Urban Radiance Field Representation with Deformable Neural Mesh Primitives

Fan Lu[1 *] Yan Xu[2 *] Guang Chen[1 †] Hongsheng Li[2,3,4] Kwan-Yee Lin[2,3 †] Changjun Jiang[1]
[1]Tongji University [2]The Chinese University of Hong Kong [3]Shanghai AI Laboratory [4]CPII
{lufan,guangchen,cjjiang}@tongji.edu.cn
yanxu@link.cuhk.edu.hk hsli@ee.cuhk.edu.hk junyilin@cuhk.edu.hk

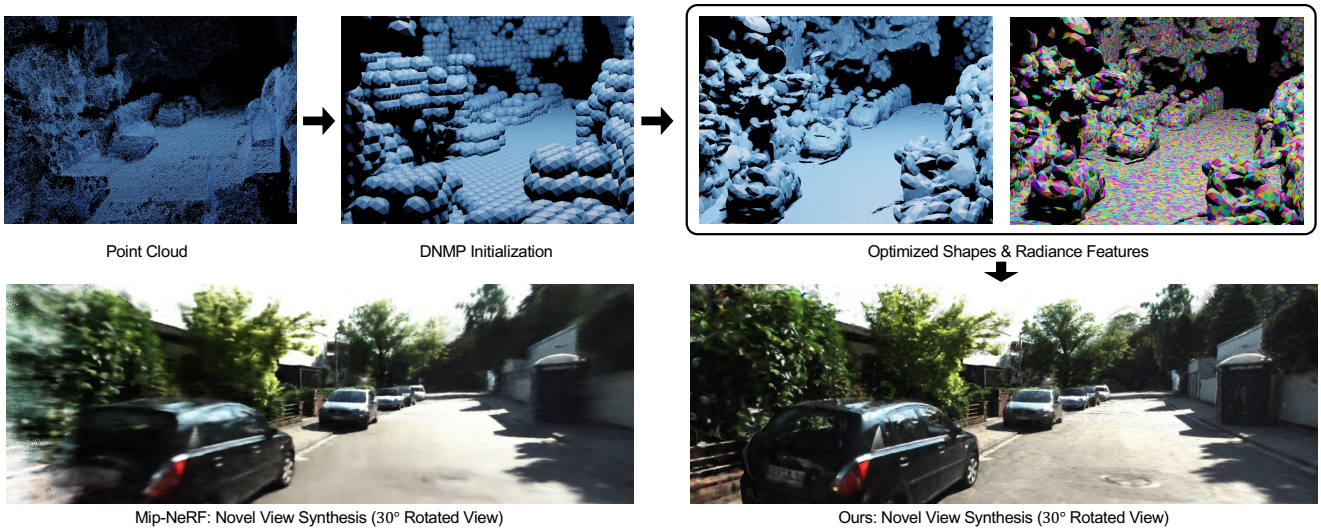

Figure 1. The basic idea of our method. Taking the patchy point clouds as inputs, we first voxelize the points and then initialize our Deformable Neural Mesh Primitive (DNMP) for each voxel. During training, the shapes of DNMPs are deformed to model the underlying 3D structures, while the radiance features of DNMPs are learnt to model the local radiance information for neural rendering. Based on our representation, we achieve efficient and photo-realistic rendering for urban scenes.

Abstract

Neural Radiance Fields (NeRFs) have achieved great success in the past few years. However, most current methods still require intensive resources due to ray marching-based rendering. To construct urban-level radiance fields efficiently, we design Deformable Neural Mesh Primitive (DNMP), and propose to parameterize the entire scene with such primitives. The DNMP is a flexible and compact neural variant of classic mesh representation, which enjoys both the efficiency of rasterization-based rendering and the powerful neural representation capability for photo-realistic image synthesis. Specifically, a DNMP consists of a set of connected deformable mesh vertices with paired vertex features to parameterize the geometry and radiance information of a local area. To constrain the degree of freedom for optimization and lower the storage budgets, we enforce the shape of each primitive to be decoded from a relatively low-dimensional latent space. The rendering colors are decoded from the vertex features (interpolated with rasterization) by a view-dependent MLP. The DNMP provides a new paradigm for urban-level scene representation with appealing properties: (1) *High-quality rendering. Our method achieves leading performance for novel view synthesis in urban scenarios.* (2) *Low computational costs. Our representation enables fast rendering (2.07ms/1k pixels) and low peak memory usage (110MB/1k pixels). We also present a lightweight version that can run 33× faster than vanilla NeRFs, and comparable to the highly-optimized Instant-NGP (0.61 vs 0.71ms/1k pixels). Project page: https://dnmp.github.io/.*

1. Introduction

Synthesizing photo-realistic images of 3D scenes is a long-standing problem in computer vision, and has been the focus of research in the past decades. However, even with years of effort, the current paradigms still face great

* equal contribution. Fan's contributions: 1) Code implementation; 2) Novel database construction for latent space training and Loss Eq. (1); 3) Conducted most of the experiments. Yan's contributions: 1) Proposed DNMP. The structure design of DNMP and latent space construction for shape control; 2) Proposed representing the entire scene with local primitives as well as the idea of hierarchical DNMP representation. 3) Drafted the paper and designed most of the experiments.

†corresponding authors.

challenges especially in urban outdoor scenarios, due to the increased representation complexity and demanding computational resources.

To achieve high-quality image rendering, the computer graphics community has explored various techniques for scene representation, including point clouds [23], meshes [20, 22], voxels [46], implicit functions [6, 21], *etc*. The mesh-based representation is widely used in modern rendering pipelines due to its compact and efficient nature. However, constructing water-tight mesh models of urban scenes for modern graphic engines is still difficult. Besides, the textures and illuminations are difficult to be realistically recovered with the classic techniques.

The recent neural rendering methods circumvent the mesh construction step and represent the scene with implicit neural functions. NeRF [31] and its advanced variants [29, 39, 54, 59, 4] proposed to store the density and radiance information of a volume inside multi-layer perceptrons (MLPs), and adopt volumetric rendering for view synthesis. Recently, some researchers [50] also made efforts to extend the NeRF models to large-scale scenes by independently representing a city block-by-block and merging the representations together thereafter. Although remarkable progress has been made, their rendering process is still computationally intensive as the implicit functions need to be evaluated thousands of times to densely sample the space during volumetric rendering. Most of the computational resources are wasted on the samples in empty spaces, and the situation will further escalate in outdoor scenes where empty space dominates.

