A Survey in Neural Volume Methods

Tianhang Cheng
tcheng12@illinois.edu

Ziyang Xie
ziyang8@illinois.edu

Haozhen Zheng
haozhen3@illinois.edu

1. Introduction

Neural volume has recently emerged as a powerful method for model and render 3D scenes using neural networks. Traditional 3D representations sucha s mesh, pointclouds and voxel are limited by their explicit nature, memory requirements and resolution constraints. These conventional representations struggle to model 3D scenes in high levels of detail or efficiency at scale. Neural volume representations address these limitations by leveraging neural networks to encode 3D scenes with higher flexibility.

In this survey, we break down neural volume methods into three main categories: continous representation, discrete representation and neural rendering. Continuous representations (Sec. 2) model 3D scenes with implicit functions that can be evaluated at arbitrary resolutions. Discrete representations (Sec. 3), on the other hand, divide the 3D space into fixed voxels or structures, enabling efficient computation. Finally, we will explore how advanced neural volume methods support neural rendering (Sec. 4), techniques, offering a brand new field for 3D vision and graphics. Each category has its strengths and trade-offs, contributing to the rapid advancements in neural volume methods.

2. Continuous Representation

In continuous representations, 3D coordinates $\mathbf{x} = (x, y, z)$ are mapped to various properties, such as signed distance, occupancy or radiance, via a continuous function $f(\mathbf{x})$ to represent the scene, enabling smooth, high-resolution modeling of 3D scenes without the constraints of fixed structures.

2.1. Implicit Function Representations

DeepSDF [30] represents 3D shapes as continuous signed distance functions (SDFs), enabling accurate reconstruction from incomplete or sparse observations by defining surfaces as the zero level set of a scalar field. Similarly, Occupancy Networks [25] introduce implicit representations by modeling 3D surfaces as decision boundaries of neural network classifiers, providing a more memory-efficient alternative to voxel grids while still delivering precise shape reconstructions.

To better handle complex surfaces and high-frequency details, SIREN [37] utilizes sinusoidal activation functions to represent 3D shapes with smooth and continuous geometry, making it especially effective for modeling complex surface features. NKSR [17] extends this approach by employing Neural Kernel Fields to reconstruct surfaces from sparse point clouds, effectively handling noise and efficiently scaling for larger scenes.

Recently, Neural Radiance Fields (NeRF) [26] have emerged as a significant breakthrough in representing 3D scenes with its powerful novel view synthesis ability. NeRF takes a 3D location and a viewing direction as inputs and outputs the color and density of the scene of that point. By integrating through the camera rays using volume rendering, NeRF can synthesize novel views of a scene with photorealistic quality. Unlike explicit representations like voxel grids or meshes, NeRF allows for efficient modeling of high-resolution details in both geometry and appearance from sparse views.

2.2. Primitive-Based Representations

Although implicit function-based methods offer a more flexible way to represent 3D shapes, recent work has explored hybrid approaches that combine the advantages of implicit representations with primitive-based modeling to provide improved control and interpretability for neural volume representation.

For example, Neural Parts [31] introduces a novel continuous 3D primitive representation using invertible neural networks. This method abstracts complex 3D shapes into deformable neural primitives, allowing for expressive and flexible shape modeling. Mixture of Volumetric Primitives [23] introduces a technique where complex shapes are decomposed into a set of volumetric primitives, enabling efficient neural rendering by leveraging these primitives for both global and local scene representations. This approach allows for faster and more memory-efficient rendering while preserving high fidelity in scene reconstruction. Similarly, HybridSDF [45] combines geometric primitives like spheres and cylinders with deep implicit surfaces, empowering users to directly manipulate shape features by editing geometric parameters.

3. Discrete Representation

Different from continues representation that maps a 3D coordinate with a continuous function, like MLP, discrete representation directly represents the scene with many individual elements, like 3D grid and discrete points. We will discuss different types in the following sections.

