Patch-based 3D Natural Scene Generation from a Single Example

Weiyu Li[1*] Xuelin Chen[2*†] Jue Wang[2] Baoquan Chen[3]
[1] Shandong University [2] Tencent AI Lab [3] Peking University

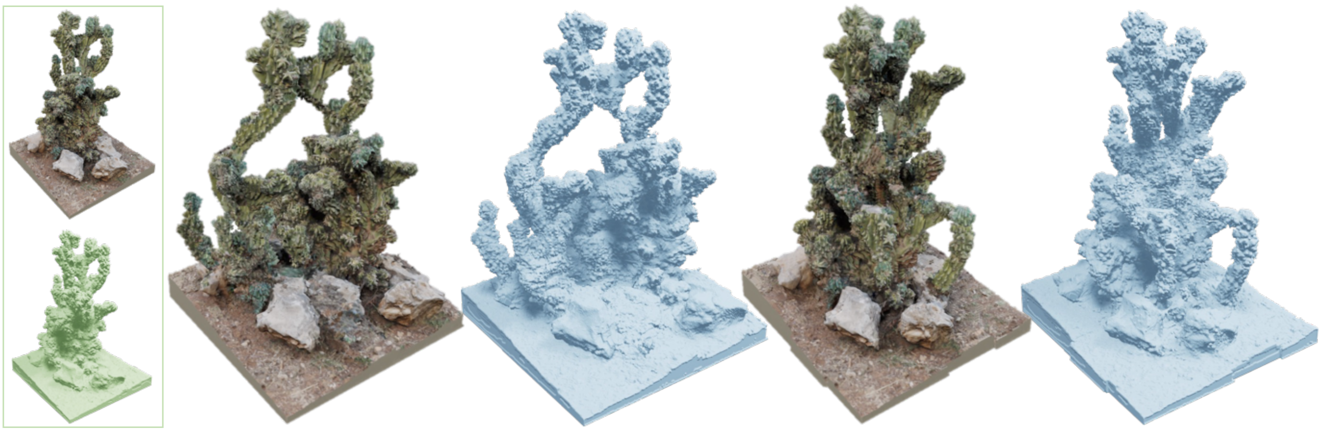

Figure 1. From a single example (inset), our method can generate diverse samples with highly realistic geometric structure and visual appearance. None of existing 3D generative models can achieve these results. More generated samples that are intricate and intriguing, ranging from single-object scenes (e.g., plants) to mid-scale ones, and large terrains, can be found in Section 4 and the supplementary.

Abstract

We target a 3D generative model for general natural scenes that are typically unique and intricate. Lacking the necessary volumes of training data, along with the difficulties of having ad hoc designs in presence of varying scene characteristics, renders existing setups intractable. Inspired by classical patch-based image models, we advocate for synthesizing 3D scenes at the patch level, given a single example. At the core of this work lies important algorithmic designs w.r.t the scene representation and generative patch nearest-neighbor module, that address unique challenges arising from lifting classical 2D patch-based framework to 3D generation. These design choices, on a collective level, contribute to a robust, effective, and efficient model that can generate high-quality general natural scenes with both realistic geometric structure and visual appearance, in large quantities and varieties, as demonstrated upon a variety of exemplar scenes. Data and code can be found at http://wyysf-98.github.io/Sin3DGen.

*Joint first authors
†Corresponding author

1. Introduction

3D scene generation generally carries the generation of both realistic geometric structure and visual appearance. A wide assortment of scenes on earth, or digital ones across the internet, exhibiting artistic characteristics and ample variations over geometry and appearance, can be easily listed. Being able to populate these intriguing scenes in the virtual universe has been a long pursuit in the community.

Research has taken several routes, among which a prevalent one is learning to extract common patterns of the geometry *or* appearance from homogeneous scene samples, such as indoor scenes [14, 25, 34, 37, 59, 63, 71, 72, 75], terrains [15, 19, 21, 26], urban scenes [12, 30, 44], etc. Another line learns to generate single objects [6, 7, 16, 25, 33, 35, 45, 74]. A dominant trend in recent has emerged that learns 3D generative models to jointly synthesize 3D structures and appearances via differentiable rendering [4, 5, 8, 18, 43, 50]. Nevertheless, all these learning setups are limited in their ability to generalize in terms of varied scene types. While a more promising direction is the exemplar-based one, where one or a few exemplars featuring the scene of interest are provided, algorithm designs tailored for certain scene types in existing methods [38–40, 73] again draw clear boundaries of scene characteristics they can handle.

This work seeks to generate *general natural* scenes, wherein the geometry and appearance of constituents are often tightly entangled and contribute jointly to unique features. This uniqueness hinders one from collecting sufficient homogeneous samples for learning common features, directing us to the exemplar-based paradigm. On the other hand, varying characteristics across different exemplar scenes restrain us from having ad hoc designs for a certain scene type (e.g., terrains). Hence, we resort to classical patch-based algorithms, which date long before the deep learning era and prevail in several image generation tasks even today [13, 17, 20]. Specifically, given an input 3D scene, we synthesize novel scenes at the patch level and particularly adopt the multi-scale generative patch-based framework introduced in [17], where the core is a *Generative Patch Nearest-Neighbor* module that maximizes the bidirectional visual summary [54] between the input and output. Nevertheless, key design questions yet remain in the *3D* generation: What representation to work with? And how to synthesize *effectively* and *efficiently*?

In this work, we exploit a grid-based radiance field – Plenoxels [69], which boasts great visual effects, for representing the input scene. While its simplicity and regular structure benefit patch-based algorithms, important designs must be adopted. Specifically, we construct the exemplar pyramid via coarse-to-fine training Plenoxels on images of the input scene, instead of trivially downsampling a pretrained high-resolution one. Furthermore, we transform the high-dimensional, unbounded, and noisy features of the Plenoxels-based exemplar at each scale into more well-defined and compact geometric and appearance features, improving the robustness and efficiency in the subsequent patch matching.

On the other end, we employ heterogeneous representations for the synthesis inside the generative nearest neighbor module. Specifically, the patch matching and blending operate in tandem at each scale to gradually synthesize an intermediate *value*-based scene, which will be eventually converted to a *coordinate*-based counterpart at the end. The benefits are several-fold: a) the transition between consecutive generation scales, where the value range of exemplar features may fluctuate, is more stable; b) the transformed features in the synthesized output is inverted to the original "renderable" Plenoxels features; so that c) the visual realism in the Plenoxels-based exemplar is preserved intactly. Last, working on voxels with patch-based algorithms necessarily leads to high computation issues. So we use an *exact-to-approximate* patch nearest-neighbor module in the pyramid, which keeps the search space under a manageable range while introducing negligible compromise on the visual summary optimality. These designs, on a collective level, essentially lay a solid foundation for an effective and efficient 3D generative model.

