Transactions in GIS

WILEY

Transactions in GIS

RESEARCH ARTICLE

LCA-NeRF: Large-Scale Scene Neural Rendering With Clustered Appearance Embedding

Ziyu Zhang[1] | Xiaojun Xiang[2] | Hanqing Jiang[2] | Shuhan Shen[1]

[1]Institute of Automation, Chinese Academy of Sciences, Beijing, China | [2]SenseTime Research, Hangzhou, China

Correspondence: Shuhan Shen (shshen@nlpr.ia.ac.cn)

Received: 16 December 2024 | **Revised:** 12 June 2025 | **Accepted:** 25 July 2025

Funding: This work was supported by the National Natural Science Foundation of China (no. U22B2055 and 62273345), the Beijing Natural Science Foundation (no. L223003), and the Key R&D Project in Henan Province (no. 231111210300).

ABSTRACT

Recently, Neural Radiance Fields (NeRF) have been used for urban-scale scenes with potentially infinite scales. Prior work often necessitates training times of several tens of hours in large-scale scenes. Although fast-converging NeRF variants have been applied to urban-scale scenes, they struggle to capture sufficient details and prove challenging in handling dramatic lighting variations in practical usage. It could be argued that these limitations arise from two key factors: limited model capacity and insufficient attention to the geometric distribution of nonlinear light sources within the scene. To address these challenges, we propose a novel approach for modeling large-scale scenes. Building upon the foundation of fast-converging NeRF variants, we incorporate a block-based strategy to reduce training costs and capture additional scene details. Furthermore, we introduce clustered appearance embeddings to model the nonlinear lighting present in space, enabling smoother transitions in radiance during practical use. We evaluated our approach against other methods for large scenes on the Mill19 and Urbanscene3D datasets, surpassing the state-of-the-art methods and converging within a few hours.

1 | Introduction

Recently a myriad of studies in Neural Rendering have garnered significant attention. Specifically, the Neural Radiance Fields (NeRF) (Mildenhall et al. 2021) utilize a straightforward Multi-Layer Perceptron (MLP) to encode intricate 3D scenes, abstracting them into a probabilistic field. Utilizing volumetric rendering methods, it ultimately results in the generation of high-fidelity novel perspectives.

Early NeRF-related works primarily focused on object-centric reconstruction at a small scale. As the scale of the scene expands, particularly in urban-level environments characterized by substantial variations in lighting conditions, the capacity of the neural network and its capability to adapt to diverse lighting scenarios present novel challenges. Since increasing network depth or width to enhance capacity can significantly escalate both training and inference costs, most existing methods for large-scale scenes commonly partition the scene into foreground and background, subdivide the foreground into multiple sub-blocks, and then employ a divide-and-conquer strategy to train each sub-block independently (Rebain et al. 2021; Tancik et al. 2022; Turki et al. 2022, 2023; Zhenxing and Xu 2022). To further improve the rendering quality under complex lighting conditions in large-scale scenes, image-dependent appearance embeddings are introduced to vary the emitted radiance of the scene in image (Martin-Brualla et al. 2021; Tancik et al. 2022; Turki et al. 2022; Zhenxing and Xu 2022). Furthermore, in an effort to address the training costs associated with large-scale scenes, researchers have introduced hash grid (Müller et al. 2022) acceleration techniques. These strategies have demonstrated their effectiveness by improving the convergence speed during training, making it a more efficient process for reconstructing large-scale scenes (Turki et al. 2023; Zhang et al. 2023).

However, when it comes to large-scale scenes, particularly expansive outdoor environments with intricate lighting conditions, existing methods still exhibit limitations in terms of their robustness to lighting variations and their ability to capture fine details. We think that this mainly comes from two aspects. First, existing methods often overlook the correlations between embeddings when using appearance embeddings, leading to abrupt lighting changes when rendering consecutive frames with the latest embeddings. Second, the prevalent use of MLPs in current block-based methods for scene representation makes it challenging to capture fine details in outdoor large-scale scenes. Conversely, methods employing hash grids (Reiser et al. 2023; Zhang et al. 2023), which often avoid block-based strategies, may experience slower convergence as the number of images increases. To address the above issues, we propose a large-scale scene neural rendering pipeline with clustered appearance embedding, termed as LCA-NeRF, which considers blocking strategy, training efficiency, and robustness to nonlinear lighting variations. In LCA-NeRF, we follow Mega-NeRF (Turki et al. 2022) and related methods (Tancik et al. 2022; Turki et al. 2023) for spatial partitioning to address the issue of large-scale scenes, and employ NGP (Müller et al. 2022) as the representation for each sub-model to accelerate convergence. Figure 1 clearly demonstrates that our LCA-NeRF can render urban-scale scenes with remarkable realism and exhibits superior clarity in details compared to other block-based state-of-the-art methods. For robustness to lighting conditions, we posit that appearance embeddings fundamentally capture the nonlinear lighting of the scene, often related to the geometric positions of light sources. When the scene is observed from different viewpoints, the radiance conditions for the current perspective can be inferred based on geometric relationships. Building upon this insight, in LCA-NeRF we propose appearance embeddings based on spatial clustering, which perform well in large outdoor scenes with significant lighting variations.

To summarize, our contributions are as follows:

- We propose spatially clustered appearance embeddings, allowing us to address scenes with significant lighting variations without the need for training on the test dataset.
- We propose a block-based training framework using hash grids for large-scale radiance field reconstruction, offering improved scalability and convergence speed.
- Building upon these two key points, we developed a large-scale scene neural rendering pipeline, termed LCA-NeRF, that surpasses current state-of-the-art block-based methods in both speed and rendering quality.

2 | Related Work

2.1 | Multi-View 3D Reconstruction

3D reconstruction is a classic task in computer vision, which involves estimating the three-dimensional structure of objects or scenes from a set of two-dimensional images. Typically, structure from motion (SfM) is required to obtain sparse 3D points of the object and the camera poses (Moulon et al. 2016; Schönberger and Frahm 2016). Subsequently, multi-view stereo (MVS) is employed to derive the object's three-dimensional representation, such as meshes or dense point clouds (Schönberger et al. 2016; Xu et al. 2022). Moreover, P-MVSNet (Luo et al. 2019) introduces a learning-based multi-view stereo framework that leverages both isotropic and anisotropic 3D convolutions for estimating a dense depth map. Geo-NeuS (Fu et al. 2022) addresses the lack of explicit multi-view geometry constraints in volume-rendered neural implicit surface learning. MR-LSD (Guo et al. 2022) proposes a multi-resolution line segment detector for 2D extraction, reconstructs a 3D line cloud via an improved Line3D++ matching, and generates a manifold surface mesh through Bayesian plane fitting.

2.2 | Neural Radiance Field

Since the advent of neural rendering and implicit representation (Eslami et al. 2018) in 2018, various studies employing neural networks for scene representation have made continuous breakthroughs in areas like 3D reconstruction and novel view synthesis (Jiang et al. 2020; Park et al. 2019; Sitzmann et al. 2019; Yariv et al. 2020). Specifically, NeRF (Mildenhall et al. 2021) and its related variants have made significant progress in fields, such as multi-view reconstruction (Fu et al. 2022; Li, Müller, et al. 2023; Long et al. 2023; Wang et al. 2021; Yariv

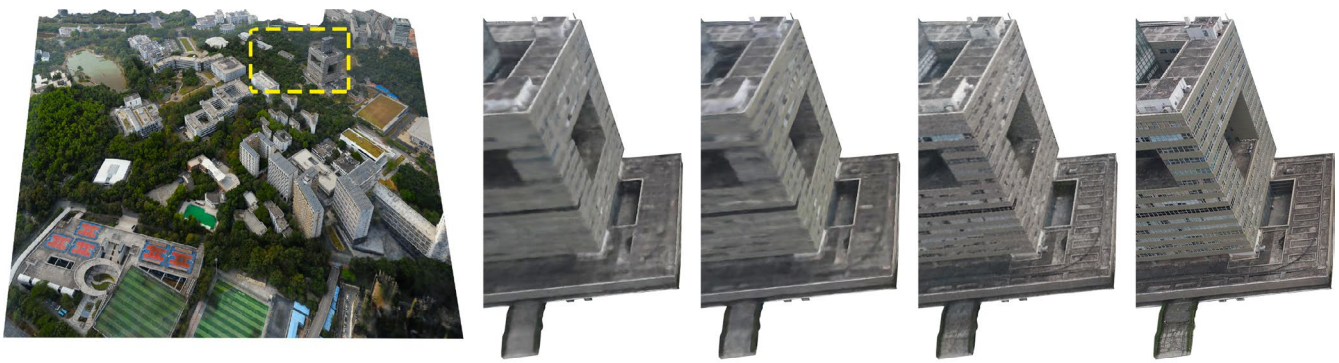

FIGURE 1 | In the Campus dataset (left), comprising over 5000 images and spanning a ground area of 2.07 km^2, LCA-NeRF shows a notable improvement in terms of rendering quality compared to state-of-the-art block-based methods.

et al. 2021), novel view synthesis (Barron et al. 2021, 2022; Chen et al. 2022; Reiser et al. 2023), and autonomous driving (Li, Li, and Zhu 2023; Meuleman et al. 2023; Yang et al. 2023).

2.3 | Fast Training

The vanilla NeRF typically requires dozens of hours to converge, and variants based on MLP cannot achieve order-of-magnitude improvements in training time. Subsequent acceleration techniques mainly focus on scene representation and sampling methods. The former transfers information from the MLP to more accessible discrete data structures, such as feature grid (Müller et al. 2022; Sun et al. 2022a, 2022b) and feature plane (Chan et al. 2022; Fridovich-Keil et al. 2023), to expedite inference speed. In contrast to the two-stage sampling in vanilla NeRF, the latter methods employ the ray marching algorithm to quickly skip over free space, thereby achieving acceleration (Müller et al. 2022; Reiser et al. 2023).