More recently, researchers have identified this issue and propose to combine neural rendering with explicit point-cloud reconstruction to improve the efficiency [55, 36]. In this way, the empty spaces can be skipped by taking the explicit reconstruction as a reference, which significantly saves computational resources. These methods associate point-wise learnable high-dimensional features to the reconstructed point clouds for spatial radiance encoding. During rendering, only the points around the intersections of the view ray with the point clouds will be sampled for feature aggregation. Based on such aggregation mechanism, a dense and perfect reconstruction is vital for photo-realistic rendering. However, the reconstructed points are usually not uniformly distributed from the current reconstruction algorithms [18] and missing regions may be also ubiquitous. The noisy reconstruction will increase the learning difficulty of the implicit function and degrade the final rendering quality.

In this work, we propose an efficient radiance field representation for large-scale environments by combining efficient mesh-based rendering and powerful neural representations. Specifically, we develop Deformable Neural Mesh Primitive (DNMP) and propose to represent the entire radiance field in a bottom-up manner with such primitives, where each DNMP parameterizes the geometry and radiance of a local area. A DNMP consists of a set of connected deformable mesh vertices and each vertex is paired with a feature vector for radiance modeling. To constrain the degree of freedom for shape optimization and decrease the storage budgets, we enforce the mesh vertices of each DNMP to be decoded from a relatively low-dimensional latent code. The latent code will be optimized to deform the primitive shapes for 3D structure modeling during training.

Different from previous methods [55, 36] that rely on inefficient k-Nearest Neighbors (k-NN) algorithm to gather related features for rendering, we can directly leverage the rasterization pipeline for feature interpolation, which is more efficient. Based on rasterization, for each view ray, a set of features is collected by interpolations from the triangulated vertex features. These interpolated features are thereafter input to an implicit function (implemented with MLPs) to get the corresponding radiance and opacity values for volumetric rendering.

To represent the entire scene, we first coarsely voxelize the scene according to the 3D reconstruction results (from Multi-View Stereo (MVS) [45] or hardware sensors), and then parameterize the geometry and radiance of each voxel with a DNMP. Considering the practical 3D reconstruction results may be noisy and full of missing regions, we further propose to voxelize the scene with hierarchical resolutions and separately represent the radiance fields accordingly with hierarchically-sized DNMPs. The rendering results from different hierarchy levels will be blended. Based on our DNMP-based hierarchical representation, we achieve more robustness against noisy 3D reconstructions compared with the previous point-cloud based methods [55, 36].

We evaluate our method on two urban datasets, *i.e*., KITTI-360 [26] and Waymo Open Dataset [49]. Our method enables photo-realistic rendering and achieves leading performance for novel view synthesis. We achieve a much faster speed and produce fewer peak memory footprints compared with vanilla NeRFs. We also present a lightweight version to further accelerate the rendering. This lightweight version can run at an interactive rate only with limited sacrifices on rendering quality, the speed of which is even comparable with the highly-optimized Instant-NGP's [32]. Moreover, our method can be easily embedded into modern graphic rendering pipelines and naturally supports scene editing, which provides the potential for possible applications such as VR/AR.

2. Related Work

Neural rendering. The early-phase neural rendering techniques [30, 43, 25, 2] proposed to directly project 3D signals to a 2D image plane and train a 2D CNN that maps projected signals to the final output image. Their mapping process only relies on the CNN regression ability without explicit physical modeling of the 3D space, which could bring performance bottlenecks in synthesiz-

ing novel views. The recent volume rendering based approaches [31, 29, 39, 54, 59, 4, 9, 42, 14] alleviate such dilemmas by storing the densities and radiances of a scene within an implicit neural function and synthesizing novel views with volumetric rendering. Mip-NeRFs [4, 5] render anti-aliased conical frustums instead of rays, which are widely applied given the good rendering results in general. However, the implicit function needs to be evaluated thousands of times on the densely sampled space points for volumetric sampling, leading to inefficient training and inference. In the past few years, many methods are proposed to accelerate NeRFs. Some methods [19, 27, 35, 38, 55] use proxy scene structures or surface information to reduce the number of samples in empty spaces for inference acceleration, while other solutions improve the inference or training speed by marrying NeRF with efficient data structures [41, 17, 19, 57, 48, 13, 32, 7]. As a representative, Instant-NGP [32] significantly improves the efficiency of classic NeRF via the hierarchical space division and highly-optimized CUDA implementation. However, these methods lack explicit surface constraints, which may lead to less robustness against viewpoint changes for view synthesis. Moreover, all these methods still have not been widely supported by modern graphic rendering pipelines and also have difficulties in supporting scene editing for downstream applications.

Outdoor NeRFs. Recently, some researchers tried to extend NeRF to outdoor scenes [42, 29, 50, 36]. NeRF-W [29] incorporates frame-specific codes in the rendering pipeline to handle the photometric variation and transient objects. Block-NeRF [50] extends the NeRF to urban scenarios and builds up the block-wise radiance fields with individual NeRFs to composite the complete scene. However, these methods still rely on costly volumetric sampling as indicated above, which will waste huge amounts of computational resources in empty spaces. Rematas *et al*. [42] include LiDAR point clouds for supervision to facilitate geometry learning. Neural Point Light Field (NPLF) [36] leverages the explicit 3D reconstructions from LiDAR data to represent the radiance field for rendering efficiency. But they still simply aggregate several nearest feature points around each view ray for rendering without considering the scene geometry in detail, which causes bottlenecks for high-resolution rendering.

3D Shape Reconstruction. The classic pipeline for 3D reconstruction from color images usually first estimates the camera poses based on structure-from-motion [44, 1] and then recovers dense depths with Multi-View Stereo (MVS) techniques [45, 8, 15, 56, 51]. These methods can handle ideal scenarios but may generate incomplete reconstruction results under adverse conditions, *e.g*., illumination changes, textureless areas, *etc*. The methods based on implicit fields [37, 11, 58, 53] are generally more robust but an expensive iso-surfacing step [28] is required to extract the mesh from the representation, which is prone to quantization errors. Prior to our work, some works [52, 16, 47, 33] proposed deformable meshes for shape reconstruction and rendering. However, these techniques are still sensitive to practical noisy data and are usually limited to object-level shape optimization.