To our knowledge, our method is the *first* 3D generative model that can generate 3D general natural scenes from a *single* example, with *both* realistic geometry and visual appearance, in large quantities and varieties. We validate the efficacy of our method on random scene generation with an array of exemplars featuring a variety of general natural scenes, and show the superiority by comparing to baseline methods. The importance of each design choice is also validated. Extensive experiments also demonstrates the versatility of our method in several 3D modeling applications.

2. Related Work

3D Generative Models. The goal of 3D generative models is to synthesize 3D contents with realistic geometric structures and visual appearances. While procedural models are capable of mass-producing particular 3D models, they take expertise and time to obtain rules and elementary assets. Hence, automating this process has been an active area of research, resulting in a vast body of work.

A prevalent route is the learning-based one, assuming having access to sufficient homogeneous samples for training. Some learn to generate realistic 3D geometric structures, such as indoor scenes [14, 34, 37, 59, 63, 71, 72], terrains [15,19], urban scenes [12,44], etc. Others focus on the visual appearance, attempting to automatically texturize or assign materials for geometric scaffolds [21, 25, 26, 30, 75]. Another line has been directed at generating single objects with realistic structures or/and textures [6, 7, 16, 33, 35, 45, 74], showing the potential in enriching the elementary asset library. A dominant trend in recent has also emerged [4,5,8,18,43,50], where deep generative models are trained on large volumes of images collected from scenes of a specific category, to allow joint synthesis of realistic 3D structure and appearance with neural radiance fields. Nevertheless, all these learning setups require large volumes of training data, and are limited in their ability to generalize, especially in terms of varied scene types.

A more relevant direction is the exemplar-based one, where one or a few exemplars featuring the scene of interest are provided. However, existing methods with algorithm designs tailored for certain scene types again draw clear boundaries of scene characteristics they can handle. [73] extract height field patches from exemplars to synthesize terrains, but the synthesis is guided with particular emphasis on dominant visual elements in terrains. [38–40] use structured units specified in the input exemplar to facilitate architecture model synthesis. Extending texture image synthesis, [32] synthesizes signed distance fields from an input geometry, the method can not generalize to complex general natural scenes, and the result is inadequate for displaying due to the lack of appearance properties.

In this paper, we aim for 3D general natural scenes, with an emphasis on generating both realistic geometry and appearance. Lacking the necessary volume of data characterizing the target scene, along with the difficulties of having ad hoc designs in presence of varying scene characteristics, we advocate for synthesizing novel 3D scenes at the patch level, given a single exemplar scene.

Generative Image Models. Generative image models have made great strides in the past years. State-of-the-art methods can now learn generative models from large volumes of homogeneous image samples, achieving unprecedented success in producing realistic images [9, 28, 29, 31, 57, 58]. On the other end, there has also been a surge of developments to learn a generative model from a single training image [24, 51, 53, 66]. But, these learning-based single image models typically require a long training time. Differing from these learning-based paradigms, a classical patch-based approach, that dates back long before the deep learning era, is revived in [11, 17, 20], showing amazing performance. The core of these models is to maximize the bidirectional patch similarity between the input and synthesized output in a coarse-to-fine manner, and have demonstrated their capability to generate diverse outputs from a single exemplar image, with orders of magnitude faster than learning-based ones. Our work is particularly inspired by this line of work but must address challenges arising from lifting the multi-scale generative patch-based framework to *effective* and *efficient* 3D scene generation.

3D Scene Representations. While it is common to represent an image as a distributed amplitude of colors over a 2D grid, more often than not, the 3D representation varies. Polygon meshes and points offer a compact representation, with precedents in patch-based synthesis [22, 52], but the irregularity makes them intractable for high-quality 3D generation. The same holds for point clouds. Recently, the community has indeed witnessed a revolution started by an emerging representation, i.e. neural radiance field [41], which approximates the 5D plenoptic function [3] of the underlying scene with neural networks and shows unprecedentedly photo-realistic visual results. An explosion of techniques occurred since then that improves the representation in various aspects [23, 42, 47, 48, 55, 67, 68, 70]. We refer readers to [56, 64] for more in-depth summaries. Among these variants, we opt for a simple yet expressive voxel-based representation – Plenoxels [69], which has shown great competence on novel view synthesis. Its simplicity and regular structure benefit patch-based algorithms, however, important designs must be taken to fit it into our framework for high-quality generation of general natural scenes.

Concurrent Work. Concurrent works [27, 60] propose to learn a 3D generative model from images of an input scene, producing variations that can be rendered with realistic imagery. [62] focus on generating diverse geometric structures from an input shape. Their core idea is to extend 2D SinGAN [51] for learning the internal distribution in the 3D exemplar, differing significantly from our technical route. While these methods require a long training time (typically days), our method can generate high-quality samples in minutes, without offline training. Last, [46] can generate arbitrary 3D models represented by NeRF, with pretrained powerful image diffusion models as priors.

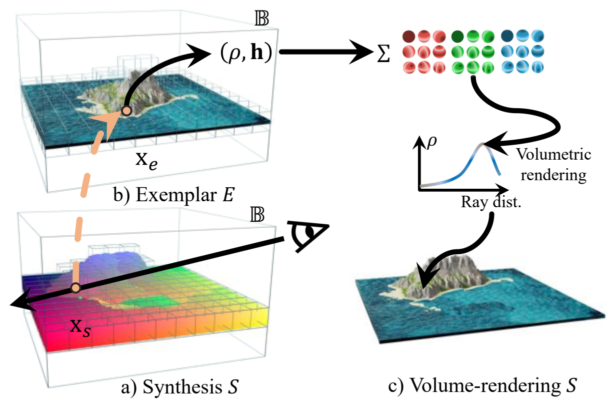

Figure 2. a) The synthesized scene S is represented as a field mapping a coordinate in S to one in E. b) The Plenoxels-based exemplar E uses a sparse grid, where each occupied voxel stores a scalar opacity ρ and spherical harmonic coefficients $\mathbf{h}$. c) Appealing imagery of S can be produced via the volumetric rendering function. Empty voxels are omitted for simplicity.

3. Method

The input 3D scene to our method can be a real-world or digital scene, as we first train Plenoxels on the images of the input scene to obtain a Plenoxels-parameterized exemplar. Then, our method synthesize novel variations at the patch level, with a multi-scale generative patch-based framework. In the following, we describe important designs w.r.t. the scene representation (Section 3.1 & 3.2) and the generative patch nearest-neighbor field module (Section 3.3), that, integrated into the multi-scale patch-based framework (Section 3.2), contribute collectively to our success.

3.1. Scene Representations

Exemplar Scene Representation. We assume the exemplar scene E lies within an axis-aligned box $\mathbb{B}$ centered at the origin, around which we can distribute cameras to capture images for training Plenoxels. As per Plenoxels, E is represented by a sparse voxel grid, where each occupied voxel center stores features including a scalar opacity ρ and a vector of spherical harmonic (SH) coefficients $\mathbf{h}$ for each color channel: $E : \mathbf{x} \rightarrow (\rho, \mathbf{h})$, where $\mathbf{x}$ indicates a voxel center within $\mathbb{B}$. These features can be further trilinearly interpolated to model the full plenoptic function continuously in space. Notably, the appearance feature uses 2-degree harmonics, which requires 9 coefficients per color channel for a total of 27 harmonic coefficients per voxel.