2.4 | Urban-Scale Reconstruction

Currently, many methods have achieved commendable results in urban-scale scenarios. Guo et al. (2024) segment urban buildings from aerial images and fuse multi-view instances into a sparse point cloud using a voting-based approach. Lin et al. (2024), Liu, Luo, Fan, et al. (2024), and Liu, Luo, Mao, et al. (2024) divide city-scale scenes into blocks and apply 3DGS (Kerbl et al. 2023) or 2DGS (Huang et al. 2024) for parallel training, enabling high-quality rendering and surface reconstruction. For NeRF-like methods, Mega-NeRF (Turki et al. 2022) and Block-NeRF (Tancik et al. 2022) spatially decompose urban-scale scenes into several sub-blocks. Each sub-block is allocated its dataset based on visibility or spatial clustering, allowing for parallel training. Switch-NeRF (Zhenxing and Xu 2022) employs a gating network to allocate 3D points to different sub-blocks. This gating network can be optimized alongside the sub-NeRFs for various scene partitions.

As the scale of the scene increases, the number of images used for radiance field reconstruction also grows, leading to a slower convergence rate. This issue is particularly prominent in MLP-based methods. Some approaches have adopted the aforementioned hybrid representations to address this issue. Building on the foundation of Mega-NeRF, GP-NeRF (Zhang et al. 2023) employs a hash grid (Müller et al. 2022) and feature planes (Chan et al. 2022) to represent scenes, achieving a notable improvement in training speed by an order of magnitude. Grid-NeRF (Xu et al. 2023), on the other hand, utilizes tensor fields for scene representation, achieving a compact representation and faster training while enhancing the quality of the rendered output.

In urban-scale scenarios, the space outside the modeling region is typically considered as an unbounded background area. Several methods (Barron et al. 2022; Reiser et al. 2023; Zhang et al. 2020) for unbounded scenes compress larger distant spatial regions into smaller areas within the space using transformation functions. This allows the model to represent boundless scenes with reduced capacity. In contrast to these approaches, our LCA-NeRF primarily emphasizes the expansion of appearance embeddings to enable reasoning based on geometric relationships.

3 | Method

In Figure 2, we present the pipeline of our methodology, which primarily includes space and data partitioning as introduced in Section 3.2, spatial weighting of appearance embeddings as discussed in Section 3.3, sub-model training encompassed in Sections 3.4 and 3.5.

3.1 | Preliminaries

3.1.1 | Neural Radiance Fields

Given a set of images with known extrinsic parameters $\{(R_1, t_1), \ldots, (R_L, t_L)\}$, NeRF-like methods represent the scene as a probability field into a MLP. Specifically, taking a 3D point $\mathbf{x}_i = (x, y, z)$ and the view direction $\mathbf{d}_i = (d_x, d_y, d_z)$ generated from the ray emitted from camera $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$ as input, the NeRF MLP outputs the volume density σ_i and color $\mathbf{c}_i$ correspond to the input:

$$(\sigma_i, \mathbf{c}_i) = \mathrm{MLP}(\mathbf{x}_i, \mathbf{d}_i) \tag{1}$$

Following this, volume rendering is employed to aggregate the contributions of these sample points, yielding the pixel color. Essentially, this computation is equivalent to determining the expected values of color (Mildenhall et al. 2021).

$$\begin{gathered} \hat{C}(\mathbf{r}) = \sum_{i=0}^{N-1} \omega_i \mathbf{c}_i, \quad \hat{D}(\mathbf{r}) = \sum_{i=0}^{N-1} \omega_i t_i \\ \text{where } \omega_i = T_i \alpha_i, \ T_i = \prod_{j=0}^{j=i-1} 1 - \alpha_j \end{gathered} \tag{2}$$

where $\alpha_i = 1 - \exp(-\sigma_i \delta_i)$ denotes the opacity of $\mathbf{x}_i$ from the direction specified by $\mathbf{d}$, and $\delta_i = t_{i+1} - t_i$ denotes the distance between adjacent sample points.

3.1.2 | Multi-Resolution Hashgrid

In implicit representations like NeRF, a large MLP model is needed (e.g., nine layers with 256 channels), increasing query costs during sampling (Barron et al. 2022; Fu et al. 2022; Mildenhall et al. 2021; Wang et al. 2021). Moreover, since the MLP is approximating the volumetric density function, as the scale of the scene grows and details proliferate, the convergence rate of the MLP noticeably decelerates.

Some approaches opt to offload part of the scene information from the MLP to more accessible explicit data structures $\mathcal{G}$ (e.g., a feature grid or feature plane). This strategy not only reduces the size of the MLP but also supersedes the time-consuming MLP computations with fast interpolation operations, thereby

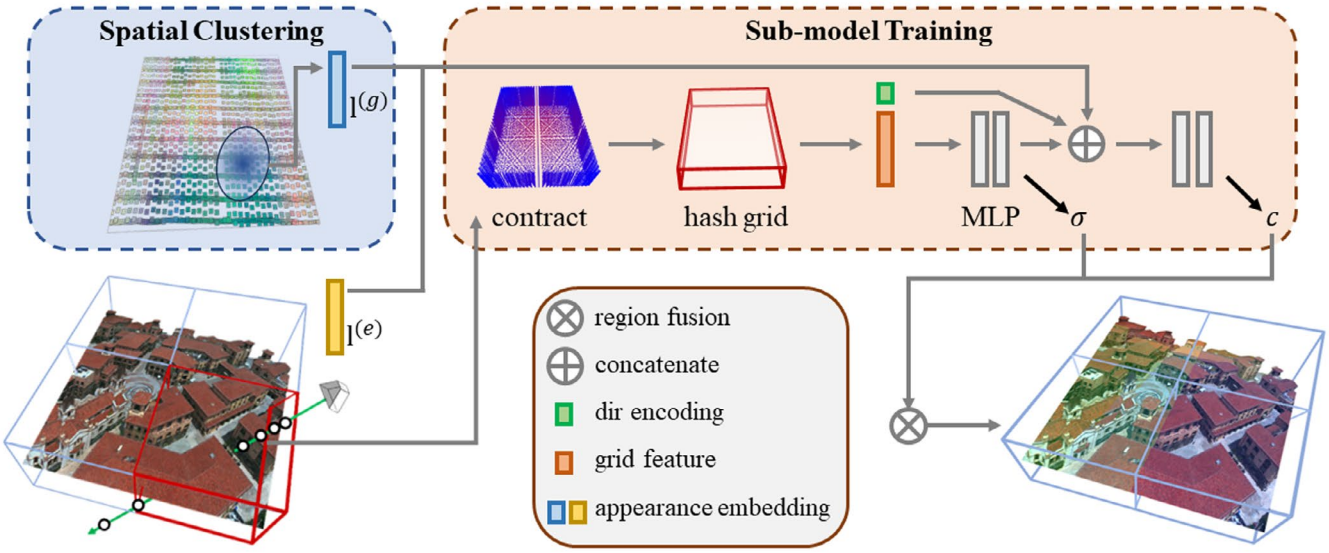

FIGURE 2 | Overview of LCA-NeRF. We initially divide the scene into several sub-models based on their geometric properties, with each sub-model occupying a rectangular spatial region and undergoing independent training. Ultimately, the output of each sub-model is fused using Inverse Distance Weights (IDW). Our key contribution lies in the use of spatial clustering-based appearance embeddings $\mathbf{l}^{(g)}$, which establish a correlation between appearance embeddings and geometric properties. Concurrently, employing $\mathbf{l}^{(e)}$ to depict the camera's intrinsic exposure.

enhancing the query speed (Chan et al. 2022; Chen et al. 2022; Liu et al. 2020; Müller et al. 2022; Sun et al. 2022a):

$$\begin{aligned} \mathbf{f}_i &= \text{interp}(\mathcal{G}, \mathbf{x}_i), \\ (\sigma_i, \mathbf{c}_i) &= \text{MLP}_{\text{shallow}}(\mathbf{f}_i, \mathbf{d}_i) \end{aligned} \tag{3}$$

Owing to the weak inter-element correlations in discrete representations, they tend to converge more quickly. Additionally, with an increase in the number of parameters, their representational capability is enhanced. Following instant-NGP (Müller et al. 2022), we employ an L-level feature grid to expedite our training process.

3.2 | Space and Data Partitioning

To address large-scale urban scenes with potentially infinite scale, we follow the block-based approach of Mega-NeRF (Turki et al. 2022), which involves decomposing the scene into multiple 2D grids. This is particularly effective in our case, as the variance in altitude among camera poses in our scenes is small compared to the variations in latitude and longitude.

$$\mathbf{O}_i, \mathbf{S}_i \in \mathbb{R}^3 \quad i = 1, \ \dots \ , N_{block} \tag{4}$$

where $\mathbf{O}_i$ and $\mathbf{S}_i$ denotes the centroid and scale of the sub bounding box. Additionally, we apply PCA to the sparse point cloud of SfM to extract its two principal axes and align them with the coordinate axes x and y. This is generally reasonable for large-scale scenes, as they tend to be relatively flat.

Mega-NeRF (Turki et al. 2022) and Switch-NeRF (Zhenxing and Xu 2022) adopt pixel-level visible partitioning to ensure each submodule only trains on potentially relevant pixels. However, this process is time-consuming and often requires tens of GB of storage. To simplify this step, we use an image-level partitioning strategy. Specifically, given the segmented blocks $(\mathbf{O}_i, \mathbf{S}_i)$ and camera parameters, we generate rays $\{r_{uv}\}_{u=1\dots W}^{v=1\dots H}$ for all pixels in image I_j. If a ray intersects the block at a bounded point, its mask is set to True. If the number of True masks exceeds $1/N_{block}$ of the image size, the image is assigned to the block's training set.