3.1. Grid-based methods

For example, Neural Volumes [22] encodes multiview images into a 3D grid with 3D convolution network. When rendering novel views, they apply ray marching in the cube and warp the ray coordinates to the cube. Similarly, DVGO [38] represent the scene as a dense feature cube. Each coordinate in the ray will sample feature from the grid and decode into density and color with a small MLP. These models usually converge very fast, but waste storage space in empty areas and are less efficient.

3.2. Point-based methods

While grid-based method adapts regular grid to represent 3D scene, point-based methods represent the scene as an unordered sets of discrete points or point-like structure. SNP [56] attaches spherical harmonics (SHs) for each point and do differential rendering. Neural Point [1] attaches descriptors with each point in the scene, and use a 2D Convolution Network to decode the rasterized feature image into RGB image. Point-NeRF [52] first extract point cloud with features by aggregating the depth map and 2D CNN features. Then it uses ray marching to find the K nearest neighbors and decode into color and density with a small MLP. The popular Gaussian-Splatting [16, 18, 49] in recent days represent the scene as a set of 3D Gaussians, which reaches high quality real-time rendering result.

The advantage of point-based representation is that they can modeling complex scenes and obtaining their photo-realistic views, while avoiding explicit surface estimation. Moreover, since different point are independent to each other, it's easy to edit or manipulate the local space of the scene. For instance, we could do realistic physical simulation [51] for the scene by directly applying Material Point Method (MPM) on Gaussian kernels.

However, it's hard to extract high quality geometry or mesh from these discrete representations. To resolve this, people usually constrain the points to be close to surface. For example, 2DGS [16] forces the Gaussian-Splatting to be as close to a flat disk as possible, and NeuralCAD [8] achieve this by enforcing zero gaussian curvature. Another limitations is that point-based representation will takes a lot of memory to save a scene. To resolve this, people usually adapts hierarchical representation or compress the GS into smaller size. For example, Contextgs [47] add anchors in the scene and computes the cosine similarity between each gaussians, which can represent the scene with 10 times compression. Octree-GS [34] introduces LOD-structured 3D Gaussian approach supporting level-of-detail decomposition for scene representation that contributes to the final rendering results. In this way, they could compress and accelerate GS query

3.3. Mesh-based methods

MobileNeRF [5] represents the scene as a grid mesh with a learnable deformation field and feature field. They first rasterize the mesh to a deferred rendering buffer. For each visible fragment, they execute a neural deferred shader that converts the feature and view direction to the corresponding output pixel color. Similarily, Nvdiffrec [28] and Nvdiffrec-MC [14] directly optimize topology of a triangle mesh, learn materials through volumetric texturing. These works can extract explicit mesh after optimization, which can used in advanced scene editing, material decomposition, and high quality view interpolation, all running at interactive rates in triangle-based renderers (rasterizers and path tracers).

3.4. Other types

Plane-based methods [12, 42] represent the scene as multi-plane images (MPI), and we can render disparity or novel views from these parallel planes. These methods does not require complex traning, but only works for single-view input and cannot handle extreme viewing angles.

Triangle meshes have difficulty modeling thin structures like hair, volumetric representations like Neural Volumes are too low-resolution given a reasonable memory budget, and high-resolution implicit representations like Neural Radiance Fields are too slow for use in real-time applications. Therefore, primitives-based methods [24] represent the scene as a mixture of volumetric primitives (MVP), which combines the completeness of volumetric representations with the efficiency of primitive-based rendering when render dynamic 3D content. However, there are often high overlap between adjacent volumetric primitives, thus the efficiency is lower than GS-based methods.

4. Efficient Neural Rendering

3D scene reconstruction that supports real-time rendering is crucial for various computer vision and graphics applications, such as games, robotics, and telepresence. Existing methods like NeRF[26], DeepSDF[30], JaxNeRF[6], Mip-NeRF[2], and JaxNeRF+[6] have achieved state-of-the-art performance with compact neural representations. Despite requiring small memory and rendering view-dependent, photo-realistic scenes with highly complex geometry, these methods [6, 26, 30] usually necessitate prolonged training periods and are unable to achieve real-time rendering capabilities. NeRF, with its purely MLP-based structure, takes

more than 20 hours to converge, and rendering must be performed by querying the MLP multiple times. This makes the decoding of RGB colors prohibitively expensive, resulting in slow rendering speeds.

