transactions on Graphics (TOG)* 41.6 (2022), 1–16 3, 7.
- [DN21] DHARIWAL, PRAFULLA and NICHOL, ALEXANDER. “Diffusion models beat gans on image synthesis”. *Advances in Neural Information Processing Systems* 34 (2021), 8780–8794 3.
- [DWX*21] DENG, XIN, WANG, HAO, XU, MAI, et al. “LAU-Net: Latitude Adaptive Upscaling Network for Omnidirectional Image Super-Resolution”. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2021, 9189–9198 2.
- [EPV*19] EDER, MARC, PRICE, TRUE, VU, THANH, et al. “Mapped Convolutions”. *CoRR* abs/1906.11096 (2019). arXiv: 1906.11096. URL: <http://arxiv.org/abs/1906.11096> 2.
- [fhe17] FHERNAND. *Bedroom*. <https://sketchfab.com/3d-models/bedroom-869e6ec859a84240b9a099ae829f47fa>. Last retrieved 2022-05-20. Aug. 2017 5.
- [GPM*14] GOODFELLOW, IAN J., POUGET-ABADIE, JEAN, MIRZA, MEHDI, et al. “Generative Adversarial Nets”. *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*. NIPS'14. Montreal, Canada: MIT Press, 2014, 2672–2680 2.
- [HRU*17] HEUSEL, MARTIN, RAMSAUER, HUBERT, UNTERTHINER, THOMAS, et al. “Gans trained by a two time-scale update rule converge to a local nash equilibrium”. *Advances in neural information processing systems* 30 (2017) 7.
- [IZZ17] ISOLA, PHILLIP, ZHU, JUN-YAN, ZHOU, TINGHUI, and EFROS, ALEXEI A. “Image-to-Image Translation with Conditional Adversarial Networks”. *CVPR* (2017) 2.
- [JMK*22] JANG, HYEONJOONG, MEULEMAN, ANDRÉAS, KANG, DAHYUN, et al. “Egocentric Scene Reconstruction From an Omnidirectional Video”. *ACM Transactions on Graphics (Proc. SIGGRAPH 2022)* 41.4 (2022) 2.
- [KAH*20] KARRAS, TERO, AITTALA, MIKA, HELLSTEN, JANNE, et al. “Training generative adversarial networks with limited data”. *Advances in Neural Information Processing Systems* 33 (2020), 12104–12114 5, 8.
- [KAL*21] KARRAS, TERO, AITTALA, MIKA, LAINE, SAMULI, et al. “Alias-free generative adversarial networks”. *Advances in Neural Information Processing Systems* 34 (2021) 2–5.



Figure 12: *Uncurated full cube-maps generated by our network trained on Pano3D.*

- [Kei20] KEISUKE OKUBO, TAKAO YAMANAKA. “Omni-Directional Image Generation from Single Snapshot Image”. *IEEE International Conference on Systems, Man, and Cybernetics (SMC2020)*. 2020 3.
- [KKL21] KIM, HEE-JAE, KANG, JE-WON, and LEE, BYUNG-UK. “360° Image Reference-Based Super-Resolution Using Latitude-Aware Convolution Learned From Synthetic to Real”. *IEEE Access* 9 (2021), 155924–155935. DOI: [10.1109/ACCESS.2021.3128574](https://doi.org/10.1109/ACCESS.2021.3128574) 2.
- [KLA19] KARRAS, TERO, LAINE, SAMULI, and AILA, TIMO. “A style-based generator architecture for generative adversarial networks”. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, 4401–4410 2.
- [LCC*19] LIN, CHIEH HUBERT, CHANG, CHIA-CHE, CHEN, YU-SHENG, et al. “COCO-GAN: Generation by Parts via Conditional Coordinating”. *IEEE International Conference on Computer Vision (ICCV)*. 2019 2, 7.
- [LCL*22] LIN, CHIEH HUBERT, CHENG, YEN-CHI, LEE, HSIN-YING, et al. “InfinityGAN: Towards Infinite-Pixel Image Synthesis”. *International Conference on Learning Representations*. 2022. URL: <https://openreview.net/forum?id=ufGMqIM0a4b2>, 7.
- [LKK*16] LEE, JUNGJIN, KIM, BUMKI, KIM, KYEHYUN, et al. “Rich360: Optimized Spherical Representation from Structured Panoramic Camera Arrays”. *ACM Trans. Graph.* 35.4 (July 2016). ISSN: 0730-0301. DOI: [10.1145/2897824.2925983](https://doi.org/10.1145/2897824.2925983). URL: <https://doi.org/10.1145/2897824.2925983> 2.
- [LXM*20] LIN, KAI-EN, XU, ZEXIANG, MILDENHALL, BEN, et al. “Deep Multi Depth Panoramas for View Synthesis”. *Computer Vision – ECCV 2020*. Ed. by VEDALDI, ANDREA, BISCHOF, HORST, BROX, THOMAS, and FRAHM, JAN-MICHAEL. Cham: Springer International Publishing, 2020, 328–344. ISBN: 978-3-030-58601-0 2.
- [MST*20] MILDENHALL, BEN, SRINIVASAN, PRATUL P., TANCIK, MATTHEW, et al. “NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis”. *ECCV*. 2020 10.
- [MST*21] MILDENHALL, BEN, SRINIVASAN, PRATUL P., TANCIK, MATTHEW, et al. “NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis”. *Commun. ACM* 65.1 (Dec. 2021), 99–106. ISSN: 0001-0782. DOI: [10.1145/3503250](https://doi.org/10.1145/3503250). URL: <https://doi.org/10.1145/3503250> 3.