Interestingly, we observe that for methods using feature grids or planes, the difference between pixel-level and image-level partitioning is minimal, due to the strong fitting capacity of the hash grid. In contrast to pixel-level segmentation, which can take hours and consume tens of GB, our method only computes the intersection between each pixel and the block, completing partitioning in minutes and requiring only tens of MB.

3.3 | Spatial Clustering of Appearance Embeddings

In the context of outdoor large-scale scenes, it is essential to address the challenges associated with dynamic lighting variations. In NeRF-W (Martin-Brualla et al. 2021), appearance embeddings (AE) are exclusively used as inputs for the color network, providing the model the capability to adjust the radiance of the scene in specific images flexibly. AE embeds illumination conditions into a continuous space, enabling smooth interpolation across different lighting environments. From this perspective, AE may correlate with geometry, as neighboring viewpoints in large-scale, continuous captures typically share similar illumination conditions.

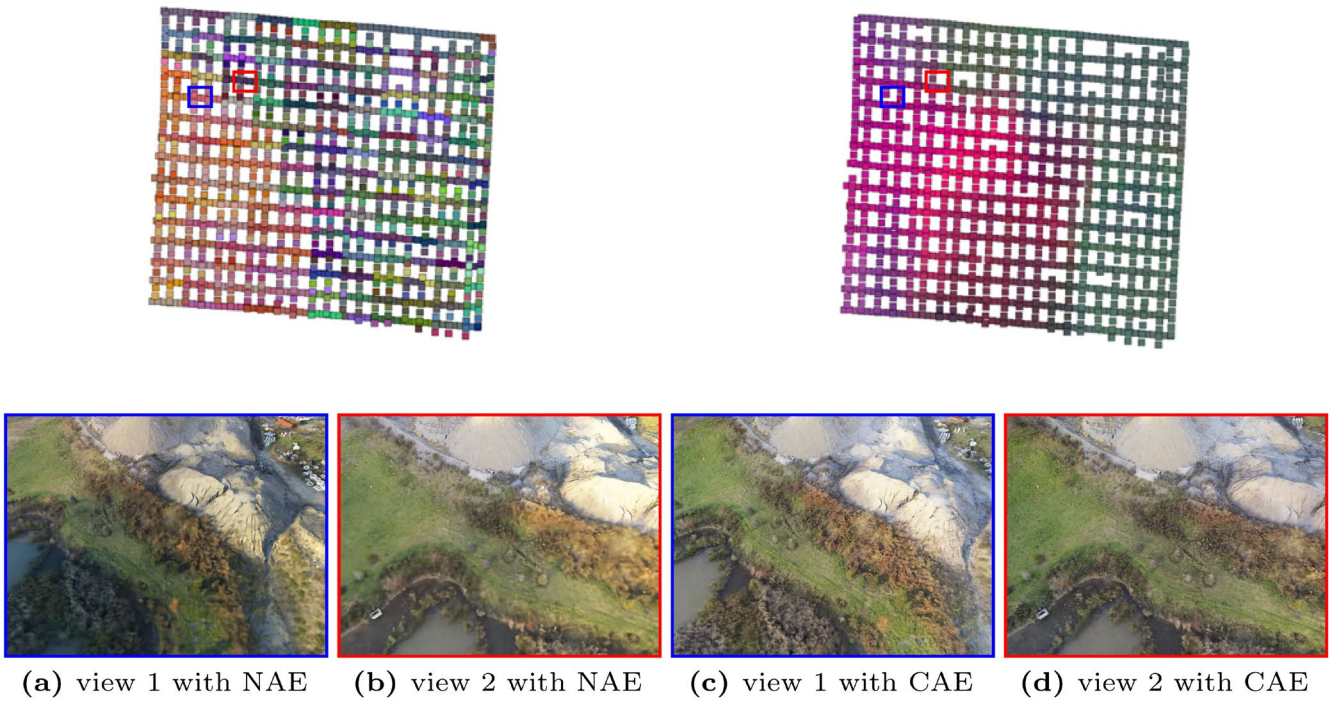

FIGURE 3 | The comparison between using nearest (NAE) and clustered (CAE) appearance embeddings. The top row shows the visualization of AEs for a given scene, with the left image corresponding to NAE and the right to CAE. The second row presents rendering results using AEs extracted from the highlighted regions (blue and red boxes). Training with NAE results in local inconsistencies, leading to noticeable lighting differences between (a) and (b). In contrast, CAE benefits from spatial clustering, which improves local consistency and enables smoother transitions between (c) and (d).

However, NeRF-W and following methods (Martin-Brualla et al. 2021; Turki et al. 2022; Zhang et al. 2023; Zhenxing and Xu 2022) link AE with image indices, causing AE's distribution to lack geometric correlations. This results in a large variance in the distribution of AE across different cameras, making it challenging to reasonably infer the lighting conditions of new viewpoints during usage, as shown in Figure 3. Consequently, it generally necessitates individual optimization of embeddings for new images. Particularly, in block-based methods, additional processing is required. Block-NeRF (Tancik et al. 2022) introduces a strategy of matching similar AEs across different blocks and freezing the network weights while solely optimizing the AEs. Furthermore, Mega-NeRF (Turki et al. 2022), GP-NeRF (Zhang et al. 2023), and Switch-NeRF (Zhenxing and Xu 2022) require training on half of the pixels of the test set images to optimize embeddings for test images, which can be observed in their code and issues. In light of this, we introduce spatially clustered appearance embeddings (CAE), allowing them to manifest clustering characteristics in space.

Specifically, for a given set of N calibrated images, we maintain an embedding array $\mathbf{L}_e = \left\{\mathbf{l}_i^{(e)}\right\}_{i=1,\dots,N}$ characterizing the appearance of image exposure, along with an appearance embedding array $\mathbf{L}_g$ associated with the camera poses $\mathbf{P} = \left\{P_i, \mathbf{l}_i^{(g)}\right\}_{i=1,\dots,N}$. For any given image J that requires rendering with its pose denoted as $P_j = [R_j|\mathbf{t}_j]$, the appearance embedding for image J is computed as the sum of the camera's own exposure $\mathbf{l}_i^{(e)}$ and the local illumination $\mathbf{l}_i^{(g)}$, where the weight of the local illumination is determined by the angular and positional deviations.

$$\begin{aligned} \sigma &= \mathrm{MLP}^n(\mathbf{x}) \\ c &= \mathrm{MLP}^n(\mathbf{x}, \mathbf{d}, \mathbf{l}_j) \\ \text{where } n &= \left\{i : \|\mathbf{x} - \mathbf{O}_i\| < \|\mathbf{S}_i\|\right\}, \\ \mathbf{l}_j &= \mathbf{l}_j^{(e)} + \sum_{i \in S_k(j)} \omega\left(S_k(j)_i\right) \mathbf{l}_i^{(g)} \end{aligned} \tag{5}$$

Here, $S_k(j)$ represents a subset of the set $\mathbf{P}$ that is related to the pose P_j, $\omega(\cdot)$ represents the reciprocal weight. We first compute the distance among viewpoints using both rotation and position errors, and then select the top-k closest viewpoints to form $S_k(j)$, as shown in Equation (6). In large-scale scenes, the changes in observed content due to pose displacement are relatively small compared to changes in perspective. Therefore, we have introduced a hyperparameter λ_{RT} to adjust the weights between the spatial neighborhood and the viewpoint neighborhood.

$$\begin{aligned} S_k(j) &= \arg \min_k \left\{\mathrm{d}\left(P_j, P_n\right)\right\}_{n=1,\dots,N} \\ \mathrm{d}\left(P_j, P_n\right) &= \lambda_{\mathrm{RT}} \arccos\left(\frac{\mathrm{tr}\left(R_j R_n^T\right) - 1}{2}\right) \\ &\quad + \|\mathbf{t}_j - \mathbf{t}_n\| \end{aligned} \tag{6}$$

We adopt this strategy not only during validation but also throughout the training phase. This approach ensures a more

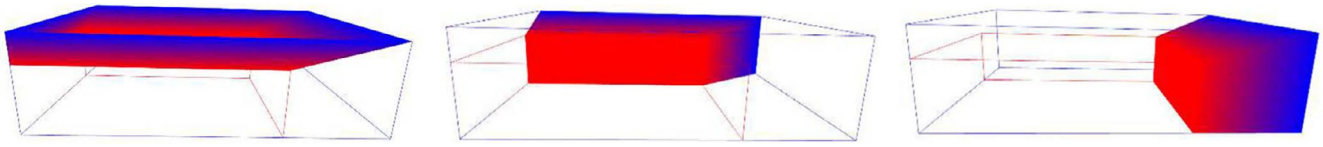

FIGURE 4 | Contraction region illustration. Background contraction maps unbounded points in the background into a bounded region. Red areas indicate weaker contraction, resulting in more loosen mapped points; blue areas indicate stronger contraction, resulting in more compact point distributions.

concentrated spatial distribution of our appearance embeddings, resulting in smoother transitions across various views, as shown in Figure 3. Per-image appearance embeddings lack consistent distribution within geometric neighborhoods, whereas spatially clustered appearance embeddings demonstrate enhanced local consistency.

3.4 | Anisotropic Background Contraction

In urban-scale scenes, there always exist vantage points where the camera perceives distant, potentially infinite regions. A high dynamic range in depth can significantly compromise the resolution in representations such as MLP or feature grid. Therefore, parameterizing the unbounded background becomes imperative. Typically, large-scale methods based on MLP use L_2 norm contraction function for background parameterization (Barron et al. 2022). In contrast, hash grid-based methods (Reiser et al. 2023), as introduced by MeRF, employ piecewise-projective norm contraction function to facilitate faster sampling.