- Section 3.1 covers neural primitive methods that enable real-time rendering based on volumetric representations. Some NeRF-based models [11, 21, 32, 55] pre-store the radiance field in an explicit volumetric representation, allowing for direct extraction of feature values from voxel grids during rendering. While there are other approaches based on multiplane images (MPI) [48] or layered depth images [7, 36, 43], we primarily focus on the prior trend of methods here.
- Section 3.2 introduces methods for learning hierarchical representations that enable the rendering of view-dependent scenes in a coarse-to-fine manner.
- In Section 3.3, we will explore methods that accelerate rendering speed through hybrid representations, allowing for real-time rendering while ensuring fine-grained reconstruction with various levels of detail.

4.1. Neural Volume Primitive Representation

Recent advancements in neural graphics have led to the development of neural primitives that leverage volumetric representations for efficient rendering and optimization. These volumetric representations enable rapid access to scene data by allowing for direct querying of voxel information. This direct querying facilitates quick lookups of spatial details, significantly reducing the computational overhead typically associated with traditional neural rendering methods.

The purely MLP-based NeRF necessitates more than 100 neural network evaluations to render a single image pixel. Instead of computing the radiance field via such a heavy MLP, SNeRG [15] accelerates the rendering process by storing the trained NeRF in sparse 3D voxel grids and learning a shallower MLP decoder to compute the view-dependent specular color, leveraging the ray marching algorithm on the grids. Different from NeRF's learning objective, PlenOctree [55] further eliminates the viewing direction as input to the model by learning a spherical harmonic function. Octree-structured voxel grids sampled from the model after training allow further fine-tuning directly on the grid using the NeRF loss. DIVeR [50] also leverages voxel grids to pre-store the trained NeRF and uniformly samples the ray (one sample per grid) during inference, but this method does not support fast rendering as effectively as SNeRG and PlenOctree.

There is also another trend among NeRF-based works that directly optimize the sparse voxel grids using different learning functions, such as spherical harmonic functions. NSVF [21] directly optimizes the voxel embeddings to learn local properties and designs a fast differentiable rendering method to ray-sample via the sparse voxel grid. DVGO [39] also directly optimizes the volume density explicitly encoded in a voxel grid while imposing two priors to avoid suboptimal geometry resolution. Notably, DVGO achieves real-time rendering and contributes to super-fast convergence speed ($\leq$ 15 minutes on NeRF datasets [26]). Concurrently, PlenOctrees can achieve performance on par with a NeRF model trained for 10 hours after merely 15 minutes of fine-tuning. However, they require over 50 hours for generalizable pretraining. MVSNeRF [3] uses a CNN network to reconstruct a neural scene encoding volume that consists of per-voxel neural features, followed by a decoder MLP that produces the radiance field by trilinearly interpolating the features. Unlike PlenOctrees and MVSNeRF, Plenoxels [10] is an end-to-end model that learns the per-voxel density and the harmonic coefficients for each color channel. Both MVSNeRF and Plenoxels compute the within-grid RGB value along the ray sampling by trilinearly interpolating the values resting at the vertices of the grid, while PlenOctrees treat the RGB values as constant within a voxel grid. NGLOD [40] encodes the model using a sparse voxel octree that holds a collection of features Z. Given a 3D coordinate x, the sum of the corresponding interpolated features of x from various depth levels of the octree is fed into an MLP to obtain a signed distance, enabling the off-the-shelf method of sphere tracing [13] for fast rendering of neural SDFs.