3. Method

To parameterize the large-scale urban radiance field effectively, we propose Deformable Neural Mesh Primitive (DNMP). DNMP is a neural variant of classic mesh representation, which takes advantage of both the efficient rasterization-based rendering and the powerful neural representation capability. A DNMP is capable of modeling the geometry and radiance information of a local 3D space in an expressive and compact manner, and the entire radiance field is hierarchically constituted by a series of DNMPs.

3.1. Deformable Neural Mesh Primitive

Triangle meshes are widely used in computer graphics. The meshes can compactly represent 3D surfaces and be efficiently rendered based on rasterization. To leverage the efficiency of mesh-based representation and the impressive radiance representation ability of neural features, we develop Deformable Neural Mesh Primitive (DNMP) and parameterize the entire scene with such primitives. DNMP is an enclosed triangle neural mesh, constituted by a set of deformable mesh vertices $\mathcal{V} = \{\mathbf{v}_i | i = 1, \dots, N\}$ paired with learnable vertex radiance features $\mathcal{F} = \{\mathbf{f}_i | i = 1, \dots, N\}$. The vertices define the shape of each DNMP, while the vertex features encode the radiance information of a local area.

Shape parameterization of DNMP. A DNMP may contain tens of vertices. To constrain the degree of freedom for shape optimization, we design an auto-encoder [24] to learn a compact latent space to parameterize the primitive shapes.

Specifically, we first establish a database that contains a huge number of meshes created from local 3D structures and train our auto-encoder with them. These local structures are collected from different types of datasets, both indoors and outdoors. We assume the database is large enough to capture all the possible local structural variations for compact latent space learning.

Our auto-encoder is designed based on PointNet [40], containing a shape encoder and a shape decoder. The shape encoder encodes the geometric information of differently-shaped meshes into compact shape latent codes $\mathbf{z}$. The shape decoder G directly decodes the shape latent code $\mathbf{z}$ to DNMP's vertices $\mathcal{V} = \{\mathbf{v}_i | i = 1, \dots, N\}$ (in a predefined order), *i.e*., $\mathcal{V} = G(\mathbf{z})$. The output vertices can be trivially converted to an enclosed mesh according to the predefined connectivity relations. To learn the latent space, we encourage the shapes of decoded DNMPs to match the input meshes. The training loss of the auto-encoder contains

The details are in supplementary materials.

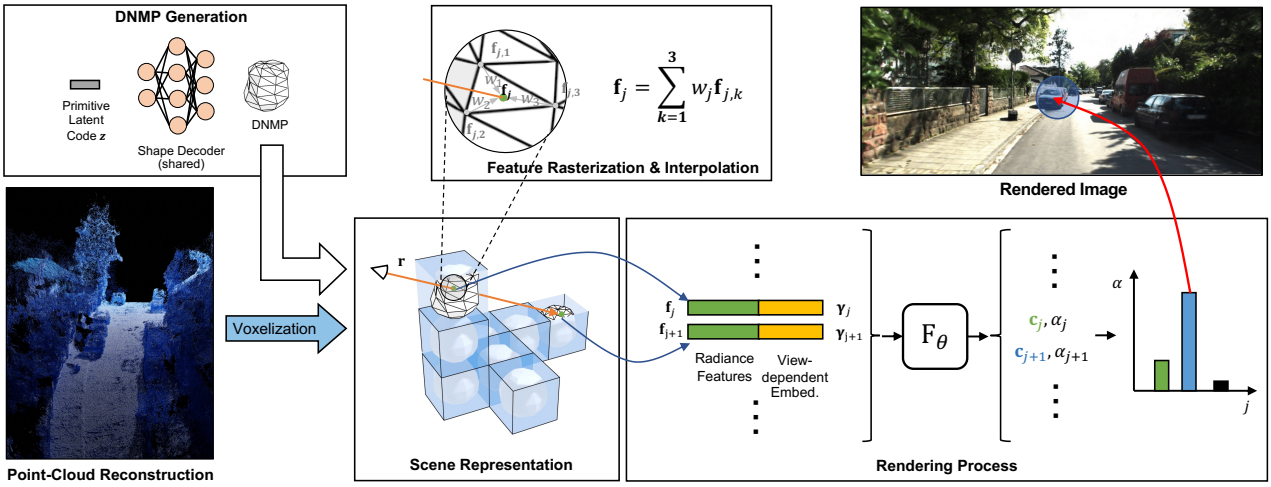

Figure 2. The overview of our framework. The entire scene is voxelized based on the point-cloud reconstruction, where each voxel is assigned a DNMP to parameterize the geometry and radiance of the local area. By rasterization, we can obtain the interpolated radiance features $\{\mathbf{f}_j | j = 0, 1, \dots J\}$ from the intersected DNMPs for each view ray $\mathbf{r}$. Thereafter, these interpolated features along with the view-dependent embeddings $\{\boldsymbol{\gamma}_j | j = 0, 1, \dots, J\}$ are sent to an implicit function F_θ to predict the radiance value c_j and opacity α_j of each intersection point. Finally, the rendering color $\hat{\mathbf{C}}(\mathbf{r})$ of the view ray $\mathbf{r}$ is obtained by blending the radiance values according to the opacities $\{\alpha_j | j = 0, 1, \dots, J\}$.

two parts:

$$L_{ae} = L_{chamfer}(\mathcal{V}) + L_{regularize}(\mathcal{V}). \quad (1)$$

The chamfer distance loss $L_{chamfer}$ enforces the closeness between two sets of randomly sampled points from the input and output mesh faces. Besides, we also encourage the normal consistency [10] and mesh smoothness [34] with a regularization loss $L_{regularize}$, which is described detailedly in supplementary materials. To constrain the latent code space, we normalize the latent code to unit length, *i.e.*, $||\mathbf{z}||_2 = 1$. The dimension of the latent space is empirically set to 8 for reliable shape optimization.

Radiance parameterization. To encode the radiance information, we associate independent learnable feature vectors to DNMP's vertices. During rendering, unlike the previous point-based methods [55, 36] that need to query features with time-consuming k-NN searching [12], we can efficiently obtain the related features via rasterization. Moreover, in this manner, the encoded radiance is better aligned with the local surfaces, which is helpful to improve the view consistency of the radiance field.