Exemplar Transformation. While Plenoxels features can be used to render pleasing imagery, naively using them for the patch distance is unsuitable. Density values are not well-bounded, contain outliers, and can not accurately describe the geometric structure within a patch. On the other hand, high-dimensional SH coefficients are excessively consumptive for patch-based frameworks. Hence, we transform the exemplar features for the input to the generative patch nearest neighbor module. First, the density field is converted to a signed distance field (SDF). Specifically, the signed distance at each voxel is computed against the surface mesh extracted from the density field by Marching Cubes [36]. Note that Plenoxels prunes unnecessary voxels during training, which creates holes and irregular structures in invisible regions. So we flood-fill these regions with high-density values, prior to the mesh extraction. Last, we rescale and truncate the signed distance to ignore distance values far away from the surface. Formally, the geometry transformation is as follow: $G(\mathbf{x}) = \max\Big(-1, \min\big(1, SDF(\mathbf{x})/t\big)\Big)$, where the truncated scale t is set to 3 times of the voxel size at each generation scale. Moreover, we normalize SH coefficient vectors and use the principal component analysis (PCA) to reduce the dimensionality (from 27 to 3 by default), significantly reducing the computation overhead. Finally, the transformed exemplar $\hat{E}$ is now given as:

$$\hat{E} : \mathbf{x} \rightarrow \big(G(\mathbf{x}), P(\mathbf{h})\big), \quad (1)$$

where $G(\cdot)$ denotes transforming of the geometric feature, and $P(\cdot)$ transforming the appearance feature.

Synthesized Scene Representation. In the multi-scale generation, the output scene S at each scale is represented by a coordinate-based mapping field, instead of a value-based one that stores features. Specifically, S is represented as a field that maps a 3D voxel center in the synthesis grid to one in the exemplar E, $S : \mathbf{x}_s \rightarrow \mathbf{x}_e$, with which the original Plenoxels features $E\big(S(\mathbf{x}_s)\big)$ can be queried for S.

Note, in addition to discrete grid samplings, dense samplings $\mathbf{x}_s$ in S can also be mapped to the continuous exemplar space, by simply considering the local offset δ to the nearest voxel center, i.e., $S(\mathbf{x}) = S(\mathbf{N}(\mathbf{x})) + \delta$, where $\mathbf{N}(\cdot)$ returns the nearest voxel center of $\mathbf{x}$. This is particularly useful, as it enables upscaling S to finer grids in the multi-scale framework, and sufficient sampling for rendering the final generation result with high-quality imagery.

Viewing Synthesized Results. The synthesized scene can be projected onto 2D through the volume rendering equation as in NeRF [41], yielding highly photo-realistic imagery under varying views. We refer readers to [69] for

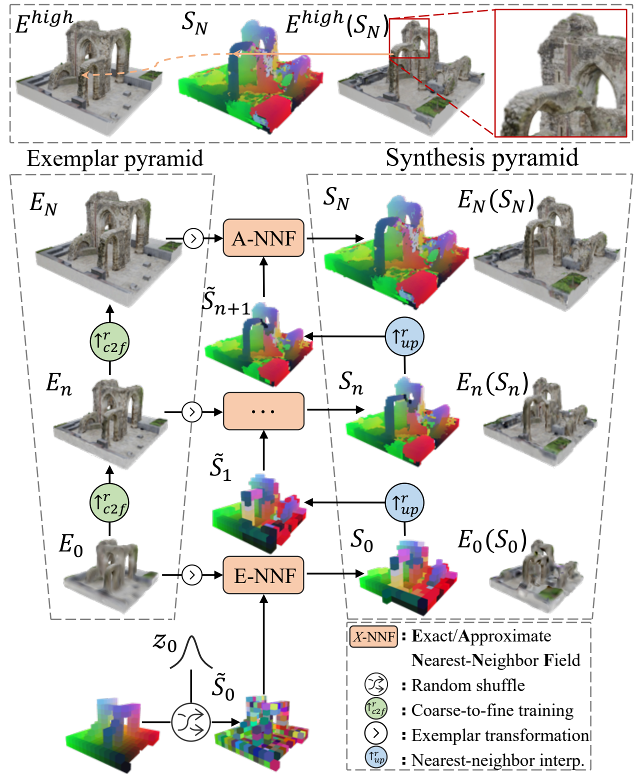

Figure 3. Multi-scale generation. At each scale n, the NNF module updates the generation based on the transformed exemplar $\hat{E}_n$ and an initial guess $\tilde{S}_n$ upscaled from the previous. The coarsest scale takes a shuffled identity mapping field as input. Note that our coordinate-based representation S_N can map to patches in a higher-resolution exemplar for higher quality (top row).

more details. Figure 2 illustrates how a synthesized result, paired with the exemplar, can display appealing imagery.

3.2. Multi-scale Generation

We use the same multi-scale framework as in previous works [10,17,51], which generally employs a coarse-to-fine process, so we have the opportunity to synthesize a more detailed scene based on an initial guess upscaled from the previous scale. In this pyramidal pipeline, different information is captured and reproduced at varying scales, spanning from global layouts at coarser scales to fine geometric and appearance details at finer scales (See Figure 3).

Exemplar Pyramid Construction. Given the input scene, we build a pyramid $(E_0, ..., E_N)$, where E_{n-1} is a downscaled version of E_n by a factor r^{-1} ($r = 4/3$). By default, we use $N = 7$ (8 scales in total) for balancing quality and efficiency. Specific resolutions in the pyramid are listed in the supplementary. When working with an exemplar pyramid obtained by recursively downsampling a pretrained high-resolution exemplar, we observed lots of

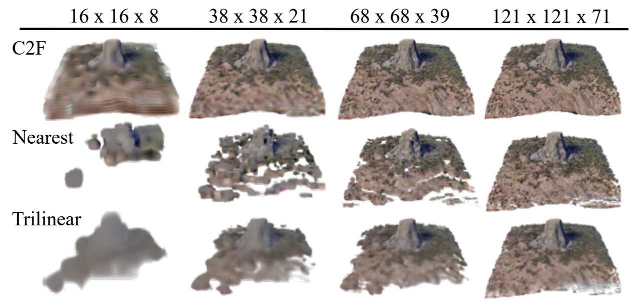

Figure 4. Exemplar pyramid. Coarse-to-fine training (top) shows more consistency between consecutive exemplars, whereas common downsampling algorithms (mid and bottom) result in missing geometry (e.g., the ground) and blurry appearance.

artifacts due to missing features at coarser exemplars, and severe feature inconsistency between exemplars at consecutive scales. Hence, we build the exemplar pyramid by coarse-to-fine training Plenoxels, at increasing resolutions synchronized with the multi-scale framework. Such exemplar pyramid prevents losing thin structures at coarser scales, and offers rather smoother transition and consistent features between consecutive exemplars, leading to stable transition in the multi-scale generation (See Figure 4).