In contrast to the aforementioned methods, which sample rays from an explicit volumetric representation, FastNeRF [11] caches the deep radiance map components, allowing for more efficient memory usage. By splitting the network into two tasks—one solely dependent on position p and the other only on the ray direction d—FastNeRF makes caching feasible to fit into the CPU/GPU memory of commercial machines by caching various combinations of p and d. Similarly, TensoRF [4] also separately models the density σ and view-dependent color c, leveraging a per-voxel multi-channel feature 3D grid. This splitting approach makes caching feasible and potentially supports various types of appearance features depending on different learning functions (such as small MLPs or spherical harmonic functions).

4.2. Hierarchical Representation

Building on the concept of neural volume representation, hierarchical representations significantly enhance rendering speed by enabling efficient management of complex scenes and facilitating selective detail rendering based on the viewer's perspective. Methods such as NGLOD [40], Xcube [35], NeuralVDB [19], Plenoxels [10], and DVGO [39] can be viewed through a multi-resolution or hierarchical lens, as they all employ techniques that optimize rendering by transitioning from coarse to fine detail. This approach captures the overall structure and layout of the scene

while enhancing visual fidelity through fine-grained details, allowing for the rendering of view-dependent scenes with complex geometry and ensuring efficient rendering without sacrificing quality.

One form of hierarchical representation involves obtaining multiple feature vectors from each level of a multi-resolution octree, given a coordinate x. These vectors can be summed or concatenated in a Laplacian pyramid fashion as input to an MLP. For instance, InstantNGP [27] concatenates the interpolated features from multi-resolution hash table grids as input to the MLP for the radiance field. Such a data structure allows the hash tables for all resolutions to be queried in parallel, facilitating CUDA caching by sequentially processing level-to-level hash tables.

Similarly, NGLOD sums the feature values from each level of the octree as input to the MLP, and [41] arranges features in a multi-resolution sparse octree, training a separate codebook for each level of the tree.

4.3. Hybrid Representation

Neural volume representations have proven to be capable of efficient rendering (some $\geq$ 150 FPS). However, these methods are sometimes limited by caching capacity and may not capture fine-grained details in large, complex scenes. Although these approaches enhance rendering speed, they rarely achieve the performance needed for real-time rendering on high-definition datasets; for instance, InstantNGP [27] achieves less than 10 FPS on high-resolution real-world datasets.

While hierarchical representations effectively enhance rendering performance for fine-grained details by leveraging multi-resolution features, hybrid representations further optimize this process by combining the strengths of both explicit and implicit modeling approaches. Approximately 95% of a scene can be efficiently modeled as mere surface during convergence [44]. By integrating neural surface primitives with volumetric representation, these hybrid methods allow for more flexible and lightweight scene modeling, enabling even faster rendering (40 FPS) on high-resolution datasets.

Several methods [29, 46, 54] derive density values from SDF models and render them as volumetric representations. By leveraging the hierarchical learning paradigm, UNISURF [29] first reduces the sampling area by recovering surfaces in the initial stage and subsequently optimizes volumetrically within this reduced area. These methods retain fast rendering speed while improving surface geometry.

Building on the idea of reducing sample density to accelerate training and rendering, HybridNeRF [44] proposes a weighted Eikonal regularization (an adaptive level-of-detail indicator) to indicate where the model cannot accurately reconstruct the scene via an SDF and bakes the network into binary occupancy grids during the rendering stage for cheaper retrieval, following the framework of MERF [33].

5. Datasets

In the realm of 3D reconstruction, the choice of dataset plays a crucial role in training and evaluating algorithms. Various datasets have been developed to provide diverse scenes, varying lighting conditions, and different geometrical complexities. This section introduces several commonly used datasets that facilitate the advancement of 3D reconstruction and novel view synthesis techniques, particularly those utilizing Neural Radiance Fields (NeRF) and other volumetric representations. Each dataset offers unique characteristics and serves specific purposes in benchmarking methods for view synthesis, depth estimation, and scene understanding. The following subsections detail the key attributes of these datasets.