However, the aforementioned background contraction method is isotropic. Considering that in urban-scale scenes, the horizontal distribution is considerably wider than the vertical distribution, applying the same contraction to both horizontal and vertical directions results in overly compacted points postcontraction, which is not conducive to the trilinear interpolation of the hash grid. To address this, we have implemented a segmented approach, applying varying degrees of contraction to the horizontal and vertical directions respectively, to achieve anisotropic contraction. We begin by presenting the infinity norm version of Mip-NeRF 360's contraction (Barron et al. 2022):

$$\mathbf{x}=\begin{cases}\mathbf{x}, & \|\mathbf{x}\|_p \le 1\\ \left(1+b-\frac{b}{\|\mathbf{x}\|_p}\right)\left(\frac{\mathbf{x}}{\|\mathbf{x}\|_p}\right), & \|\mathbf{x}\|_p > 1\end{cases}, \quad \text{where } p=+\infty,\ b=2 \tag{7}$$

In Equation (7), the background region (where the infinity norm is greater than 1) is contracted to a bounded cubic shell region. This contraction is uniform along all three axes because the scales of foreground and background are set to 1 and 2 along each axis, respectively. Furthermore, we aim to control the scale of each block within large-scale scenes. Therefore, we first generalize Equation (7) to arbitrary scales for both foreground and background:

$$\mathbf{x}=\begin{cases}\mathbf{x}, & \|\mathbf{x}\|_\infty \le b_{\text{fg}}\\ \left(b_{\text{bg}}-\frac{b_{\text{fg}}}{\|\mathbf{x}\|_\infty}\left(b_{\text{bg}}-b_{\text{fg}}\right)\right)\frac{\mathbf{x}}{\|\mathbf{x}\|_\infty}, & \|\mathbf{x}\|_\infty > b_{\text{fg}}\end{cases} \tag{8}$$

where $b_{\text{fg}}, b_{\text{bg}} \in \mathbb{R}$ denote the scales of foreground and background. Considering the limited height variations in urban-scale scenes compared to latitude and longitude, we opt for a flatter rectangular prism to better fit the actual scene. To achieve this, we divided the background region (cubic spherical shell) into six truncated pyramids. Due to symmetry, these six truncated pyramids can be represented using only the three truncated pyramids, as shown in Figure 4.

For each truncated cone, we perform contraction according to Equation (8), resulting in our final Equation (9). Where $\mathbf{b}_{\text{fg}}, \mathbf{b}_{\text{bg}} \in \mathbb{R}^3$ represent the scales of foreground and background AABB. And $\mathbf{b}_{\text{fg}}^i(\mathbf{b}_{\text{bg}}^i)$ denotes the i-th element of $\mathbf{b}_{\text{fg}}(\mathbf{b}_{\text{bg}})$. By utilizing the contraction function in Equation (9), we achieve a tighter compression of the background content.

3.5 | Optimization

After defining our clustered appearance embeddings and anisotropic background contraction, we will now delve into our sampling strategy followed by our optimization techniques.

3.5.1 | Point Sampling

We have incorporated ray parameterization inspired by Mip-NeRF 360 (Barron et al. 2022) and adopted the hierarchical sampling approach introduced by NeRF (Mildenhall et al. 2021) into our work. Additionally, we've implemented an AABB to distinguish between foreground and background sampling. Specifically, we sample $N_f = 128$ points in the foreground and $N_b = 64$ points in the background, while restricting the background sampling points to a bounded region. In contrast to Mega-NeRF and similar approaches (Turki et al. 2022; Zhang et al. 2023; Zhenxing and Xu 2022), we employ a single hash grid to represent both foreground and background concurrently, which conserves computational resources.

3.5.2 | Loss Function

We utilize Equation (2) for volumetric rendering to obtain the predicted pixel color. The Mean Squared Error (MSE) between the predicted color $\hat{C}(\mathbf{r}$ and the ground truth color $C(\mathbf{r}))$ serves as our RGB loss function.

$$\text{warp}(x,y,z) = \begin{cases} [x,y,z]^T & \text{if } |x| \le \mathbf{b}^1_{\text{fg}},\ |y| \le \mathbf{b}^2_{\text{fg}},\ |z| < \mathbf{b}^3_{\text{fg}} \\ \begin{bmatrix} \left(\mathbf{b}^1_{\text{bg}} - \frac{\mathbf{b}^1_{\text{fg}}}{|x|}\left(\mathbf{b}^1_{\text{bg}} - \mathbf{b}^1_{\text{fg}}\right)\right)\frac{x}{|x|} \\ \left(\mathbf{b}^2_{\text{bg}} - \frac{\mathbf{b}^2_{\text{fg}}}{|x|/\alpha}\left(\mathbf{b}^2_{\text{bg}} - \mathbf{b}^2_{\text{fg}}\right)\right)\frac{y}{|x|/\alpha} \\ \left(\mathbf{b}^3_{\text{bg}} - \frac{\mathbf{b}^3_{\text{fg}}}{|x|/\gamma}\left(\mathbf{b}^3_{\text{bg}} - \mathbf{b}^3_{\text{fg}}\right)\right)\frac{z}{|x|/\gamma} \end{bmatrix} & \text{if } |x| > \mathbf{b}^1_{\text{fg}},\ \frac{|x|}{\alpha} \ge |y|,\ \frac{|x|}{\gamma} \ge |z| \\ \begin{bmatrix} \left(\mathbf{b}^1_{\text{bg}} - \frac{\mathbf{b}^1_{\text{fg}}}{\alpha|y|}\left(\mathbf{b}^1_{\text{bg}} - \mathbf{b}^1_{\text{fg}}\right)\right)\frac{x}{\alpha|y|} \\ \left(\mathbf{b}^2_{\text{bg}} - \frac{\mathbf{b}^2_{\text{fg}}}{|y|}\left(\mathbf{b}^2_{\text{bg}} - \mathbf{b}^2_{\text{fg}}\right)\right)\frac{y}{|y|} \\ \left(\mathbf{b}^3_{\text{bg}} - \frac{\mathbf{b}^3_{\text{fg}}}{|y|/\beta}\left(\mathbf{b}^3_{\text{bg}} - \mathbf{b}^3_{\text{fg}}\right)\right)\frac{z}{|y|/\beta} \end{bmatrix} & \text{if } |y| > \mathbf{b}^2_{\text{fg}},\ |y| \ge \frac{|x|}{\alpha},\ \frac{|y|}{\beta} \ge |z| \\ \begin{bmatrix} \left(\mathbf{b}^1_{\text{bg}} - \frac{\mathbf{b}^1_{\text{fg}}}{\gamma|z|}\left(\mathbf{b}^1_{\text{bg}} - \mathbf{b}^1_{\text{fg}}\right)\right)\frac{x}{\gamma|z|} \\ \left(\mathbf{b}^2_{\text{bg}} - \frac{\mathbf{b}^2_{\text{fg}}}{\beta|z|}\left(\mathbf{b}^2_{\text{bg}} - \mathbf{b}^2_{\text{fg}}\right)\right)\frac{y}{\beta|z|} \\ \left(\mathbf{b}^3_{\text{bg}} - \frac{\mathbf{b}^3_{\text{fg}}}{|z|}\left(\mathbf{b}^3_{\text{bg}} - \mathbf{b}^3_{\text{fg}}\right)\right)\frac{z}{|z|} \end{bmatrix} & \text{if } |z| > \mathbf{b}^3_{\text{fg}},\ |z| \ge \frac{|x|}{\gamma},\ |z| \ge \frac{|y|}{\beta} \end{cases}$$
$$\text{where } \alpha = \frac{\mathbf{b}^1_{\text{fg}}}{\mathbf{b}^2_{\text{fg}}}, \beta = \frac{\mathbf{b}^2_{\text{fg}}}{\mathbf{b}^3_{\text{fg}}}, \gamma = \frac{\mathbf{b}^1_{\text{fg}}}{\mathbf{b}^3_{\text{fg}}} \tag{9}$$

$$\mathcal{L}_{\text{rgb}} = \sum_{\mathbf{r}\in\mathcal{R}} \left\| \hat{C}(\mathbf{r}) - C(\mathbf{r}) \right\|_2^2 \tag{10}$$

Additionally, we incorporated the distortion loss from Mip-NeRF 360 (Barron et al. 2022) to encourage a unimodal distribution of the Probability Density Function (PDF) along the ray. This step is pivotal in reducing common artifacts such as floating objects and ghosting, which often appear in methods based on hash grids.

$$\mathcal{L}_{\text{dist}} = \sum_{i,j} \omega_i \omega_j \left| \frac{s_i + s_{i+1}}{2} - \frac{s_j + s_{j+1}}{2} \right| + \frac{1}{3} \sum \omega_i^2 \left(s_{i+1} - s_i\right) \tag{11}$$

where s_i denotes the interval between sample points in the disparity space. And ω_i denotes the weights of volume rendering corresponding to the sample point. Lastly, we employed the S3IM (Xie et al. 2023) approach to further enhance the clarity of the rendered images.

$$\begin{aligned} \mathcal{L}_{\text{S3IM}} &= 1 - \text{S3IM}\left(\hat{\mathcal{R}}, \mathcal{R}\right) \\ \text{S3IM}\left(\hat{\mathcal{R}}, \mathcal{R}\right) &= \frac{1}{M} \sum_{m=1}^{M} \text{SSIM}\left(\mathcal{P}^{(m)}\left(\hat{C}\right), \mathcal{P}^{(m)}(C)\right) \end{aligned} \tag{12}$$

where M signifies the number of iterations in which rays from the mini-batch $\mathcal{R}$ are stochastically selected to constitute the patch $\mathcal{P}$. Ultimately, we arrive at our loss function.

$$\mathcal{L} = \lambda_{\text{rgb}}\mathcal{L}_{\text{rgb}} + \lambda_{\text{dist}}\mathcal{L}_{\text{dist}} + \lambda_{\text{S3IM}}\mathcal{L}_{\text{S3IM}} \tag{13}$$

3.5.3 | Fusion Process

In this section, we provide supplementary details on the fusion process of LCA-NeRF and showcase the fusion results. During the partitioning phase in LCA-NeRF, each sub-model's AABB is expanded by a certain factor to achieve overlap. In the fusion phase, we begin by sampling points across the entire scene. Subsequently, we query the AABB associated with each sampled point and infer the density and color from the respective sub-model. When sampling points fall within the overlapping region of AABBs, we employ an inverse distance weight (IDW) scheme to weight the output of each block accordingly. When the sampled point lies outside all AABBs, we select the nearest sub-model for rendering. In Figure 5, we present the fusion results. Depth maps in Figure 5 demonstrate our block fusion method's good geometric consistency. Global perspectives show that block radiance field reconstruction with CAE maintains lighting consistency, even from distant viewpoints.