Coarse-to-fine Generation. At each scale n, an initial guess $\tilde{S}_n$ is produced by upsampling the output in the previous scale: $\tilde{S}_n = S_{n-1} \uparrow^r$, with the same factor r to match with the exemplar. Then, the mapping field in $\tilde{S}_n$ is updated by the generative nearest neighbor module with matched coordinates in the exemplar. The patch size at all scales is p^3 ($p = 5$ by default), which captures around 1/3 of the content in the coarsest exemplar. Unlike adding noise to raw exemplar values as in the image synthesis, our initial guess $\tilde{S}_0$ at the coarsest scale is an identity mapping field shuffled with Gaussian noise $z_0 = \mathcal{N}(0, \sigma^2)$, $\sigma = 0.5$ by default, scaled by the extents of the bounding box $\mathbb{B}$, which is natural for our coordinate-based synthesis.

3.3. Generative Patch Nearest-Neighbor Field

Usually, two stages, namely the patch *matching* and *blending*, operate in tandem in the nearest neighbor field (NNF) search of patch-based algorithms. Specifically, the matching finds the most suitable patch from the exemplar for each in S, and then the latter blends of the *values* of overlapping patches together. This is vital to a robust EM-like optimization in patch-based image synthesis [1,2], leading to converging synthesis results in several iterations.

Value-/Coordinate-based Synthesis. In this work, we use heterogeneous representations for the synthesis in NNF. Specifically, at each scale n, the patch matching and blending first operate in tandem for $T-1$ iterations, to gradually synthesize an intermediate *value*-based scene with averaged values over overlapping patches. Then, when the synthesis is stable at the last iteration, the final output of NNF uses coordinate-based representation, which stores only the center location of the nearest patch in $\hat{E}_n$. As aforementioned, this design offers stable transition between consecutive generation scales, where the value range of exemplar features may fluctuate, and, importantly, helps us trace back to the original Plenoxels features that can be rendered into photo-realistic imagery, via simply mapping to the original exemplar, even to a higher-resolution version for the final generated scene (See top of Figure 3). Specifically, each iteration in NNF at each scale proceeds as follows:

(1) *Extract Patches*: Patches in $\hat{E}_n(\tilde{S}_n)$ are extracted to form a query patch set Q, and ones in $\hat{E}_n$ form a key set K.

(2) *Match Nearest Neighbors*: We first compute distance between each query patch Q_i and each key patch K_j as the weighted sum of the appearance and geometric features using $L2$ distance:

$$D_{i,j} = w_a||Q^a_{i,j} - K^a_{i,j}||^2 + (1 - w_a)||Q^g_{i,j} - K^g_{i,j}||^2, \tag{2}$$

where w_a (0.5 by default) is the trade-off parameter. To control the visual completeness in the synthesis by the bidirectional similarity [54], the final patch similarity scores normalize the distance with a per-key factor:

$$C_{i,j} = \frac{D_{i,j}}{(\alpha + \min_l(D_{l,j}))}, \tag{3}$$

where α (0.01 by default) controls the degree of completeness, and smaller α encourages completeness.

(3) *Update* $\tilde{S}_n$: For each query patch Q_i in $\hat{E}_n(\tilde{S}_n)$, we find its nearest patch in K_l, then update $\tilde{S}_n$ with averaged values over overlapping patches for the first $T-1$ iterations, and with the nearest patch center for the last iteration.

Exact-to-Approximate NNF. Although the computation above can be in parallel performed on GPUs, brutally enumerating all pairs of patches would apparently lead to surprisingly huge distance matrices as the resolution increases, preventing us from obtaining high-resolution synthesis even with modern powerful GPUs. Hence, to avoid searching in tremendous space, we propose to perform the NNF in an *exact-to-approximate* manner. Specifically, at first 5 coarser scales, *exact nearest-neighbor field* (E-NNF) search is performed with $T_e = 10$ times to stabilize global layout synthesis when the memory consumption is low. At rest 3 finer scales, an *approximate nearest-neighbor field* (A-NNF) search – PatchMatch [1] – with jump flood [49] is used for $T_a = 2$ times to reduce memory footprint from $O(M^2)$ to $O(M)$ (M is the number patches), which is equivalent to only considering visual coherence.

Figure 5. Random generation. Our method generalizes to all these scenes with highly varied structures and appearances, producing highly diverse and realistic scenes. The supplementary presents exemplars and more artistic imagery rendered with these 3D scenes.

4. Experiments

We collected a rich variety of 3D scene models to examine the performance of our method on random scene generation, ranging from rocks to plants, sculptures, landscapes, terrains, artistic scenes, etc. Some are digitalized *real-world* scenes, e.g., the *Devil's Tower*. These scenes possess varying degrees of complexity in terms of geometry and appearance. In the following, we present experiments conducted to evaluate various aspects of the proposed solution. Unless specified, we use the default parameters described above, 512 for the resolution along the max dimension of the E^{high}, and 512×512 image resolution for rendering. Full visualization of all exemplars, more technical details and experimental results can be found in the supplementary.

Random Generation. Figure 5 presents results obtained by our method on exemplar-based random scene generation. These results show our method can generalize to scenes of highly varied features, yielding high-quality and diverse scenes similar to the exemplar. A particular feature of our method is the photo-realism and view-dependent effects of the exemplar are inherited in the results, as evidenced by Figure 5 and 7. Each sample is generated in 1$\sim$3 minutes on a V100 GPU depending on the scenes, and viewing the results can be executed at an interactive rate (15 fps).

Comparisons. We particularly compare to GRAF and StyleNeRF, which are representative GAN-based 3D generative models. We cast them into exemplar-based models via training separately on images of each exemplar. In addition, we also compare to GPNN-3D, which trivially extends [17] for our task. We investigate the advantages of exemplar-based scene generation using our method against these alternatives, on various exemplars listed in Table 1. Figure 8 presents part of their visual results. Generally, GAN-base baselines suffer from notorious mode collapse, producing almost identical results due to lacking diverse training scenes. The visuals also tend to be more blurry and noisy, compared to our sharp imagery. GPNN-3D can not synthesize high-resolution results due to computational efficiency issues, and quickly fails at coarse scales, producing meaningless content. For quantitative comparisons, we produce 50 generated scenes from each exemplar with each method, render multi-view images and extract 3D surface points of the exemplar and of each generated scene, and then rate the *Visual Quality* (V-Qua.) *Visual Diversity* (V-Div.), *Geometry Quality* (G-Qua.), and *Geometry Diversity* (G-Div.) using common metrics employed in both 2D [51] and 3D [61] generation. The supplementary contains more details. Table 1 presents quantitative results, where, by rating with the combination of these established metrics, ours outperforms baselines by large margins, suggesting high quality and diversity from both 2D and 3D perspective.