3.2. DNMP-Based Scene Representation

To construct a view-consistent radiance field, we emphasize the optimization of both geometry and radiance. For initialization, the target scene is first voxelized based on the point-cloud reconstruction from MVS [45] or hardware sensors. Then, each voxel is assigned a DNMP to parameterize the structure geometry and radiance information in the local area. Before training, the shape latent code of each DNMP is initialized to correspond to a spherical template (scaled to match the voxel size), as illustrated in Fig. 1. For convenience, we denote the set of all the shape latent codes of the scene as $\mathcal{Z} = \{\mathbf{z}_l | l = 1, 2, \dots, L\}$, where l indexes the DNMP and L is the total number of DNMPs of the scene.

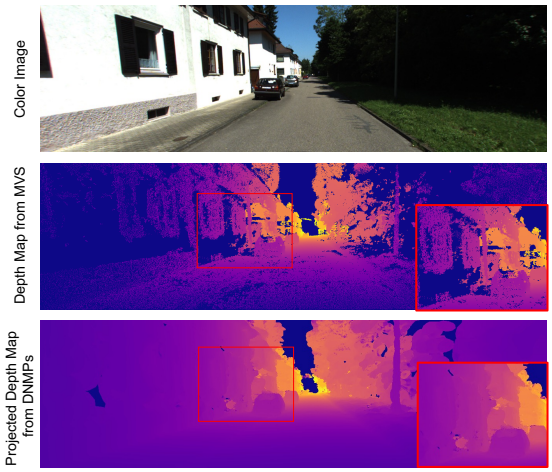

Figure 3. Even supervised with the incomplete depths with Eq. (2), the DNMPs can be generally deformed to reflect the underlying geometry thanks to the compact latent space learnt from huge number of local 3D structures.

DNMP-based shape optimization. To abstract the scene's geometry with DNMPs, we need to optimize the shape latent codes. We leverage the estimated depth maps D (by MVS [45] or other sensors) of captured video sequences of the scene to optimize the latent code. During training, we render a corresponding depth map $\hat{D}$ based on the current

We only show the rendering process for one hierarchy in this figure. The final color will be blended from several hierarchies with Eq. 5.

shapes of DNMPs in a differentiable manner. As the DNMP shapes are controlled by the latent codes $\mathcal{Z}$, the rendered depth $\hat{D}$ can be written as a differentiable function of latent codes $\mathcal{Z}$ as $\hat{D}(\mathcal{Z})$. Then, we supervise the rendered depth map $\hat{D}$ with the pre-estimated depth map D via the L1 loss

$$L_{geo} = ||\hat{D}(\mathcal{Z}) - D||_1, \tag{2}$$

which deforms the DNMPs to abstract the scene's geometry. We found that even supervised with imperfect depth maps D (shown in Fig. 3), the latent codes $\mathcal{Z}$ can be optimized to reflect the underlying geometry reasonably, thanks to the strong structural priors learnt by the pretrained shape decoder (Sec. 3.1) from huge amounts of data.

Hierarchical representation. As the 3D reconstruction results with MVS [45] may have many missing regions in outdoor environments, DNMP may thus fail to be initialized for the missing parts, which will cause unsatisfactory rendering results. To avoid this, we model the entire scene in a hierarchical manner. Concretely, we voxelize the reconstructed point cloud with hierarchical sizes, *e.g.*, 0.5m, 1m, *etc*. to make the missing regions can be covered by larger-sized voxels. Thereafter, we initialize the DNMP (with respectively scaled sizes) for each hierarchy and optimize their shapes with Eq. (2).

3.3. Radiance Modeling and View Synthesis

Rasterization and radiance feature interpolation. To render a pixel, we collect the related features with rasterization as shown in Fig. 2. The radiance features used for rendering, denoted as $\{\mathbf{f}_j | j = 1, 2 \ldots, J\}$, are essentially interpolated from the triangulated vertex features of the intersected mesh faces, where j indexes the intersections. Besides, we use the view-ray direction $\mathbf{d} \in \mathbb{R}^3$ and the surface normal $\mathbf{n}_j \in \mathbb{R}^3$ at the intersection point to model the view-dependent factor $\boldsymbol{\gamma}_j = \{\mathbf{n}_j, \mathbf{d}\}$.

Rendering. The radiance feature $\mathbf{f}_j$ as well as view-dependent factor $\boldsymbol{\gamma}_j$ is input to a MLP F_θ (shared among a sequence) after positional encoding [31]. F_θ will predict a pair of radiance value $\mathbf{c}_j$ and opacity α_j for each intersected point:

$$\mathbf{c}_j, \alpha_j = \mathrm{F}_\theta(\mathbf{f}_j, \boldsymbol{\gamma}_j). \tag{3}$$

The opacity α_j here represents the probability that the ray will terminate at the j-th point [3]. In our implementation, F_θ predicts the opacity α_j only based on $\mathbf{f}_j$. The view-dependent factor $\boldsymbol{\gamma}_j$ is only input to the branch split from near the end of the entire network to predict the view-dependent radiance value $\mathbf{c}_j$ following the spirit of [31].

We empirically keep J nearest intersections for each view ray and predict their radiance and opacity values, *i.e.*, $\{(\mathbf{c}_j, \alpha_j) | j = 1, \ldots, J\}$. Thereafter, to render the pixel of this view ray $\mathbf{r}$, the expected color is calculated as

$$\hat{\mathbf{C}}(\mathbf{r}) = \sum_j^J T_j \alpha_j \mathbf{c}_j, \quad T_j = \prod_{p=1}^{j-1}(1 - \alpha_p). \tag{4}$$

Blending hierarchical DNMPs. To improve the unsatisfactory rendering caused by the missing regions in point-cloud reconstruction, we need to blend the rendering results of DNMPs from different hierarchies. We denote the rendered colors from different hierarchies with Eq. (4) as $\{\hat{\mathbf{C}}_1(\mathbf{r}), \hat{\mathbf{C}}_2(\mathbf{r}), \ldots, \hat{\mathbf{C}}_S(\mathbf{r})\}$, where the subscripts denote the hierarchy levels. We practically blend these results from the finest level ($\hat{\mathbf{C}}_1(\mathbf{r})$) to the coarsest level ($\hat{\mathbf{C}}_S(\mathbf{r})$) to better keep the texture details:

$$\hat{\mathbf{C}}(\mathbf{r}) = \hat{\mathbf{C}}_1(\mathbf{r}) + (1 - \mathcal{A}_1)\hat{\mathbf{C}}_2(\mathbf{r}) + \cdots + \prod_{s=1}^{S-1}(1 - \mathcal{A}_s)\hat{\mathbf{C}}_S(\mathbf{r}), \tag{5}$$

where $\mathcal{A}_s$ is calculated as the summation of the accumulated weights of the respective hierarchy, *i.e.*, $\mathcal{A}_s = \sum_j^J T_j \alpha_j$, and $\mathcal{A}_s$ will be manually set to 0 if there is no view-ray intersection in this level.