4 | Experiment

4.1 | Experimental Setup

4.1.1 | Datasets

We evaluate LAC-NeRF on Building, Rubble datasets from Mill 19 (Turki et al. 2022), and Campus, Sci-Art datasets from UrbanScene3D (Lin et al. 2022). The rubble dataset covers an area of $206 \times 248m^2$ and involves 1678 images. The building dataset covers $262 \times 438m^2$ and involves 1940 images. The sci-art dataset covers $291 \times 491m^2$ and involves 3019 images. The campus dataset covers $1346 \times 1542m^2$ and involves 5871 images.

4.1.2 | Evaluation Metrics

We evaluate the existing methods and our method in terms of PSNR, SSIM (Wang et al. 2004), and the VGG implementation of LPIPS (Zhang et al. 2018) to quantitatively assess our results in novel view synthesis.

4.1.3 | Compared Methods

We compare our model with Mega-NeRF (Turki et al. 2022) with eight partitions and Switch-NeRF (Zhenxing and Xu 2022) with eight partitions. Although GP-NeRF adopts the codebase of Mega-NeRF, it lacks a comprehensive fusion algorithm within its code, and the issue of block fusion is not addressed in the paper. Moreover, only a single block is used for training in their experiments. Therefore, in our experiments, GP-NeRF (Zhang et al. 2023) with 1 partition is utilized.

Crucially, Mega-NeRF (Turki et al. 2022), GP-NeRF (Zhang et al. 2023), and Switch-NeRF (Zhenxing and Xu 2022) utilize per-image appearance embeddings, necessitating the use of half of the pixels from the test dataset during training to optimize the corresponding appearance embeddings, followed by testing on the remaining half, which can be observed in their code. In

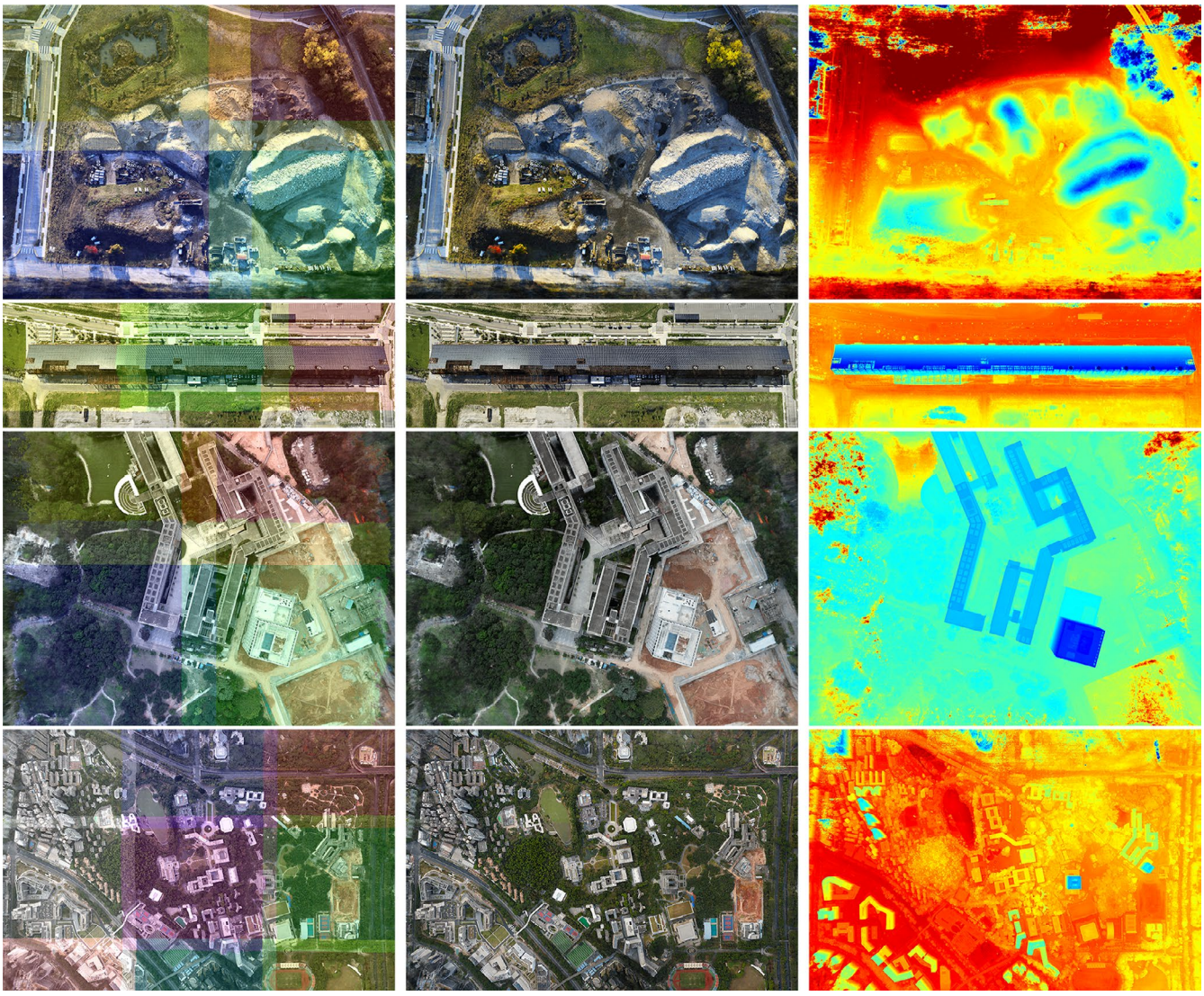

FIGURE 5 | Fusion results. The left column displays the fused visual representation with colored masks denoting different blocks. The middle column presents a global scene view, while the right column shows the corresponding depth maps. Blue indicates lower depths, and red indicates higher depths. The rows correspond to different scenes: rubble, building, sciart, and campus.

TABLE 1 | Quantitative comparison results with Mega-NeRF, GP-NeRF, and Switch-NeRF on Mill-19 datasets. See Section 4.2 for detailed descriptions.

	Mill 19—Rubble				Mill 19—Building			
	PSNR↑	SSIM↑	LPIPS↓	Time (h)↓	PSNR↑	SSIM↑	LPIPS↓	Time (h)↓
Mega-NeRF (Turki et al. 2022)	24.06	0.55	0.512	30:48	20.93	0.547	0.504	29:49
GP-NeRF (Zhang et al. 2023)	24.08	0.563	0.497	**1:22**	20.99	0.565	0.490	**1:22**
Switch-NeRF (Zhenxing and Xu 2022)	<u>24.31</u>	0.562	0.496	42:30	<u>21.54</u>	0.579	0.474	42:30
LCA-NeRF (ours)	24.17	<u>0.638</u>	<u>0.398</u>	<u>3:17</u>	21.23	<u>0.592</u>	<u>0.365</u>	<u>3:30</u>
LCA-NeRF-TE (ours)	**24.38**	**0.647**	**0.366**	<u>3:17</u>	**21.61**	**0.614**	**0.348**	<u>3:30</u>

Note: **Bold** represents the best result, and <u>underlined</u> represents the second-best result.

contrast, our LCA-NeRF infers appearance embeddings based on the spatial relationships among appearance embeddings in the training set. Therefore, LCA-NeRF refrains from training on the test dataset images, thus avoiding potential overestimation of test performance. To provide a fair comparison under identical experimental conditions, we have also evaluated the

TABLE 2 | Quantitative comparison results with Mega-NeRF, GP-NeRF, and Switch-NeRF on UrbanScene3D datasets. See Section 4.2 for detailed descriptions.

	UrbanScene3D—SciArt				UrbanScene3D—Campus			
	PSNR↑	SSIM↑	LPIPS↓	Time (h)↓	PSNR↑	SSIM↑	LPIPS↓	Time (h)↓
Mega-NeRF (Turki et al. 2022)	25.60	0.770	0.390	27:39	23.42	0.537	0.618	28:03
GP-NeRF (Zhang et al. 2023)	25.56	0.783	0.373	**1:22**	23.46	0.544	0.611	**1:22**
Switch-NeRF (Zhenxing and Xu 2022)	<u>26.52</u>	0.795	0.360	42:30	<u>23.62</u>	0.541	0.609	45:30
LCA-NeRF (ours)	26.11	<u>0.802</u>	<u>0.312</u>	<u>4:17</u>	23.58	<u>0.584</u>	<u>0.398</u>	<u>5:40</u>
LCA-NeRF-TE (ours)	**26.57**	**0.811**	**0.289**	<u>4:17</u>	**23.75**	**0.588**	**0.384**	<u>5:40</u>

Note: **Bold** represents the best result, and <u>underlined</u> represents the second-best result.