- **NeRF-Synthetic[26]:** This dataset comprises a variety of synthetic scenes, each containing multiple images captured from different viewpoints. It includes scenes like the *Lego*, *Ficus*, and *Fern*, and is specifically designed for training and evaluating Neural Radiance Fields (NeRF). The dataset provides ground truth depth information, making it suitable for benchmarking view synthesis methods.
- **BlendedMVS[53]:** BlendedMVS is a multi-view stereo dataset designed for training and evaluating methods in 3D reconstruction. It consists of a diverse set of indoor and outdoor scenes captured with various camera setups. The dataset provides dense depth maps, high-quality RGB images, and associated camera poses, enabling the assessment of depth estimation and scene reconstruction performance.
- **Tanks & Temples[20]:** This benchmark dataset is focused on evaluating 3D reconstruction algorithms. It includes a series of challenging outdoor and indoor scenes with complex geometries and lighting conditions. Each scene comes with high-quality images, camera poses, and ground truth geometry, making it ideal for assessing the accuracy and robustness of reconstruction methods.
- **Shiny[48]**: This benchmark dataset contains 8 scenes captured with smartphone with challenging view-dependent effects. For instance, the scenes contain significant challenging patterns like rainbow reflection on the CD and refraction through a liquid bottle, making no known methods to handle extremely sharp light
- **Spaces[9]**: This benchmark dataset is introduced in DeepView[9] to test the challenging light field captures of view synthesis models. Spaces consists of 100 indoor and outdoor scenes, captured using a 16-camera rig. The distance between these cameras is approximately 10cm, which allows the mixture of different camera views during training, making it flexible to evaluate on model's view synthesis capability.

References

[1] Kara-Ali Aliev, Artem Sevastopolsky, Maria Kolos, Dmitry Ulyanov, and Victor Lempitsky. Neural point-based graphics. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXII 16*, pages 696–712. Springer, 2020. 2

[2] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5855–5864, 2021. 2

[3] Anpei Chen, Zexiang Xu, Fuqiang Zhao, Xiaoshuai Zhang, Fanbo Xiang, Jingyi Yu, and Hao Su. Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 14124–14133, 2021. 3

[4] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *European conference on computer vision*, pages 333–350. Springer, 2022. 3

[5] Zhiqin Chen, Thomas Funkhouser, Peter Hedman, and Andrea Tagliasacchi. Mobilenerf: Exploiting the polygon rasterization pipeline for efficient neural field rendering on mobile architectures. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16569–16578, 2023. 2

[6] Boyang Deng, Jonathan T Barron, and Pratul P Srinivasan. Jaxnerf: an efficient jax implementation of nerf. *URL http://github. com/googleresearch/google-research/tree/master/jaxnerf*, 2020. 2

[7] Helisa Dhamo, Keisuke Tateno, Iro Laina, Nassir Navab, and Federico Tombari. Peeking behind objects: Layered depth prediction from a single image. *Pattern Recognition Letters*, 125:333–340, 2019. 3

[8] Qiujie Dong, Rui Xu, Pengfei Wang, Shuangmin Chen, Shiqing Xin, Xiaohong Jia, Wenping Wang, and Changhe Tu. Neurcadrecon: Neural representation for reconstructing cad surfaces by enforcing zero gaussian curvature. *arXiv preprint arXiv:2404.13420*, 2024. 2

[9] John Flynn, Michael Broxton, Paul Debevec, Matthew DuVall, Graham Fyffe, Ryan Overbeck, Noah Snavely, and Richard Tucker. Deepview: View synthesis with learned gradient descent. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2367–2376, 2019. 4

[10] Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5501–5510, 2022. 3

[11] Stephan J Garbin, Marek Kowalski, Matthew Johnson, Jamie Shotton, and Julien Valentin. Fastnerf: High-fidelity neural rendering at 200fps. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 14346–14355, 2021. 3

[12] Yuxuan Han, Ruicheng Wang, and Jiaolong Yang. Single-view view synthesis in the wild with learned adaptive multiplane images. In *ACM SIGGRAPH 2022 Conference Proceedings*, pages 1–8, 2022. 2

[13] John C Hart. Sphere tracing: A geometric method for the antialiased ray tracing of implicit surfaces. *The Visual Computer*, 12(10):527–545, 1996. 3