Radiance feature learning. The DNMPs' vertex features are empirically initialized with the positional encoding [31] of the vertex coordinates before training. During training, we supervise the radiance feature learning with the camera images $\mathbf{C}$ in the training video sequence:

$$L_{rad} = \sum_{\mathbf{r} \in \mathcal{R}} ||\hat{\mathbf{C}}(\mathbf{r}) - \mathbf{C}(\mathbf{r})||_2^2, \tag{6}$$

where $\hat{\mathbf{C}}(\mathbf{r})$ and $\mathbf{C}(\mathbf{r})$ are the rendered color and the ground-truth color of view ray $\mathbf{r}$. This loss is applied for all the view rays $\mathcal{R}$ defined by the training image sequence.

Non-structured regions. Given that our method mentioned above is based on the explicit geometric abstraction of the scene, the non-structured regions (*e.g.*, sky) cannot be well handled. We use Mip-NeRF [4] to handle these regions in our implementation.

4. Experiments

4.1. Experimental Setup

Implementation details. For the hierarchical representation, we voxelize the point clouds with two hierarchical sizes, *i.e.*, 0.5m and 1m. During rendering, we blend radiance values of the nearest 4 intersection points for the 0.5m hierarchy, *i.e.*, J is set to 4. For the hierarchy with 1m DNMPs, J is set to 2. The frequency of positional encoding for vertex feature initialization is set to 3, resulting in 21-dimensional features. Please refer to supplementary materials for more details.

Datasets. We conduct experiments on two urban datasets, *i.e.*, KITTI-360 [26] and Waymo Open Dataset [49]. KITTI-360 is a large-scale dataset containing $4 \times 83,000$ images captured in urban environments with a driving distance of around 73.7 km. We select 5 sequences from them to evaluate our method. For the evaluation of novel view synthesis, we select every second image in each sequence as the test set and train our model on the remaining images.

Table 1. Ablation studies to show the effectiveness of our DNMP-based shape optimization on KITTI-360 dataset.

Method	direct shape optim.	w/o shape optim.	w/o hierarchy	Ours
PSNR↑	21.41	20.36	23.21	**23.41**
SSIM↑	0.789	0.758	0.840	**0.846**
LPIPS↓	0.397	0.421	0.322	**0.305**

Figure 4. View synthesis results with different strategies for primitive shape optimization. The results of *direct shape optim.* are slightly better than *w/o shape optim.*, but are still much less satisfactory compared with our DNMP-based shape optimization.

For Waymo Open Dataset, we follow [36] to select the 6 sequences mainly containing static objects for our experiments. We select every 10th image in the sequences as the test set and take the remaining ones as the training set. Details of datasets splits are provided in supplementary materials. To better evaluate and compare the synthesis capability for details, we train and evaluate our methods and all the baselines with full-resolution images (*i.e.*, 1408×376 for KITTI-360 and 1920×1280 for Waymo dataset).

Evaluation metrics. Following the previous methods [31, 4], our evaluations are based on three widely-used metrics, *i.e.*, peak signal-to-noise ratio (PSNR), structural similarity index measure (SSIM), and the learned perceptual image patch similarity (LPIPS) [60].

4.2. Ablation Study

We conduct thorough ablation studies on KITTI-360 dataset to analyze the effects of the proposed components on the task of novel view synthesis.

Shape optimization of DNMPs. We introduce two ablation variants to demonstrate the effectiveness of the proposed DNMP-based shape optimization. For the variant *direct shape optim.* in Table 1, instead of decoding the DNMP vertices from the latent code, we assign learnable vertex offset parameters to deform the spherical mesh templates and directly optimize these parameters using the same loss function L_{geo} (Eq. (2)). For the variant *w/o shape optim.*, we simply use the original spherical mesh templates without further shape optimization. According to the novel view synthesis performance in Table 1 and visualizations in Fig. 4, although *direct shape optim.* achieves better results than *w/o shape optim.*, it is still inferior to our DNMP-based shape optimization. We visualize and compare the optimized scene geometries in Fig. 5. Without the constraints on degree of freedom, the version *direct shape optim.* becomes sensitive to incomplete and noisy depth maps from MVS, which leads to noisy geometry recovery. The unsatisfactory scene geometry should explain the degradation in rendering quality. In contrast, our proposed DNMP-based shape optimization is much more stable, which leads to robust surface modeling and better rendering quality.

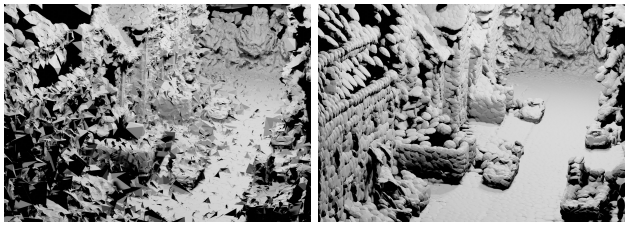

Figure 5. Visualization of the scene geometry optimized with different strategies. Compared with directly optimizing the vertex parameters, our DNMP-based shape optimization is more robust.

Table 2. Hyperparameter analysis based on KITTI-360 dataset.

Metric	# Intersection points J			Radiance feature dim.			DNMP radius			Light-weight
	2	4	8	15	21	27	2 m	1 m	0.5 m	
PSNR↑	23.23	23.41	**23.43**	23.21	**23.41**	23.28	22.61	23.20	**23.21**	23.27
SSIM↑	0.843	0.846	**0.847**	0.839	**0.846**	0.845	0.818	0.838	**0.840**	0.842
LPIPS↓	0.313	**0.305**	0.306	0.321	0.305	**0.301**	0.376	0.332	**0.322**	0.307

Rendering process. As we mentioned in Sec. 3.3, we collect the J nearest radiance features from the mesh faces intersected by the view ray and calculate the rendering color with Eq. (4). We conduct experiments with different intersection numbers J and the performance comparison is shown in Table 2 (in the column *# Intersection points* J). Our performance is relatively robust to the intersection number J and a small $J = 4$ is generally enough for good view synthesis performance due to our effective DNMP-based shape optimization. Besides, we also try with the radiance features of different dimensions (denoted as *Radiance feature dim.* in Table 2). It is shown that 21-dimensional feature is sufficient for good rendering quality.