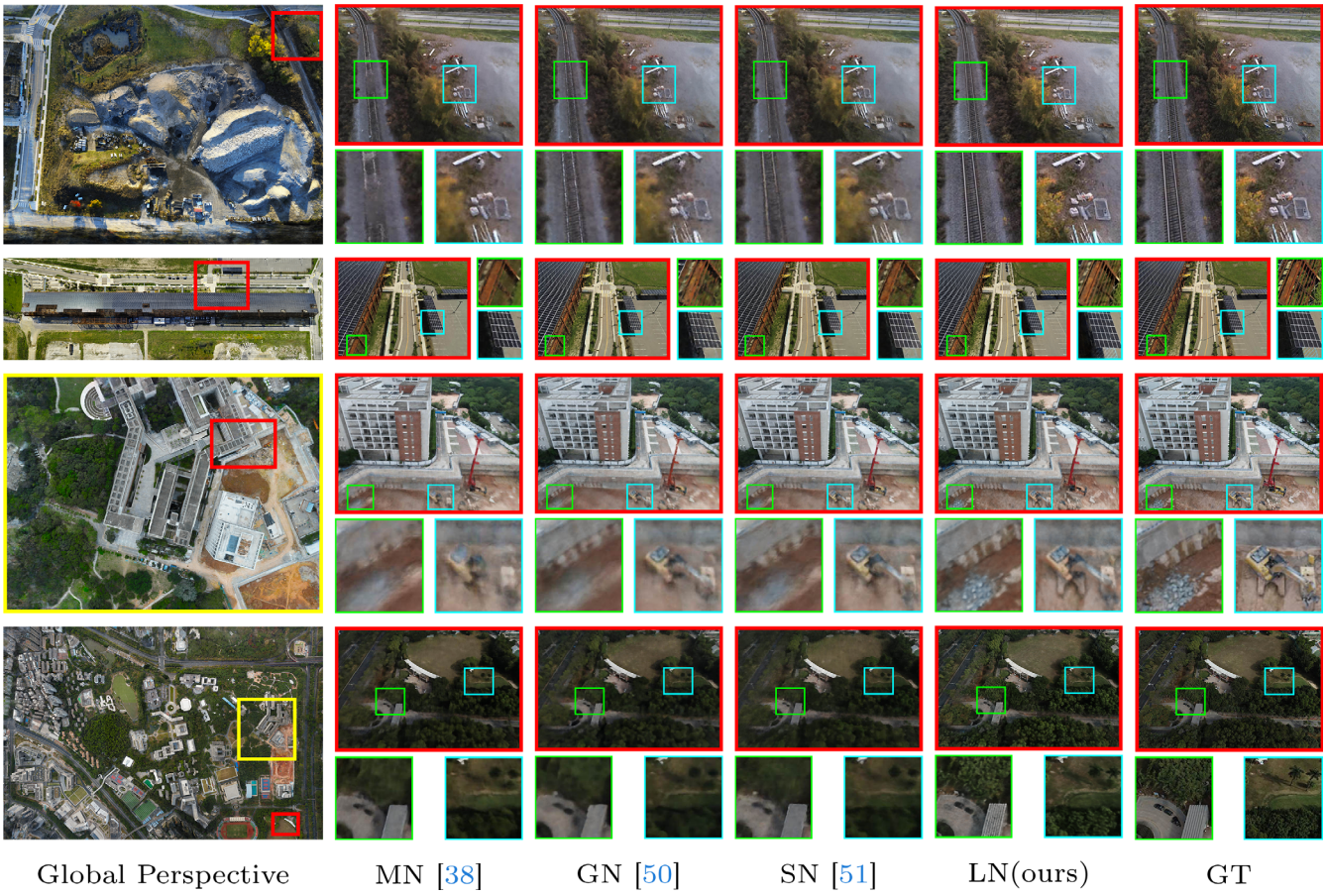

FIGURE 6 | Qualitative comparison results on the Mill-19 and Urbanscene3D. The leftmost column displays large-scale scenes rendered from a global perspective using LCA-NeRF. Moving from the second to the fifth column, the methods are Mega-NeRF, GP-NeRF, Switch-NeRF, and LCA-NeRF, respectively. The rightmost column represents the ground truth. The first row corresponds to the rubble scene, the second row to the building scene, the third row to the sciart scene, and the fourth row to the campus scene.

training-on-test version of LCA-NeRF, termed LCA-NeRF-TE, that is training on half of the pixels from the test set and subsequently testing on the other half of the pixels.

4.1.4 | Setting

We used 2×2 sub-models for the rubble, sci-art scenes, 3×1 sub-models for the building scene, and 3×3 sub-models for the campus scene. Each scene was trained with 60 K iteration using a batch size of 96×96 rays. The learning rate was set to 0.006. We set $\lambda_{\text{rgb}} = 1.0$, $\lambda_{\text{dist}} = 0.15$, $\lambda_{\text{S3IM}} = 1.0$, $\lambda_{\text{RT}} = 0.3$, $k_{\text{top}} = 10$. The dimension of the appearance embedding was set to 48. The hash table size is set to $L = 16$, $T = 2^{20}$, $F = 2$. The lowest and highest resolutions are set to 16 and 4096. While we independently sample foreground and background regions, we utilize different portions of a shared hash grid to represent both foreground and background. This approach offers the advantage of avoiding

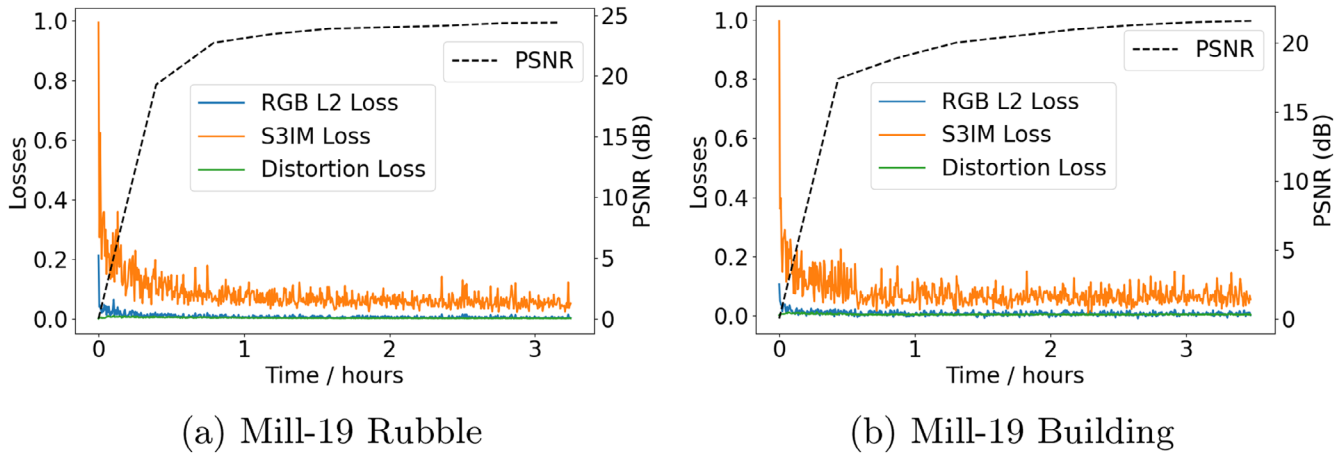

FIGURE 7 | Loss convergence and PSNR curves of LCA-NeRF during training on the Mill-19 dataset. LCA-NeRF consistently converges within a few hours across various scenes.

additional overhead for maintaining a separate hash grid dedicated solely to the background. We constructed our work using the PyTorch-Lightning framework (Falcon 2019) and utilized Adam optimizer (Kingma and Ba 2014) for model optimization. The model was trained and evaluated on four NVIDIA RTX A5000 GPUs.

4.2 | Benchmark Performance and Analysis

4.2.1 | Comparison With Existing Methods

We conducted qualitative and quantitative comparisons with existing block-based methods (Turki et al. 2022; Zhenxing and Xu 2022) and hash-grid-based method (Zhang et al. 2023) for large-scale scenes.

The result in Tables 1 and 2 indicates that our LCA-NeRF has achieved the best performance on the four scenes. Our SSIM and LPIPS metrics outperform other methods, with a slight lead in PSNR. The improvements in SSIM and LPIPS can be attributed to the enhanced richness and clarity of scene details. This can also be substantiated by Figure 6, which demonstrates that our method exhibits greater clarity in details compared to other approaches. This enhancement is a result of our utilization of multiple hash grids for scene modeling, which provides a superior model capacity compared to other methods. Lower PSNR results of LCA-NeRF can be attributed to the fact that we did not use half of the pixels from the test set images for training to generate corresponding embeddings. Instead, we inferred embeddings through spatial clustering. As a result, we may exhibit weaker performance in PSNR, which is sensitive to photometric and chromatic aspects.

To maintain fairness, we conducted experiments under the same configuration. LCA-NeRF-TE, built upon LCA-NeRF, trains on half of the pixels from the test set images and tests on the other half. This approach consistently yields superior results across all four datasets. The results of LCA-NeRF-TE suggest that training appearance embeddings separately for specific viewpoints can improve rendering quality. However, in practical applications, it is often unknown which appearance embedding to use for new viewpoints. Therefore, we ultimately rely on the results obtained by LCA-NeRF.

TABLE 3 | Per-block average memory consumption during training. Since the network and hash grid sizes remain fixed, memory usage varies little across different scenes.