Ablation. We compare to several variants derived from our full method: 1) *Ours (w/o TSDF)* uses an occupancy

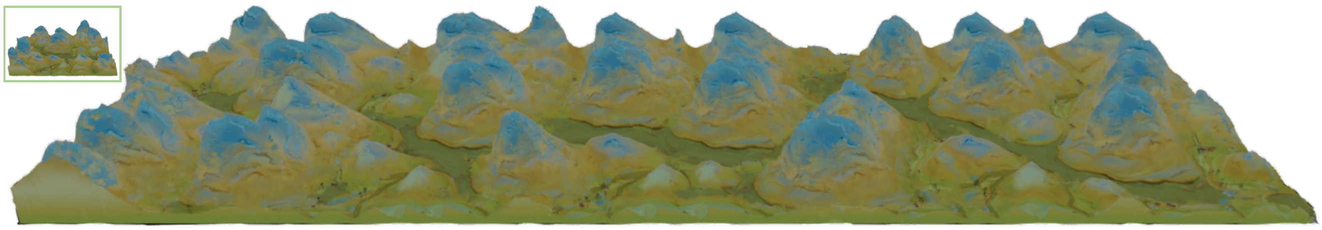

Figure 6. A novel *"A Thousand Li of Rivers and Mountains"* [65] is rendered from a generated 3D sample, that is of a different size, resolution and aspect ratio to the *Vast Land* exemplar (inset). Specification: E_N - $288 \times 288 \times 112$, E^{high} - $512 \times 512 \times 200$, S_N - $747 \times 288 \times 112$, $E^{high}(S_N)$ - $1328 \times 512 \times 200$, final rendering resolution - 4096×1024.

Table 1. Quantitative comparisons. Ours outperforms baselines by large margins, with high quality and diversity scores in terms of both visual and geometric content. We highlight top two in bold and underline the top one. Note GPNN-3D's high diversity scores can be explained by noisy contents shown in the visual results.

	GRAF				StyleNeRF				GPNN-3D				Ours			
	V-Qua.↓	V-Div.↑	G-Qua.↓	G-Div.↑	V-Qua.↓	V-Div.↑	G-Qua.↓	G-Div.↑	V-Qua.↓	V-Div.↑	G-Qua.↓	G-Div.↑	V-Qua.↓	V-Div.↑	G-Qua.↓	G-Div.↑
St Alphage	**0.078**	**0.046**	**0.473**	0.040	0.206	0.032	0.769	0.012	3.929	0.041	134.997	0.059	**0.022**	**0.312**	**0.612**	**0.473**
Devil's Tower	**0.233**	0.084	**0.480**	0.021	0.470	0.021	1.000	0.011	2.495	**0.694**	6.545	**1.974**	**0.032**	**0.203**	**0.304**	**0.207**
Desert Lowpoly	**0.057**	0.048	0.721	0.043	0.255	0.026	0.813	0.018	1.312	**0.405**	**0.344**	**1.048**	**0.020**	**0.312**	**0.568**	**0.454**
Green Island	**0.294**	0.047	**0.277**	0.015	0.606	0.015	0.669	0.014	0.254	**1.136**	18.228	**17.673**	**0.044**	**0.172**	**0.097**	**0.081**
Stone Arch	0.101	0.055	0.603	0.029	**0.060**	0.011	**0.339**	0.005	1.608	**0.504**	53.943	**29.448**	**0.003**	**0.146**	**0.126**	**0.100**
Mountain	0.410	0.060	**0.498**	0.022	**0.222**	0.037	**0.757**	0.010	2.602	**0.787**	5.947	**1.674**	**0.105**	**0.391**	0.935	**0.467**
Vast Land	**0.072**	0.058	0.994	0.229	0.219	0.023	1.047	0.017	0.907	**0.348**	**0.690**	**1.840**	**0.014**	**0.124**	**0.177**	**0.105**

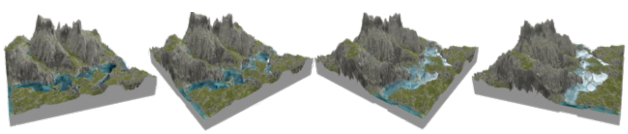

Figure 7. View-dependent effects in our synthesized results. See the reflection on the river changing under spinning cameras.

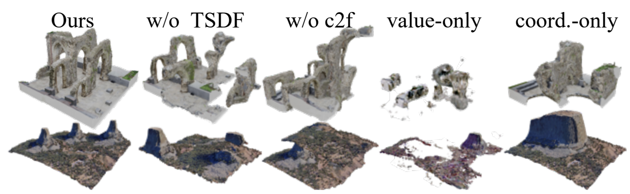

Figure 9. Ablation study. Ours (w/o TSDF) and (w/o c2f) can not well preserve the geometric structures. Ours (value-only) fails and produces with noisy content, while Ours (coord.-only) is unstable, easily leading to bulky structures or holes.

field, instead of TSDF, converted from the exemplar density field for geometric features; 2) *Ours (w/o c2f)* drops the deep coarse-to-fine exemplar training, and instead recursively trilinearly interpolates a high-resolution exemplar; 3) *Ours (value-only)* uses only value-based synthesis in NNF, and does not use TSDF and PCA as we can not trace back to original Plenxels features, and the maximum resolution is limited to 68; 4) *Ours (coord.-only)* uses only coordinate-based synthesis in NNF. Figure 9 and Table 2 present the qualitative and quantitative comparison results, respectively, showing the importance of each algorithmic design.

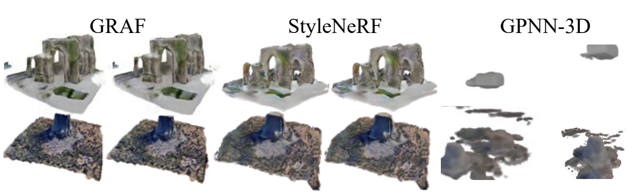

Figure 8. Visual comparisons. GAN-based baselines suffer from severe mode collapse, producing samples (two shown) almost identical to the input. GPNN-3D fails on the task.

Higher-resolution Generation. 1) In Figure 6, we show that our method supports generating a result scene of different size to the exemplar, and particularly of a much higher resolution and different aspect ratio. See specifications in the caption. 2) In addition, we also stress test with a very high-resolution setting, where E_N has 288 voxels along the max dimension, and our method can still synthesize a highly plausible sample in $\sim$10 minutes. We observed slightly improved visual quality over the default setting, as the default is sufficient for most complicated scenes. Results and details can be found in the supplementary.

Applications. In Figure 10, we demonstrate the versatility of our method in several 3D modeling applications with our unified generation framework (more details in the supplementary): 1) *Retargeting:* The goal is to resize a 3D scene

Table 2. Quantitative ablation results. While some variants produce higher diversity scores with meaningless noisy contents, ours consistently produce *diverse* results with *highest* quality scores.