Hierarchical DNMPs. We first analyze the performance with different radii of DNMPs. As shown in Table 2 (columns under *DNMP radius*), our performance is tolerant to large DNMP radii, which is beneficial for resource-constrained scenarios. To verify the necessity of the hierarchical representation in covering missing structures, we synthesize the images only with the DNMPs of the finest hierarchy, the performance of which is reported in Table 1 (denoted as *w/o hierarchy*). The performance is degraded. Examples shown in Fig. 6 demonstrate that the hierarchical DNMPs can effectively improve the completeness of the synthesized images.

Lightweight version. To better adapt to latency-sensitive

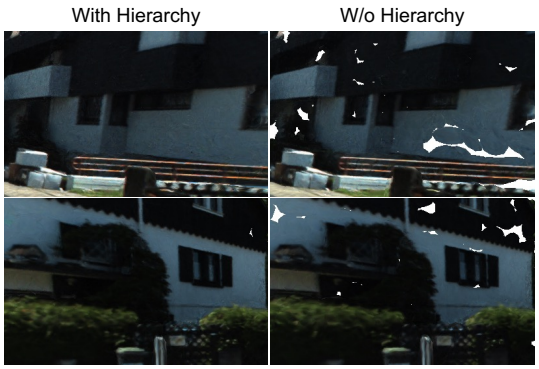

Figure 6. Rendering with or w/o hierarchical DNMPs. The proposed hierarchical DNMPs effectively completes the missing regions in the synthesized images. Better zoom in for more details.

systems, we provide a lightweight version by reducing the complexity of the MLP F_θ (Please refer to supplementary materials for more details). Due to the high capacity of our surface-aligned features, the lightweight version still achieves competitive performance as shown in Table 2.

4.3. Comparison with the State-of-the-Art Methods

We compare the proposed method with several competitive baselines on the task of novel view synthesis, including NeRF [31], NeRF-W [29], Mip-NeRF [4], Mip-NeRF 360 [5], Instant-NGP [32], and two point cloud-based neural rendering methods (*i.e.*, Point-NeRF [55] and NPLF (Neural Point Light Fields) [36]). The evaluation results on KITTI-360 dataset and Waymo dataset are shown in Table 3, 4, respectively. We also provide qualitative comparisons on Waymo dataset in Fig. 7 (Please refer to our supplementary materials for more visualization results). It is shown that our method performs consistently well on both datasets. Compared with these competitive baselines, our method achieves competitive novel view synthesis quality in terms of all the evaluation metrics. Especially, we achieve a lower LPIPS than other baselines. The metric LPIPS should be more consistent with the perception of our human beings [60]. The lower LPIPS indicates that our synthesized images should be more realistic and rich in details compared with the other methods.

We also find that, although Point-NeRF has achieved good performance in small-scale synthetic dataset, it may fail to render high-quality images in real outdoor environments due to the existence of complex occlusions and incomplete noisy point-cloud reconstructions. Moreover, the previous neural rendering method NPLF designed for outdoor scenarios is also found to be less effective to synthesize high-resolution images. We deem it could be caused by the lack of geometry optimization in NPLF. Thanks to the DNMP-based scene representation, the proposed method achieves photo-realistic rendering with rich texture details.

To further evaluate the view synthesis ability for the views that are significantly different from the training set, we rotate the original testing views of KITTI-360 dataset

Table 3. Performance comparison of novel view synthesis with other competitive baselines on KITTI-360 dataset.

Method	PSNR↑	SSIM↑	LPIPS↓
NeRF [31]	21.94	0.781	0.449
NeRF-W [29]	22.77	0.794	0.446
Instant-NGP [32]	22.89	0.836	0.353
Point-NeRF [55]	21.54	0.793	0.406
Mip-NeRF [4]	23.21	0.810	0.455
Mip-NeRF 360 [5]	23.27	0.836	0.355
Ours	**23.41**	**0.846**	**0.305**

Table 4. Performance comparison of novel view synthesis with other competitive baselines on Waymo dataset.

Method	PSNR↑	SSIM↑	LPIPS↓
NeRF [31]	26.24	0.870	0.472
NeRF-W [29]	26.92	0.885	0.418
Instant-NGP [32]	26.77	0.887	0.401
Point-NeRF [55]	26.26	0.868	0.450
NPLF [36]	25.62	0.879	0.450
Mip-NeRF [4]	26.96	0.880	0.451
Mip-NeRF 360 [5]	27.43	**0.893**	0.394
Ours	**27.62**	0.892	**0.381**

Table 5. View synthesis quality for the views that are significantly different from the training set. PSNR is adopted as the evaluation metric and the experiments are based on KITTI-360 dataset.

Method	Point-NeRF [55]	Mip-NeRF [4]	Ours
45° left rot	12.63	14.33	**15.25**
45° right rot	14.32	16.16	**17.09**

Table 6. Efficiency analysis.

Methods	# Network evaluations (/pixel)	Peak memory (GB/1K pixel)	Rendering time (ms/1K pixel)
NeRF [31]	256	0.80	20.24
Mip-NeRF 360 [5]	96	0.53	10.39
Point-NeRF [55]	40	2.06	32.44
NPLF [36]	**1**	0.10	20.08
Instant-NGP [32]	∼126	**0.02**	0.71
Ours	6	0.11	2.07
Ours (Lightweight)	6	0.03	**0.61**

to the left or the right by 45° for evaluation. The performance comparisons under this setting are shown in Table 5. Due to the explicit surface modeling with DNMPs, our method demonstrates relatively stronger robustness against the viewpoint changes compared with Mip-NeRF and Point-NeRF. According to the qualitative results in Fig. 8, our method can still output reasonable rendering results under 45° view rotations, while the counterparts have failed and generated severe blur and artifacts.