Memory usage (GB)	Rubble	Building	Sci-Art	Campus
LCA-NeRF (ours)	14.8	15.4	15.2	15.7

In terms of training time, LCA-NeRF and GP-NeRF outperform Mega-NeRF and Switch-NeRF by an order of magnitude. This is primarily attributed to the fact that both of these methods employ discrete structures and a shallower MLP for representation, leading to a notable improvement in convergence and inference speed. Our training time, while marginally longer than that of GP-NeRF, can be explained by variances in our code framework compared to GP-NeRF. Additionally, the increased time spent in calculating nearby poses contributes to this difference. As shown in Figure 7, we present the loss and PSNR curves over training time for LCA-NeRF. The model consistently converges to a suboptimal value early in training and achieves full convergence within a few hours across various scenes. Regarding model size, Mega-NeRF and Switch-NeRF, which use MLP for representation, need just tens of MB for storage. Conversely, GP-NeRF and LCA-NeRF, employing hash grids, demand hundreds of MB. Specifically, LCA-NeRF's sub-blocks each occupy 300–400 MB. We also report the GPU memory consumption of LCA-NeRF across different scenes during training time in Table 3. Since the parameter settings are fixed across scenes, the memory usage shows little variation.

In the context of large-scale block-based methods, the architecture of MLP-based approaches like Switch-NeRF and Mega-NeRF lacks the structural clarity found in grid-based methods such as GP-NeRF and our approach. Furthermore, compared to

TABLE 4 | Ablation experiments for the clustered appearance embeddings.

	Mill 19-Rubble			UrbanScene3D-SciArt		
	PSNR↑	SSIM↑	LPIPS	PSNR↑	SSIM↑	LPIPS↓
LCA-NeRF-no-em	19.50	0.511	0.486	22.02	0.537	0.544
LCA-NeRF-nearest-em	24.15	0.601	0.432	25.86	0.782	0.348
LCA-NeRF-cluster-em	**24.17**	**0.638**	**0.398**	**26.11**	**0.802**	**0.312**

Note: **Bold** represents the best result.

TABLE 5 | Quantitative comparison experiments for different values of k_{top} and λ_{RT}.

	Mill 19 Rubble		
Method	**PSNR↑**	**SSIM↑**	**LPIPS↓**
$k_{\text{top}} = 5,\ \lambda_{\text{RT}} = 0.3$	23.98	0.627	0.407
$k_{\text{top}} = 10, \lambda_{\text{RT}} = 0.3$	**24.17**	0.638	0.398
$k_{\text{top}} = 15, \lambda_{\text{RT}} = 0.3$	24.09	**0.642**	**0.395**
$k_{\text{top}} = 10, \lambda_{\text{RT}} = 0.5$	23.36	0.598	0.446
$k_{\text{top}} = 10, \lambda_{\text{RT}} = 0.1$	23.61	0.584	0.460

Note: **Bold** represents the best result.

GP-NeRF, we achieve enhanced geometric detail representation in large-scale scenes, primarily because block-wise approaches are necessary. To validate the correctness of our hypothesis, we conducted the following ablation experiments.

4.3 | Ablation Study

In this section, we conduct comparisons between LCA-NeRF and several ablations on the Mill19-Rubble and UrbanScene3D-SciArt datasets, which involve unbounded regions and significant lightning variations.

4.3.1 | Effectiveness of Spatial Clustered AE

To validate the effectiveness of spatial clustered appearance embedding, we conducted experiments involving LCA-NeRF-no-em, which trains without appearance embedding, LCA-NeRF-nearest-em, which trains/tests with nearest appearance embedding, and LCA-NeRF-cluster-em, training/testing with clustered appearance embedding. We present our results in Table 4. Our observations reveal that using clustered appearance embeddings improves LCA-NeRF's rendering quality with inference embeddings. On the other hand, relying solely on the nearest appearance embeddings for rendering makes it difficult to infer embeddings that accurately represent the lighting conditions for the current viewpoint, ultimately leading to a decrease in rendering quality.

We also have introduced two hyperparameters k_{top} and λ_{RT} in our proposed CAE. We conducted an analysis on the values of these two hyperparameters using the Rubble dataset, as shown in Table 5. Data in Table 5 indicate that CAE is more sensitive to λ_{RT}, which is related to camera distribution.

We suggest choosing smaller values for λ_{RT} as the camera distribution becomes denser. Additionally, we observed that the choice of k_{top} affects the consistency after block fusion. When k_{top} is set to smaller values, inconsistencies between blocks become more pronounced, as illustrated in Figure 8. This is primarily due to the existence of closely situated cameras under varying lighting conditions, causing ambiguity in spatially based clustering. When k_{top} is set to a smaller value, this inconsistency becomes more pronounced. Therefore, considering that illumination can vary due to different shooting times even in static scenes, we decouple illumination into $\mathbf{l}^{(g)}$, describing local scene geometry, and $\mathbf{l}^{(e)}$, depicting the camera's own exposure, and opt for a larger k_{top} value. This approach allows for fitting lighting variations over shorter time spans, as illustrated in Figure 9.

4.3.2 | Effectiveness of Hash Grid-Based Partition

To validate the effectiveness of our block-based method, we conducted tests on the Sci-art dataset. LCA-NeRF-1×1 denotes training the entire scene with a single LCA-NeRF model, while LCA-NeRF-2×2 involves dividing the space into four equal blocks and training the scene with four separate LCA-NeRF models. LCA-NeRF-3×3, on the other hand, signifies the use of nine sub-models to model a 3×3 grid of regions. Our results are presented in Table 6.

The results of 2×2 and 3×3 models indicate that dividing the scene into several sub-blocks and training them separately can enhance modeling quality. However, the improvement from 3×3 models compared to 2×2 models is marginal, and there is a slight decrease in PSNR. We attribute this to the fact that under the same experimental conditions, the expressive capacity of the hash grid has reached its limit, and finer divisions do not further improve modeling quality. Conversely, excessive partitioning fragments the dataset, limiting each sub-model from receiving sufficient data for training.

4.3.3 | Effectiveness of S3IM and Distortion Loss

Compared to other block-based approaches, we have incorporated distortion loss and S3IM loss. To effectively validate the positive impact of these two loss functions on our results, we also conducted ablation studies for each loss on the Mill19-Rubble dataset. Our results are presented in Table 7. w/o $\mathcal{L}_*$

FIGURE 8 | Qualitative analysis of the value selection for k_{top}. The first two columns show the rendering effects with different values of k_{top}, while the last column presents the ground truth with a block mask.

FIGURE 9 | Qualitative results under similar viewpoints but varying lighting conditions. Inconsistent lighting across geometry is represented by $\mathbf{l}^{(e)}$, allowing LCA-NeRF to render realistic images with minor dynamic light sources.

TABLE 6 | Comparison of different block granularities on the Sci-Art dataset.

	Urbanscene3D Sci-Art		
Method	**PSNR↑**	**SSIM↑**	**LPIPS↓**
LCA-NeRF-1×1	24.28	0.714	0.363
LCA-NeRF-2×2	**26.11**	0.802	0.312
LCA-NeRF-3×3	25.96	**0.814**	**0.276**

Note: **Bold** represents the best result.

TABLE 7 | ablation experiments of $\mathcal{L}_{\text{dist}}$ and $\mathcal{L}_{\text{S3IM}}$ on the Rubble dataset.

	Mill 19-Rubble		
Method	**PSNR↑**	**SSIM↑**	**LPIPS↓**
w/o $\mathcal{L}_{\text{S3IM}}$	24.05	0.595	0.424
w/o $\mathcal{L}_{\text{dist}}$	23.28	0.560	0.465
w/ $\mathcal{L}_{\text{dist}}$ and $\mathcal{L}_{\text{S3IM}}$	**24.17**	**0.638**	**0.398**

Note: **Bold** represents the best result.

indicates training without $\mathcal{L}_*$, while w/ $\mathcal{L}_*$ denotes the inclusion of $\mathcal{L}_*$ in training. Table 7 indicates that S3IM is most effective in improving the SSIM of rendered images. Furthermore, in our experiments, the use of S3IM loss results in clearer depictions of objects with complex structures, as illustrated in Figure 10. This suggests that applying SSIM supervision multiple times to

FIGURE 10 | Qualitative ablation experiments for $\mathcal{L}_{\text{S3IM}}$ and $\mathcal{L}_{\text{dist}}$. Utilizing $\mathcal{L}_{\text{S3IM}}$ and $\mathcal{L}_{\text{dist}}$ loss functions can result in clearer images with fewer floating objects in the scene.

TABLE 8 | Comparison of different contractions. LCA-NeRF-cubic denotes the utilization of the Equation (7), while LCA-NeRF-cuboid is associated with the Equation (9).

	Urbanscene3D Sci-Art		
Method	**PSNR↑**	**SSIM↑**	**LPIPS↓**
LCA-NeRF-cubic	25.72	0.751	0.339
LCA-NeRF-cuboid	**26.11**	**0.802**	**0.312**

Note: **Bold** represents the best result.

FIGURE 11 | Qualitative comparison of different contractions: Left using Equation (7), Right using Equation (9). Due to weaker contraction strength in the *xy* direction, anisotropic contraction, as opposed to isotropic contraction, can reveal a clearer background.

randomly permuted pixel blocks successfully captures global structural information, thereby reducing issues of aliasing and blurriness. Table 7 demonstrates that $\mathcal{L}_{\text{dist}}$ consistently improves all metrics. As shown in Figure 10, omitting $\mathcal{L}_{\text{dist}}$ results in floating objects within the scene. The use of $\mathcal{L}_{\text{dist}}$, by encouraging a unimodal distribution of pdf along the rays, effectively eliminates floating artifacts.