		V-Qua.↓	V-Div.↑	G-Qua.↓	G-Div.↑
St Alphage	Ours	**0.022**	0.312	**0.612**	0.473
	w/o TSDF	0.114	**0.568**	1.176	**1.105**
	w/o c2f	**0.024**	**0.353**	**0.847**	0.639
	value-only	3.779	0.054	56.304	**3.823**
	coord.-only	0.044	0.336	1.003	0.719
Devil's Tower	Ours	**0.032**	0.203	**0.304**	0.207
	w/o TSDF	0.047	0.263	0.422	0.350
	w/o c2f	0.082	**0.547**	2.101	**3.500**
	value-only	1.795	**0.344**	14.122	**7.650**
	coord.-only	**0.041**	0.201	**0.256**	0.492
Desert Lowpoly	Ours	**0.020**	0.312	**0.568**	0.454
	w/o TSDF	**0.041**	**0.462**	**1.347**	1.007
	w/o c2f	0.049	0.457	1.745	1.097
	value-only	0.763	0.419	29.047	**6.674**
	coord.-only	0.100	**0.487**	2.754	**1.526**

to a target size (typically of a different aspect ratio), while maintaining the local patches in the exemplar. We simply change the size of the identity mapping field and use it as the initial guess $\tilde{S}_0$ without shuffling. 2) *Editing:* Users can manipulate on a 3D proxy, which can be the underlining mapping field or mesh, for editing an exemplar or generated scene, such as removal, duplication, and modification. The manually manipulated proxy is then converted and fed as the initial guess at the coarsest scale for synthesizing the final scene. 3) *Structural analogies:* Given two scenes A and B, we create a scene with the patch distribution of A, but which is structurally aligned with B. This is realized by using the exemplar pyramid of A, and an identity mapping as the initial guess, but by replacing $\hat{E}_0(\tilde{S}_0)$ with the transformed features in B, and vice versa. 4) *Re-decoration:* With the coordinate-based representation, we can re-decorate the generated ones with ease, via simply remapping to exemplars of different appearance.

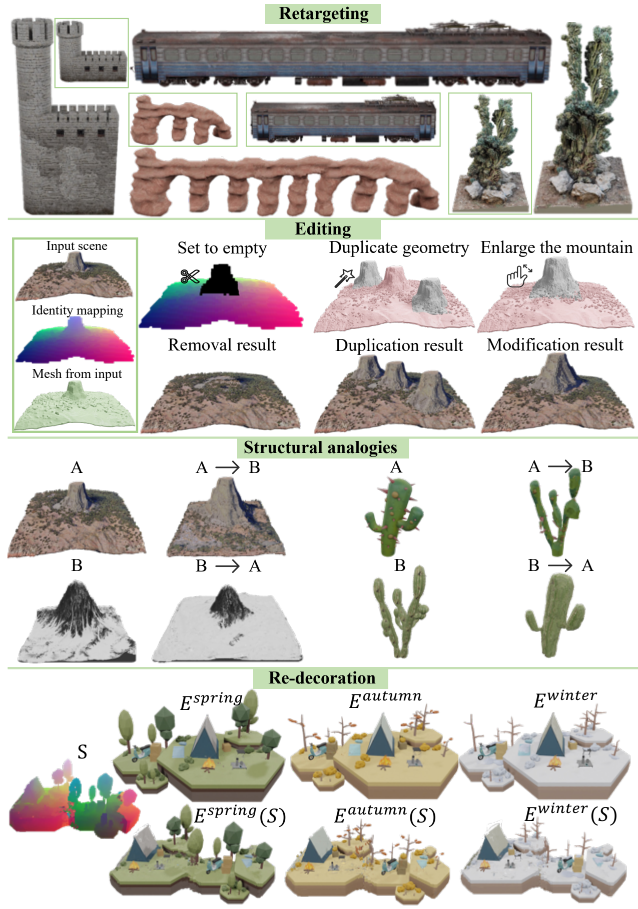

Figure 10. Applications. 1st: Retargeting 3D scenes (marked in boxes). 2nd: Editing a 3D scene (removal, duplication and modification). 3rd: Structural analogies. $A \rightarrow B$ = Visual content of A + Structure of B, and *vice versa*. 4th: Re-decoration is realized by simply re-mapping to exemplars of different appearance.

5. Discussion, Limitations and Future Work

This work makes an first attempt towards a generic generative model for synthesizing highly realistic general natural scenes from only one exemplar. Building upon Plenoxels, our method can efficiently synthesize diverse and high-quality scenes. The generated samples particulary inherit photo-realism and view-dependent effects from the example. Despite success demonstrated, we note a few shortcomings. We can not handle scenes eluding Plenoxels (e.g., transparent fluids, strong reflection), which is the actual input to our framework. Particularly, the Plenoxels-based representation is not suitable for large and unbounded scenes, leading to artifacts in the results (more discussion in supplementary). With voxelized volumetric representations, we can not perfectly synthesize scenes with tiny thin structures, and ones with highly semantic or structural information, e.g., human body and modern buildings. Moreover, in contrast to *continuous* distributions learned in neural-based methods, we work on *discrete* patch distributions and thus lack the capability of generating novel patches/pixels. A future direction is to learn a continuous distribution from a large number of homogeneous samples produced by our method, with GANs, VQ-VAEs, or diffusion models. Last, the view-dependent effects of the results are inherited from the input Plenoxels, although SH features have already *implicitly* considered the veiw-dependent lighting, consistent global illumination can not be guaranteed in our results, leading to another future direction.

Acknowledgements. This work was supported in part by National Key R&D Program of China 2022ZD0160801.

References

[1] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM Transactions on Graphics (TOG)*, 28(3):24, 2009. 5

[2] Connelly Barnes and Fang-Lue Zhang. A survey of the state-of-the-art in patch-based synthesis. *Computational Visual Media*, 3(1):3–20, 2017. 5

[3] James R Bergen and Edward H Adelson. The plenoptic function and the elements of early vision. *Computational models of visual processing*, 1:8, 1991. 3

[4] Eric Chan, Marco Monteiro, Petr Kellnhofer, Jiajun Wu, and Gordon Wetzstein. pi-gan: Periodic implicit generative adversarial networks for 3d-aware image synthesis. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 1, 2

[5] Eric R Chan, Connor Z Lin, Matthew A Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas Guibas, Jonathan Tremblay, Sameh Khamis, et al. Efficient geometry-aware 3d generative adversarial networks. *arXiv preprint arXiv:2112.07945*, 2021. 1, 2

[6] Siddhartha Chaudhuri, Evangelos Kalogerakis, Leonidas Guibas, and Vladlen Koltun. Probabilistic reasoning for assembly-based 3d modeling. In *SIGGRAPH*, pages 1–10. 2011. 1, 2

[7] Xuelin Chen, Daniel Cohen-Or, Baoquan Chen, and Niloy J Mitra. Towards a neural graphics pipeline for controllable image generation. In *Computer Graphics Forum (Eurographics)*, volume 40, pages 127–140. Wiley Online Library, 2021. 1, 2

[8] Terrance DeVries, Miguel Angel Bautista, Nitish Srivastava, Graham W Taylor, and Joshua M Susskind. Unconstrained scene generation with locally conditioned radiance fields. In *International Conference on Computer Vision (ICCV)*, pages 14304–14313, 2021. 1, 2