- [NIA21] NISHIYAMA, AKITO, IKEHATA, SATOSHI, and AIZAWA, KIYOHARU. “360° Single Image Super Resolution via Distortion-Aware Network and Distorted Perspective Images”. *2021 IEEE International Conference on Image Processing (ICIP)*. 2021, 1829–1833. DOI: [10.1109/ICIP42928.2021.9506233](https://doi.org/10.1109/ICIP42928.2021.9506233) 2.
- [NNJ*19] NAZERI, KAMYAR, NG, ERIC, JOSEPH, TONY, et al. “Edge-Connect: Structure Guided Image Inpainting using Edge Prediction”. *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*. 2019, 3265–3274. DOI: [10.1109/ICCVW.2019.004083](https://doi.org/10.1109/ICCVW.2019.004083) 3.
- [PYY*19] PAN, ZHAOQING, YU, WEIJIE, YI, XIAOKAI, et al. “Recent Progress on Generative Adversarial Networks (GANs): A Survey”. *IEEE Access* 7 (2019), 36322–36333. DOI: [10.1109/ACCESS.2019.2905015](https://doi.org/10.1109/ACCESS.2019.2905015) 2.
- [RBL*22] ROMBACH, ROBIN, BLATTMANN, ANDREAS, LORENZ, DOMINIK, et al. “High-resolution image synthesis with latent diffusion models”. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, 10684–10695 3.
- [RDN*22] RAMESH, ADITYA, DHARIWAL, PRAFULLA, NICHOL, ALEX, et al. “Hierarchical text-conditional image generation with clip latents”. *arXiv preprint arXiv:2204.06125* (2022) 3.
- [RKH*21] RADFORD, ALEC, KIM, JONG WOOK, HALLACY, CHRIS, et al. “Learning transferable visual models from natural language supervision”. *International Conference on Machine Learning*. PMLR. 2021, 8748–8763 3.
- [Ske22] SKETCHFAB, INC. *Sketchfab*. <https://sketchfab.com>. Last retrieved 2022-05-20. 2022 5.
- [SSG22] SAUER, AXEL, SCHWARZ, KATJA, and GEIGER, ANDREAS. “StyleGAN-XL: Scaling StyleGAN to Large Diverse Datasets”. Vol. abs/2201.00273. 2022. URL: <https://arxiv.org/abs/2201.00273> 2.
- [WBSS04] WANG, ZHOU, BOVIK, ALAN C, SHEIKH, HAMID R, and SIMONCELLI, EERO P. “Image quality assessment: from error visibility to structural similarity”. *IEEE transactions on image processing* 13.4 (2004), 600–612 8.
- [XLZC20] XU, MAI, LI, CHEN, ZHANG, SHANYI, and CALLET, PATRICK LE. “State-of-the-Art in 360° Video/Image Processing: Perception, Assessment and Compression”. *IEEE Journal of Selected Topics in Signal Processing* 14.1 (2020), 5–26. DOI: [10.1109/JSTSP.2020.2966864](https://doi.org/10.1109/JSTSP.2020.2966864) 1, 2.
- [XRH*18] XIA, FEI, R. ZAMIR, AMIR, HE, ZHIYANG, et al. “Gibson Env: real-world perception for embodied agents”. *Computer Vision and Pattern Recognition (CVPR), 2018 IEEE Conference on*. IEEE. 2018 5.
- [XZX*21] XU, JIALE, ZHENG, JIA, XU, YANYU, et al. “Layout-Guided Novel View Synthesis From a Single Indoor Panorama”. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021 2, 3, 10.
- [XZY*21] XIA, W., ZHANG, Y., YANG, Y., et al. “GAN Inversion: A Survey”. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 01 (June 2021), 1–17. ISSN: 1939-3539. DOI: [10.1109/TPAMI.2022.3181070](https://doi.org/10.1109/TPAMI.2022.3181070) 8.
- [YJL*18] YANG, YANG, JIN, SHI, LIU, RUIYANG, et al. “Automatic 3D Indoor Scene Modeling from Single Panorama”. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, 3926–3934. DOI: [10.1109/CVPR.2018.004132](https://doi.org/10.1109/CVPR.2018.004132) 2.
- [You22] YOUNGHO YOON Inchul Chung, KUK-JIN YOON. “SphereSR: 360-degree Image Super-Resolution with Arbitrary Projection via Continuous Spherical Image Representation”. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2022 2.
- [YWP*19] YANG, SHANG-TA, WANG, FU-EN, PENG, CHI-HAN, et al. “DuLa-Net: A Dual-Projection Network for Estimating Room Layouts From a Single RGB Panorama”. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, 3358–3367. DOI: [10.1109/CVPR.2019.003482](https://doi.org/10.1109/CVPR.2019.003482) 2.
- [ZIE*18] ZHANG, RICHARD, ISOLA, PHILLIP, EFROS, ALEXEI A, et al. “The unreasonable effectiveness of deep features as a perceptual metric”. *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, 586–595 8.
- [ZPIE17] ZHU, JUN-YAN, PARK, TAESUNG, ISOLA, PHILLIP, and EFROS, ALEXEI A. “Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks”. *Computer Vision (ICCV), 2017 IEEE International Conference on*. 2017 2.
- [ZZL*20] ZHANG, YUPENG, ZHANG, HENGZHI, LI, DAOJING, et al. “Toward Real-world Panoramic Image Enhancement”. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2020, 2675–2684. DOI: [10.1109/CVPRW50498.2020.003222](https://doi.org/10.1109/CVPRW50498.2020.003222) 2.