4.4. Efficiency Analysis

We analyze the efficiency of our system in terms of rendering time and memory consumption, and compare them with other typical methods. All the evaluations are based on an NVIDIA A100 GPU. The results are exhibited in Table 6. Given our explicit surface modeling, we only need the 6 nearest surface points intersected by the view ray (4 in the finest hierarchy and 2 in the coarser one) and evaluate the network on them, which significantly reduces the

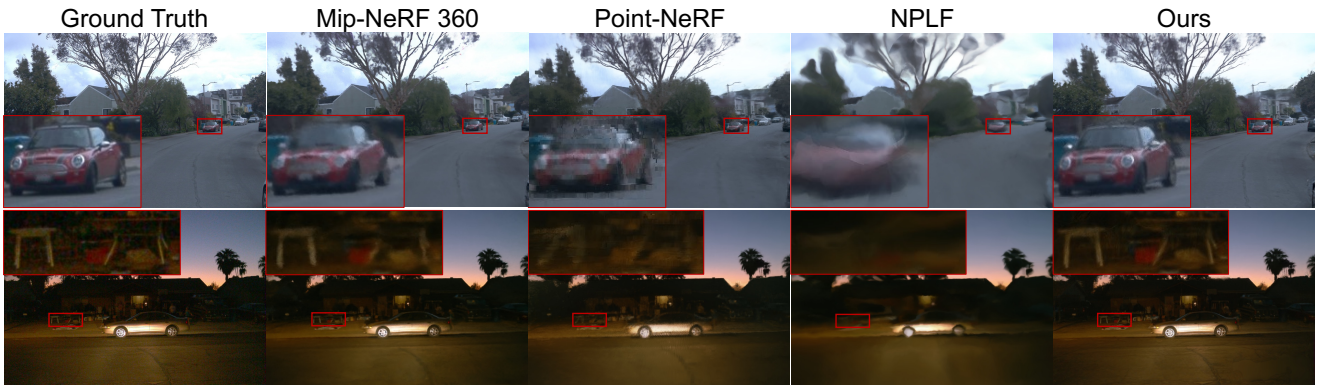

Figure 7. Qualitative comparison of novel view synthesis on Waymo dataset. Cropped patches are scaled to highlight the details. Due to the explicit and accurate surface modeling, the proposed method significantly outperforms baseline methods and effectively recover the texture details. Please refer to our supplementary materials for more visualization results.

Figure 8. Qualitative comparison of novel view synthesis with significant view differences from the training set. Compared to Mip-NeRF and Point-NeRF, the proposed method can produce high-quality results with much less blurs and artifacts. Better zoom in for more details.

runtime and GPU memory usage. Besides, we leverage rasterization for radiance feature interpolation rather than the inefficient k-NN algorithm adopted by Point-NeRF [55] and NPLF [36], which further improves the efficiency. Consequently, we achieve $5\times$ faster rendering speed with only $\sim\frac{1}{5}$ peak memory consumption compared with Mip-NeRF 360. Instant-NGP [32] is well-known for its fast inference speed based on the highly-optimized implementation. Although our system is naively implemented with PyTorch, we achieve an inference speed competitive with Instant-NGP thanks to the efficient rasterization-based rendering and the small network evaluation times. Besides, our lightweight version can achieve even faster speed while maintaining better rendering quality than Instant-NGP (as shown in Table 2, 3).

4.5. Scene Editing

Our explicit mesh representation naturally enables scene editing. We can achieve texture editing (Fig. 9 (a)) and object removal/insertion (Fig. 9 (b)) by locally modifying the radiance features of vertices and removing/inserting the DNMPs. Please refer to our supplementary materials for more visualization results.

Figure 9. Samples of scene editing. Please refer to our supplementary materials for more visualization results.

5. Conclusion

We have presented a novel neural scene representation for urban view synthesis based on the proposed Deformable Neural Mesh Primitives (DNMPs), which combines the efficiency of classic meshes and the representation capability of neural features. The entire scene is voxelized and each voxel is assigned a DNMP to parameterize the geometry and radiance of the local area. The shape of DNMP is decoded from a compact latent space to constrain the degree of freedom for robust shape optimization. The radiance features are associated to each mesh vertex of DNMPs for radiance information encoding. Extensive experiments on two public outdoor datasets have verified the effectiveness of the proposed components and demonstrated the state-of-the-art performance of the proposed method. Moreover, due to the compact and efficient mesh-based representation, we achieve a fast inference speed and much lower peak memory compared with previous methods.

However, although the remarkable rendering quality and resource efficiency have been achieved, the current version of our framework is still based on the static-scene assumption. In the future, we plan to extend our method to handle moving objects for more general application scenarios.

References

[1] Sameer Agarwal, Yasutaka Furukawa, Noah Snavely, Ian Simon, Brian Curless, Steven M Seitz, and Richard Szeliski. Building rome in a day. *Communications of the ACM*, 54(10):105–112, 2011. 3

[2] Kara-Ali Aliev, Artem Sevastopolsky, Maria Kolos, Dmitry Ulyanov, and Victor Lempitsky. Neural point-based graphics. In *European Conference on Computer Vision*, pages 696–712. Springer, 2020. 2

[3] Benjamin Attal, Jia-Bin Huang, Michael Zollhöfer, Johannes Kopf, and Changil Kim. Learning neural light fields with ray-space embedding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19819–19829, 2022. 5

[4] 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, 5, 6, 7

[5] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5470–5479, 2022. 3, 7

[6] T. Chan and Wei Zhu. Level set based shape prior segmentation. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 1164–1170 vol. 2, 2005. 2

[7] Zhiqin Chen, Thomas Funkhouser, Peter Hedman, and Andrea Tagliasacchi. Mobilenerf: Exploiting the polygon rasterization pipeline for efficient neural field rendering on mobile architectures. *arXiv preprint arXiv:2208.00277*, 2022. 3

[8] Shuo Cheng, Zexiang Xu, Shilin Zhu, Zhuwen Li, Li Erran Li, Ravi Ramamoorthi, and Hao Su. Deep stereo using adaptive thin volume representation with uncertainty awareness. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2524–2534, 2020. 3

[9] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised nerf: Fewer views and faster training for free. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12882–12891, 2022. 3

[10] Mathieu Desbrun, Mark Meyer, Peter Schröder, and Alan H Barr. Implicit fairing of irregular meshes using diffusion and curvature flow. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 317–324, 1999. 4

[11] Yueqi Duan, Haidong Zhu, He Wang, Li Yi, Ram Nevatia, and Leonidas J Guibas. Curriculum deepsdf. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VIII 16*, pages 51–67. Springer, 2020. 3

[12] Evelyn Fix and Joseph Lawson Hodges. Discriminatory analysis. nonparametric discrimination: Consistency properties. *International Statistical Review/Revue Internationale de Statistique*, 57(3):238–247, 1989. 4