[9] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021. 3

[10] Alexei A Efros and William T Freeman. Image quilting for texture synthesis and transfer. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, pages 341–346, 2001. 4

[11] Ariel Elnekave and Yair Weiss. Generating natural images with direct patch distributions matching. In *European Conference on Computer Vision (ECCV)*, 2022. 3

[12] Tian Feng, Lap-Fai Yu, Sai-Kit Yeung, KangKang Yin, and Kun Zhou. Crowd-driven mid-scale layout design. *ACM Transactions on Graphics (TOG)*, 35(4):132–1, 2016. 1, 2

[13] Noa Fish, Lilach Perry, Amit Bermano, and Daniel Cohen-Or. Sketchpatch: Sketch stylization via seamless patch-level synthesis. *ACM Transactions on Graphics (TOG)*, 39(6):1–14, 2020. 2

[14] Qiang Fu, Xiaowu Chen, Xiaotian Wang, Sijia Wen, Bin Zhou, and Hongbo Fu. Adaptive synthesis of indoor scenes via activity-associated object relation graphs. *ACM Transactions on Graphics (TOG)*, 36(6):1–13, 2017. 1, 2

[15] Eric Galin, Eric Guérin, Adrien Peytavie, Guillaume Cordonnier, Marie-Paule Cani, Bedrich Benes, and James Gain. A review of digital terrain modeling. In *Computer Graphics Forum*, volume 38, pages 553–577. Wiley Online Library, 2019. 1, 2

[16] Lin Gao, Tong Wu, Yu-Jie Yuan, Ming-Xian Lin, Yu-Kun Lai, and Hao Zhang. Tm-net: Deep generative networks for textured meshes. *ACM Transactions on Graphics (TOG)*, 40(6):1–15, 2021. 1, 2

[17] Niv Granot, Ben Feinstein, Assaf Shocher, Shai Bagon, and Michal Irani. Drop the gan: In defense of patches nearest neighbors as single image generative models. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2, 3, 4, 6

[18] Jiatao Gu, Lingjie Liu, Peng Wang, and Christian Theobalt. Stylenerf: A style-based 3d aware generator for high-resolution image synthesis. In *International Conference on Learning Representations (ICLR)*, 2022. 1, 2

[19] Éric Guérin, Julie Digne, Éric Galin, Adrien Peytavie, Christian Wolf, Bedrich Benes, and Benoît Martinez. Interactive example-based terrain authoring with conditional generative adversarial networks. *ACM Transactions on Graphics (TOG)*, 36(6):228:1–228:13, Nov. 2017. 1, 2

[20] Niv Haim, Ben Feinstein, Niv Granot, Assaf Shocher, Shai Bagon, Tali Dekel, and Michal Irani. Diverse generation from a single video made possible. *arXiv preprint arXiv:2109.08591*, 2021. 2, 3

[21] Zekun Hao, Arun Mallya, Serge Belongie, and Ming-Yu Liu. Gancraft: Unsupervised 3d neural rendering of minecraft worlds. In *International Conference on Computer Vision (ICCV)*, pages 14072–14082, 2021. 1, 2

[22] Gur Harary, Ayellet Tal, and Eitan Grinspun. Context-based coherent surface completion. *ACM Transactions on Graphics (TOG)*, 33(1):1–12, 2014. 3

[23] Peter Hedman, Pratul P. Srinivasan, Ben Mildenhall, Jonathan T. Barron, and Paul Debevec. Baking neural radiance fields for real-time view synthesis. *International Conference on Computer Vision (ICCV)*, 2021. 3

[24] Tobias Hinz, Matthew Fisher, Oliver Wang, and Stefan Wermter. Improved techniques for training single-image gans. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1300–1309, 2021. 3

[25] Arjun Jain, Thorsten Thormählen, Tobias Ritschel, and Hans-Peter Seidel. Material memex: Automatic material suggestions for 3d objects. *ACM Transactions on Graphics (TOG)*, 31(6):1–8, 2012. 1, 2

[26] Konrad Kapp, James Gain, Eric Guérin, Eric Galin, and Adrien Peytavie. Data-driven authoring of large-scale ecosystems. *ACM Transactions on Graphics (TOG)*, 39(6):1–14, 2020. 1, 2

[27] Animesh Karnewar, Tobias Ritschel, Oliver Wang, and Niloy Mitra. 3inGAN: Learning a 3D generative model from images of a self-similar scene. In *Proc. 3D Vision (3DV)*, 2022. 3

[28] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4401–4410, 2019. 3

[29] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8110–8119, 2020. 3

[30] Tom Kelly, Paul Guerrero, Anthony Steed, Peter Wonka, and Niloy J Mitra. Frankengan: guided detail synthesis for building mass-models using style-synchonized gans. *ACM Transactions on Graphics (TOG)*, 2018. 1, 2

[31] Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. *Advances in neural information processing systems*, 31, 2018. 3

[32] Sung-Ho Lee, Taejung Park, Jong-Hyun Kim, and Chang-Hun Kim. Adaptive synthesis of distance fields. *IEEE Trans. Vis. & Comp. Graphics*, 18(7):1135–1145, 2011. 2

[33] Jun Li, Kai Xu, Siddhartha Chaudhuri, Ersin Yumer, Hao Zhang, and Leonidas Guibas. Grass: Generative recursive autoencoders for shape structures. *ACM Transactions on Graphics (TOG)*, 36(4):1–14, 2017. 1, 2

[34] Manyi Li, Akshay Gadi Patil, Kai Xu, Siddhartha Chaudhuri, Owais Khan, Ariel Shamir, Changhe Tu, Baoquan Chen, Daniel Cohen-Or, and Hao Zhang. Grains: Generative recursive autoencoders for indoor scenes. *ACM Transactions on Graphics (TOG)*, 38(2):1–16, 2019. 1, 2

[35] Ruihui Li, Xianzhi Li, Ka-Hei Hui, and Chi-Wing Fu. Sp-gan: Sphere-guided 3d shape generation and manipulation. *ACM Transactions on Graphics (TOG)*, 40(4):1–12, 2021. 1, 2

[36] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. *ACM Transactions on Graphics (SIGGRAPH)*, 21(4):163–169, 1987. 4

[37] Rui Ma, Akshay Gadi Patil, Matthew Fisher, Manyi Li, Sören Pirk, Binh-Son Hua, Sai-Kit Yeung, Xin Tong, Leonidas Guibas, and Hao Zhang. Language-driven synthesis of 3d scenes from scene databases. *ACM Transactions on Graphics (TOG)*, 37(6):1–16, 2018. 1, 2

[38] Paul Merrell. Example-based model synthesis. In *Proceedings of the 2007 symposium on Interactive 3D graphics and games*, pages 105–112, 2007. 1, 2

[39] Paul Merrell and Dinesh Manocha. Continuous model synthesis. In *SIGGRAPH Asia*, pages 1–7. 2008. 1, 2