[14] Jon Hasselgren, Nikolai Hofmann, and Jacob Munkberg. Shape, light, and material decomposition from images using monte carlo rendering and denoising. *Advances in Neural Information Processing Systems*, 35:22856–22869, 2022. 2

[15] Peter Hedman, Pratul P Srinivasan, Ben Mildenhall, Jonathan T Barron, and Paul Debevec. Baking neural radiance fields for real-time view synthesis. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5875–5884, 2021. 3

[16] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–11, 2024. 2

[17] Jiahui Huang, Zan Gojcic, Matan Atzmon, Or Litany, Sanja Fidler, and Francis Williams. Neural kernel surface reconstruction. In *CVPR*, pages 4369–4379, 2023. 1

[18] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023. 2

[19] Doyub Kim, Minjae Lee, and Ken Museth. Neuralvdb: High-resolution sparse volume representation using hierarchical neural networks. *ACM Transactions on Graphics*, 43 (2):1–21, 2024. 3

[20] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics (ToG)*, 36 (4):1–13, 2017. 4

[21] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. *Advances in Neural Information Processing Systems*, 33:15651–15663, 2020. 3

[22] Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. Neural volumes: Learning dynamic renderable volumes from images. *arXiv preprint arXiv:1906.07751*, 2019. 2

[23] Stephen Lombardi, Tomas Simon, Gabriel Schwartz, Michael Zollhoefer, Yaser Sheikh, and Jason Saragih. Mixture of volumetric primitives for efficient neural rendering. *ACM Trans. Graph.*, 2021. 1

[24] Stephen Lombardi, Tomas Simon, Gabriel Schwartz, Michael Zollhoefer, Yaser Sheikh, and Jason Saragih. Mixture of volumetric primitives for efficient neural rendering. *ACM Transactions on Graphics (ToG)*, 40(4):1–13, 2021. 2

[25] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1

[26] 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. *Communications of the ACM*, 65(1):99–106, 2021. 1, 2, 3, 4

[27] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM transactions on graphics (TOG)*, 41(4):1–15, 2022. 4

[28] Jacob Munkberg, Jon Hasselgren, Tianchang Shen, Jun Gao, Wenzheng Chen, Alex Evans, Thomas Müller, and Sanja Fidler. Extracting triangular 3d models, materials, and lighting from images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8280–8290, 2022. 2

[29] Michael Oechsle, Songyou Peng, and Andreas Geiger. Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5589–5599, 2021. 4

[30] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 165–174, 2019. 1, 2

[31] Despoina Paschalidou, Angelos Katharopoulos, Andreas Geiger, and Sanja Fidler. Neural parts: Learning expressive 3d shape abstractions with invertible neural networks. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021. 1

[32] Christian Reiser, Songyou Peng, Yiyi Liao, and Andreas Geiger. Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 14335–14345, 2021. 3

[33] Christian Reiser, Rick Szeliski, Dor Verbin, Pratul Srinivasan, Ben Mildenhall, Andreas Geiger, Jon Barron, and Peter Hedman. Merf: Memory-efficient radiance fields for real-time view synthesis in unbounded scenes. *ACM Transactions on Graphics (TOG)*, 42(4):1–12, 2023. 4

[34] Kerui Ren, Lihan Jiang, Tao Lu, Mulin Yu, Linning Xu, Zhangkai Ni, and Bo Dai. Octree-gs: Towards consistent real-time rendering with lod-structured 3d gaussians. *arXiv preprint arXiv:2403.17898*, 2024. 2

[35] Xuanchi Ren, Jiahui Huang, Xiaohui Zeng, Ken Museth, Sanja Fidler, and Francis Williams. Xcube: Large-scale 3d generative modeling using sparse voxel hierarchies. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4209–4219, 2024. 3

[36] Jonathan Shade, Steven Gortler, Li-wei He, and Richard Szeliski. Layered depth images. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 231–242, 1998. 3