[13] 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

[14] Xiao Fu, Shangzhan Zhang, Tianrun Chen, Yichong Lu, Lanyun Zhu, Xiaowei Zhou, Andreas Geiger, and Yiyi Liao. Panoptic nerf: 3d-to-2d label transfer for panoptic urban scene segmentation. In *International Conference on 3D Vision (3DV)*, 2022. 3

[15] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE transactions on pattern analysis and machine intelligence*, 32(8):1362–1376, 2009. 3

[16] Jun Gao, Wenzheng Chen, Tommy Xiang, Alec Jacobson, Morgan McGuire, and Sanja Fidler. Learning deformable tetrahedral meshes for 3d reconstruction. *Advances In Neural Information Processing Systems*, 33:9936–9947, 2020. 3

[17] 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

[18] Michael Goesele, Brian Curless, and Steven M Seitz. Multi-view stereo revisited. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 2402–2409. IEEE, 2006. 2

[19] 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

[20] Hugues Hoppe, Tony DeRose, Tom Duchamp, John McDonald, and Werner Stuetzle. Mesh optimization. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pages 19–26, 1993. 2

[21] Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew Davison, et al. Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 559–568, 2011. 2

[22] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, volume 7, 2006. 2

[23] Leif Kobbelt and Mario Botsch. A survey of point-based techniques in computer graphics. *Computers & Graphics*, 28(6):801–814, 2004. 2

[24] Mark A Kramer. Nonlinear principal component analysis using autoassociative neural networks. *AIChE journal*, 37(2):233–243, 1991. 3

[25] Zhuopeng Li, Lu Li, Zeyu Ma, Ping Zhang, Junbo Chen, and Jianke Zhu. Read: Large-scale neural scene rendering for autonomous driving. *arXiv preprint arXiv:2205.05509*, 2022. 2

[26] Yiyi Liao, Jun Xie, and Andreas Geiger. Kitti-360: A novel dataset and benchmarks for urban scene understanding in 2d

and 3d. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 2, 5

[27] 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

[28] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. *ACM siggraph computer graphics*, 21(4):163–169, 1987. 3

[29] Ricardo Martin-Brualla, Noha Radwan, Mehdi SM Sajjadi, Jonathan T Barron, Alexey Dosovitskiy, and Daniel Duckworth. Nerf in the wild: Neural radiance fields for unconstrained photo collections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7210–7219, 2021. 2, 3, 7

[30] Moustafa Meshry, Dan B Goldman, Sameh Khamis, Hugues Hoppe, Rohit Pandey, Noah Snavely, and Ricardo Martin-Brualla. Neural rerendering in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6878–6887, 2019. 2

[31] 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. 2, 3, 5, 6, 7

[32] 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. 2, 3, 7, 8

[33] 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. 3

[34] Andrew Nealen, Takeo Igarashi, Olga Sorkine, and Marc Alexa. Laplacian mesh optimization. In *Proceedings of the 4th international conference on Computer graphics and interactive techniques in Australasia and Southeast Asia*, pages 381–389, 2006. 4

[35] Thomas Neff, Pascal Stadlbauer, Mathias Parger, Andreas Kurz, Joerg H Mueller, Chakravarty R Alla Chaitanya, Anton Kaplanyan, and Markus Steinberger. Donerf: Towards real-time rendering of compact neural radiance fields using depth oracle networks. In *Computer Graphics Forum*, volume 40, pages 45–59. Wiley Online Library, 2021. 3

[36] Julian Ost, Issam Laradji, Alejandro Newell, Yuval Bahat, and Felix Heide. Neural point light fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18419–18429, 2022. 2, 3, 4, 6, 7, 8

[37] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, pages 523–540. Springer, 2020. 3

[38] Martin Piala and Ronald Clark. Terminerf: Ray termination prediction for efficient neural rendering. In *2021 International Conference on 3D Vision (3DV)*, pages 1106–1114. IEEE, 2021. 3

[39] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10318–10327, 2021. 2, 3

[40] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 3

[41] 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

[42] Konstantinos Rematas, Andrew Liu, Pratul P Srinivasan, Jonathan T Barron, Andrea Tagliasacchi, Thomas Funkhouser, and Vittorio Ferrari. Urban radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12932–12942, 2022. 3

[43] Gernot Riegler and Vladlen Koltun. Stable view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12216–12225, 2021. 2

[44] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 3

[45] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, 2016. 2, 3, 4, 5

[46] Michael Schwarz and Hans-Peter Seidel. Fast parallel surface and solid voxelization on gpus. *ACM transactions on graphics (TOG)*, 29(6):1–10, 2010. 2

[47] Tianchang Shen, Jun Gao, Kangxue Yin, Ming-Yu Liu, and Sanja Fidler. Deep marching tetrahedra: a hybrid representation for high-resolution 3d shape synthesis. *Advances in Neural Information Processing Systems*, 34:6087–6101, 2021. 3

[48] 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

[49] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2446–2454, 2020. 2, 5

[50] Matthew Tancik, Vincent Casser, Xinchen Yan, Sabeek Pradhan, Ben Mildenhall, Pratul P Srinivasan, Jonathan T Barron, and Henrik Kretzschmar. Block-nerf: Scalable large scene neural view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8248–8258, 2022. 2, 3

[51] Fangjinhua Wang, Silvano Galliani, Christoph Vogel, Pablo Speciale, and Marc Pollefeys. Patchmatchnet: Learned multi-view patchmatch stereo. In *Proceedings of the*

IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 14194–14203, 2021. 3

[52] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2mesh: Generating 3d mesh models from single rgb images. In *Proceedings of the European conference on computer vision (ECCV)*, pages 52–67, 2018. 3

[53] 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. 3

[54] Zirui Wang, Shangzhe Wu, Weidi Xie, Min Chen, and Victor Adrian Prisacariu. Nerf–: Neural radiance fields without known camera parameters. *arXiv preprint arXiv:2102.07064*, 2021. 2, 3

[55] 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, 3, 4, 7, 8

[56] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. Mvsnet: Depth inference for unstructured multi-view stereo. In *Proceedings of the European conference on computer vision (ECCV)*, pages 767–783, 2018. 3

[57] 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

[58] Zehao Yu, Songyou Peng, Michael Niemeyer, Torsten Sattler, and Andreas Geiger. Monosdf: Exploring monocular geometric cues for neural implicit surface reconstruction. *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 3

[59] Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. Nerf++: Analyzing and improving neural radiance fields. *arXiv preprint arXiv:2010.07492*, 2020. 2, 3

[60] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 6, 7