[40] Paul Merrell and Dinesh Manocha. Model synthesis: A general procedural modeling algorithm. *IEEE Trans. Vis. & Comp. Graphics*, 17(6):715–728, 2010. 1, 2

[41] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision (ECCV)*, 2020. 3, 4

[42] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 41(4):102:1–102:15, July 2022. 3

[43] Michael Niemeyer and Andreas Geiger. Giraffe: Representing scenes as compositional generative neural feature fields. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 1, 2

[44] Gen Nishida, Ignacio Garcia-Dorado, Daniel G Aliaga, Bedrich Benes, and Adrien Bousseau. Interactive sketching of urban procedural models. *ACM Transactions on Graphics (TOG)*, 35(4):1–11, 2016. 1, 2

[45] Keunhong Park, Konstantinos Rematas, Ali Farhadi, and Steven M Seitz. Photoshape: Photorealistic materials for large-scale shape collections. *arXiv preprint arXiv:1809.09761*, 2018. 1, 2

[46] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv*, 2022. 3

[47] Gernot Riegler and Vladlen Koltun. Free view synthesis. In *European Conference on Computer Vision (ECCV)*, 2020. 3

[48] Gernot Riegler and Vladlen Koltun. Stable view synthesis. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 3

[49] Guodong Rong and Tiow-Seng Tan. Jump flooding in gpu with applications to voronoi diagram and distance transform. In *Proceedings of the 2006 Symposium on Interactive 3D Graphics and Games*, I3D '06, page 109–116, New York, NY, USA, 2006. Association for Computing Machinery. 5

[50] Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. Graf: Generative radiance fields for 3d-aware image synthesis. *Advances in Neural Information Processing Systems (NeurIPS)*, 33:20154–20166, 2020. 1, 2

[51] Tamar Rott Shaham, Tali Dekel, and Tomer Michaeli. Singan: Learning a generative model from a single natural image. In *International Conference on Computer Vision (ICCV)*, 2019. 3, 4, 6

[52] Andrei Sharf, Marc Alexa, and Daniel Cohen-Or. Context-based surface completion. In *ACM Transactions on Graphics (TOG)*, pages 878–887. 2004. 3

[53] Assaf Shocher, Shai Bagon, Phillip Isola, and Michal Irani. Ingan: Capturing and retargeting the" dna" of a natural image. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4492–4501, 2019. 3

[54] Denis Simakov, Yaron Caspi, Eli Shechtman, and Michal Irani. Summarizing visual data using bidirectional similarity. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8. IEEE, 2008. 2, 5

[55] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5459–5469, 2022. 3

[56] Ayush Tewari, Ohad Fried, Justus Thies, Vincent Sitzmann, Stephen Lombardi, Kalyan Sunkavalli, Ricardo Martin-Brualla, Tomas Simon, Jason Saragih, Matthias Nießner, et al. State of the art on neural rendering. In *Computer Graphics Forum*, volume 39, pages 701–727. Wiley Online Library, 2020. 3

[57] Arash Vahdat and Jan Kautz. NVAE: A deep hierarchical variational autoencoder. In *Neural Information Processing Systems (NeurIPS)*, 2020. 3

[58] Arash Vahdat, Karsten Kreis, and Jan Kautz. Score-based generative modeling in latent space. In *Neural Information Processing Systems (NeurIPS)*, 2021. 3

[59] Kai Wang, Manolis Savva, Angel X Chang, and Daniel Ritchie. Deep convolutional priors for indoor scene synthesis. *ACM Transactions on Graphics (TOG)*, 37(4):1–14, 2018. 1, 2

[60] Yujie Wang, Xuelin Chen, and Baoquan Chen. Singrav: Learning a generative radiance volume from a single natural scene. *arXiv preprint arXiv:2210.01202*, 2022. 3

[61] Rundi Wu, Xuelin Chen, Yixin Zhuang, and Baoquan Chen. Multimodal shape completion via conditional generative adversarial networks. In *European Conference on Computer Vision (ECCV)*, pages 281–296. Springer, 2020. 6

[62] Rundi Wu and Changxi Zheng. Learning to generate 3d shapes from a single example. *ACM Transactions on Graphics (TOG)*, 41(6), 2022. 3

[63] Wenming Wu, Xiao-Ming Fu, Rui Tang, Yuhan Wang, Yu-Hao Qi, and Ligang Liu. Data-driven interior plan generation for residential buildings. *ACM Transactions on Graphics (TOG)*, 38(6):1–12, 2019. 1, 2

[64] Yiheng Xie, Towaki Takikawa, Shunsuke Saito, Or Litany, Shiqin Yan, Numair Khan, Federico Tombari, James Tompkin, Vincent Sitzmann, and Srinath Sridhar. Neural fields in visual computing and beyond. *arXiv preprint arXiv:2111.11426*, 2021. 3

[65] Wang Ximeng. A thousand li of rivers and mountains. https://en.wikipedia.org/wiki/Wang_Ximeng, 1113. 7

[66] Rui Xu, Xintao Wang, Kai Chen, Bolei Zhou, and Chen Change Loy. Positional encoding as spatial inductive bias in gans. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13569–13578, 2021. 3

[67] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 3

[68] Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Basri Ronen, and Yaron Lipman. Multiview neural surface reconstruction by disentangling geometry and appearance. *Advances in Neural Information Processing Systems (NeurIPS)*, 33, 2020. 3

[69] Alex Yu, Sara Fridovich-Keil, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2, 3, 4

[70] Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. PlenOctrees for real-time rendering of neural radiance fields. In *International Conference on Computer Vision (ICCV)*, 2021. 3

[71] Lap Fai Yu, Sai Kit Yeung, Chi Keung Tang, Demetri Terzopoulos, Tony F Chan, and Stanley J Osher. Make it home: automatic optimization of furniture arrangement. *ACM Transactions on Graphics (TOG)*, 30(4), 2011. 1, 2

[72] Zaiwei Zhang, Zhenpei Yang, Chongyang Ma, Linjie Luo, Alexander Huth, Etienne Vouga, and Qixing Huang. Deep generative modeling for scene synthesis via hybrid representations. *ACM Transactions on Graphics (TOG)*, 39(2):1–21, 2020. 1, 2

[73] Howard Zhou, Jie Sun, Greg Turk, and James M Rehg. Terrain synthesis from digital elevation models. *IEEE Trans. Vis. & Comp. Graphics*, 13(4):834–848, 2007. 1, 2

[74] Chenyang Zhu, Kai Xu, Siddhartha Chaudhuri, Renjiao Yi, and Hao Zhang. Scores: Shape composition with recursive substructure priors. *ACM Transactions on Graphics (TOG)*, 37(6):1–14, 2018. 1, 2

[75] Jie Zhu, Yanwen Guo, and Han Ma. A data-driven approach for furniture and indoor scene colorization. *IEEE Trans. Vis. & Comp. Graphics*, 24(9):2473–2486, 2017. 1, 2