[37] Vincent Sitzmann, Julien N.P. Martel, Alexander W. Bergman, David B. Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. In *arXiv*, 2020. 1

[38] 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. 2

[39] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Improved direct voxel grid optimization for radiance fields reconstruction. *arXiv preprint arXiv:2206.05085*, 2022. 3

[40] Towaki Takikawa, Joey Litalien, Kangxue Yin, Karsten Kreis, Charles Loop, Derek Nowrouzezahrai, Alec Jacobson, Morgan McGuire, and Sanja Fidler. Neural geometric level of detail: Real-time rendering with implicit 3d shapes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11358–11367, 2021. 3

[41] Towaki Takikawa, Alex Evans, Jonathan Tremblay, Thomas Müller, Morgan McGuire, Alec Jacobson, and Sanja Fidler. Variable bitrate neural fields. In *ACM SIGGRAPH 2022 Conference Proceedings*, pages 1–9, 2022. 4

[42] Richard Tucker and Noah Snavely. Single-view view synthesis with multiplane images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 551–560, 2020. 2

[43] Shubham Tulsiani, Richard Tucker, and Noah Snavely. Layer-structured 3d scene inference via view synthesis. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 302–317, 2018. 3

[44] Haithem Turki, Vasu Agrawal, Samuel Rota Bulò, Lorenzo Porzi, Peter Kontschieder, Deva Ramanan, Michael Zollhöfer, and Christian Richardt. Hybridnerf: Efficient neural rendering via adaptive volumetric surfaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19647–19656, 2024. 4

[45] Subeesh Vasu, Nicolas Talabot, Artem Lukoianov, Pierre Baqué, Jonathan Donier, and Pascal Fua. Hybridsdf: Combining deep implicit shapes and geometric primitives for 3d shape representation and manipulation. In *International Conference on 3D Vision*, 2022. 1

[46] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *arXiv preprint arXiv:2106.10689*, 2021. 4

[47] Yufei Wang, Zhihao Li, Lanqing Guo, Wenhan Yang, Alex C Kot, and Bihan Wen. Contextgs: Compact 3d gaussian splatting with anchor level context model. *arXiv preprint arXiv:2405.20721*, 2024. 2

[48] Suttisak Wizadwongsa, Pakkapon Phongthawee, Jiraphon Yenphraphai, and Supasorn Suwajanakorn. Nex: Real-time view synthesis with neural basis expansion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8534–8543, 2021. 3, 4

[49] Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 4d gaussian splatting for real-time dynamic scene rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20310–20320, 2024. 2

[50] Liwen Wu, Jae Yong Lee, Anand Bhattad, Yu-Xiong Wang, and David Forsyth. Diver: Real-time and accurate neural radiance fields with deterministic integration for volume rendering. In *Proceedings of the IEEE/CVF Conference on*

Computer Vision and Pattern Recognition, pages 16200–16209, 2022. 3

[51] Tianyi Xie, Zeshun Zong, Yuxing Qiu, Xuan Li, Yutao Feng, Yin Yang, and Chenfanfu Jiang. Physgaussian: Physics-integrated 3d gaussians for generative dynamics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4389–4398, 2024. 2

[52] Qiangeng Xu, Zexiang Xu, Julien Philip, Sai Bi, Zhixin Shu, Kalyan Sunkavalli, and Ulrich Neumann. Point-nerf: Point-based neural radiance fields. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5438–5448, 2022. 2

[53] Yao Yao, Zixin Luo, Shiwei Li, Jingyang Zhang, Yufan Ren, Lei Zhou, Tian Fang, and Long Quan. Blendedmvs: A large-scale dataset for generalized multi-view stereo networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1790–1799, 2020. 4

[54] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. *Advances in Neural Information Processing Systems*, 34:4805–4815, 2021. 4

[55] Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. Plenoctrees for real-time rendering of neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5752–5761, 2021. 3

[56] Yiming Zuo and Jia Deng. View synthesis with sculpted neural points. *arXiv preprint arXiv:2205.05869*, 2022. 2