4.3.4 | Effectiveness of Contraction for Cuboid

Anisotropic background contraction is intended to map the unbounded background space into a more compact bounded region, thereby producing a clearer background. This is supported by the ablation study in Table 8 and the qualitative comparisons in Figure 11.

5 | Conclusion

In this work, we introduce LCA-NeRF, a NeRF variant tailored for large-scale scenes, addressing challenges related to lighting variations and scene details. LCA-NeRF utilizes clustered appearance embeddings for adaptive view inference and employs block-based partitioning and hash grid representation for enhanced capabilities. We assessed LCA-NeRF on multiple large-scale datasets and achieved state-of-the-art results among large-scale methods.

5.1 | Limitation

While CAE can facilitate smoother transitions between blocks, our approach does not account for dynamic scenes, thereby struggling to adapt to scenarios where lighting changes dramatically over time, such as in crowdsourced datasets. Our approach ties AE to geometric information, creating a structured AE field. Therefore, for crowdsourced datasets, we consider linking AEs to time, forming a structured dynamic AE field. Additionally, volume rendering bias hinders precise geometric recovery. Future work will integrate SDF fields and geometric priors for refined geometric reconstruction in extensive scenes.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (no. U22B2055 and 62273345), the Beijing Natural Science Foundation (no. L223003), and the Key R&D Project in Henan Province (no. 231111210300).

Conflicts of Interest

The authors declare no conflicts of interest.

Data Availability Statement

The data that support the findings of this study are available in UrbanScene3D at https://github.com/Linxius/UrbanScene3D. These data were derived from the following resources available in the public domain: - UrbanScene3D, https://vcc.tech/UrbanScene3D.

References

Barron, J. T., B. Mildenhall, M. Tancik, P. Hedman, R. Martin-Brualla, and P. P. Srinivasan. 2021. "Mip-Nerf: A Multiscale Representation for Anti-Aliasing Neural Radiance Fields." In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 5855–5864.

Barron, J. T., B. Mildenhall, D. Verbin, P. P. Srinivasan, and P. Hedman. 2022. "Mip-Nerf 360: Unbounded Anti-Aliased Neural Radiance Fields." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5470–5479.

Chan, E. R., C. Z. Lin, M. A. Chan, et al. 2022. "Efficient Geometry-Aware 3D Generative Adversarial Networks." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 16123–16133.

Chen, A., Z. Xu, A. Geiger, J. Yu, and H. Su. 2022. "Tensorf: Tensorial Radiance Fields." In *European Conference on Computer Vision*, 333–350. Springer.

Eslami, S. A., D. Jimenez Rezende, F. Besse, et al. 2018. "Neural Scene Representation and Rendering." *Science* 360, no. 6394: 1204–1210.

Falcon, W. 2019. "The PyTorch Lightning Team: PyTorch Lightning." https://doi.org/10.5281/zenodo.3828935. https://github.com/Lightning-AI/lightning.

Fridovich-Keil, S., G. Meanti, F. R. Warburg, B. Recht, and A. Kanazawa. 2023. "K-Planes: Explicit Radiance Fields in Space, Time, and Appearance." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12479–12488.

Fu, Q., Q. Xu, Y. S. Ong, and W. Tao. 2022. "Geo-Neus: Geometry-Consistent Neural Implicit Surfaces Learning for Multi-View Reconstruction." In *Advances in Neural Information Processing Systems (NeurIPS)*.

Guo, J., Y. Liu, X. Song, H. Liu, X. Zhang, and Z. Cheng. 2022. "Line-Based 3D Building Abstraction and Polygonal Surface Reconstruction From Images." *IEEE Transactions on Visualization and Computer Graphics* 30: 3283–3297.

Guo, J., H. Qin, Y. Zhou, X. Chen, L. Nan, and H. Huang. 2024. "Fast Building Instance Proxy Reconstruction for Large Urban Scenes." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 46: 7267–7282.

Huang, B., Z. Yu, A. Chen, A. Geiger, and S. Gao. 2024. "2D Gaussian Splatting for Geometrically Accurate Radiance Fields." In *ACM SIGGRAPH 2024 Conference Papers*, 1–11.

Jiang, C. M., A. Sud, A. Makadia, J. Huang, M. Niessner, and T. Funkhouser. 2020. "Local Implicit Grid Representations for 3D Scenes." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Kerbl, B., G. Kopanas, T. Leimkühler, and G. Drettakis. 2023. "3D Gaussian Splatting for Real-Time Radiance Field Rendering." *ACM Transactions on Graphics* 42, no. 4: 139.

Kingma, D. P., and J. Ba. 2014. "Adam: A Method for Stochastic Optimization." arXiv preprint arXiv:1412.6980.

Li, Z., L. Li, and J. Zhu. 2023. "Read: Large-Scale Neural Scene Rendering for Autonomous Driving." *Proceedings of the AAAI Conference on Artificial Intelligence* 37: 1522–1529.

Li, Z., T. Müller, A. Evans, et al. 2023. "Neuralangelo: High-Fidelity Neural Surface Reconstruction." In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Lin, J., Z. Li, X. Tang, et al. 2024. "Vastgaussian: Vast 3D Gaussians for Large Scene Reconstruction." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5166–5175.

Lin, L., Y. Liu, Y. Hu, X. Yan, K. Xie, and H. Huang. 2022. "Capturing, Reconstructing, and Simulating: The Urbanscene3d Dataset." In *ECCV*.

Liu, L., J. Gu, K. Zaw Lin, T. S. Chua, and C. Theobalt. 2020. "Neural sparse voxel fields." *Advances in Neural Information Processing Systems* 33: 15651–15663.

Liu, Y., C. Luo, L. Fan, N. Wang, J. Peng, and Z. Zhang. 2024. "Citygaussian: Real-Time High-Quality Large-Scale Scene Rendering With Gaussians." In *European Conference on Computer Vision*, 265–282. Springer.

Liu, Y., C. Luo, Z. Mao, J. Peng, and Z. Zhang. 2024. Citygaussianv2: Efficient and Geometrically Accurate Reconstruction for Large-Scale Scenes. arXiv preprint arXiv:2411.00771.

Long, X., C. Lin, L. Liu, et al. 2023. "Neuraludf: Learning Unsigned Distance Fields for Multi-View Reconstruction of Surfaces With Arbitrary Topologies." In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 20834–20843. https://doi.org/10.1109/CVPR52729.2023.01996.

Luo, K., T. Guan, L. Ju, H. Huang, and Y. Luo. 2019. "P-Mvsnet: Learning Patch-Wise Matching Confidence Aggregation for Multi-View Stereo."

In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 10452–10461.

Martin-Brualla, R., N. Radwan, M. S. M. Sajjadi, J. T. Barron, A. Dosovitskiy, and D. Duckworth. 2021. "NeRF in the Wild: Neural Radiance Fields for Unconstrained Photo Collections." In *CVPR*.

Meuleman, A., Y. L. Liu, C. Gao, et al. 2023. "Progressively Optimized Local Radiance Fields for Robust View Synthesis." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 16539–16548.

Mildenhall, B., P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng. 2021. "Nerf: Representing Scenes as Neural Radiance Fields for View Synthesis." *Communications of the ACM* 65, no. 1: 99–106.

Moulon, P., P. Monasse, R. Perrot, and R. Marlet. 2016. "OpenMVG: Open multiple view geometry." In *International Workshop on Reproducible Research in Pattern Recognition*, 60–74. Springer.

Müller, T., A. Evans, C. Schied, and A. Keller. 2022. "Instant Neural Graphics Primitives With a Multiresolution Hash Encoding." *ACM Transactions on Graphics* 41, no. 4: 1–15.

Park, J. J., P. Florence, J. Straub, R. Newcombe, and S. Lovegrove. 2019. "Deepsdf: Learning Continuous Signed Distance Functions for Shape Representation." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 165–174.

Rebain, D., W. Jiang, S. Yazdani, K. Li, K. M. Yi, and A. Tagliasacchi. 2021. "Derf: Decomposed Radiance Fields." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14153–14161.

Reiser, C., R. Szeliski, D. Verbin, et al. 2023. "Merf: Memory-Efficient Radiance Fields for Real-Time View Synthesis in Unbounded Scenes." *ACM Transactions on Graphics* 42, no. 4: 1–12.

Schönberger, J. L., and J. M. Frahm. 2016. "Structure-From-Motion Revisited." In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Schönberger, J. L., E. Zheng, M. Pollefeys, and J. M. Frahm. 2016. "Pixelwise View Selection for Unstructured Multi-View Stereo." In *European Conference on Computer Vision (ECCV)*.

Sitzmann, V., M. Zollhoefer, and G. Wetzstein. 2019. "Scene Representation Networks: Continuous 3D-Structure-Aware Neural Scene Representations." In *Advances in Neural Information Processing Systems. vol. 32*, edited by H. Wallach, H. Larochelle, A. Beygelzimer, F. d Alché-Buc, E. Fox, and R. Garnett. Curran Associates. https://proceedings.neurips.cc/paper_files/paper/2019/file/b5dc4e5d9b495d0196f61d45b26ef33e-Paper.pdf.

Sun, C., M. Sun, and H. T. Chen. 2022a. "Direct Voxel Grid Optimization: Super-Fast Convergence for Radiance Fields Reconstruction." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5459–5469.

Sun, C., M. Sun, and H. T. Chen. 2022b. "Improved Direct Voxel Grid Optimization for Radiance Fields Reconstruction." arXiv preprint arXiv:2206.05085.

Tancik, M., V. Casser, X. Yan, et al. 2022. "Block-Nerf: Scalable Large Scene Neural View Synthesis." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8248–8258.

Turki, H., D. Ramanan, and M. Satyanarayanan. 2022. "Mega-Nerf: Scalable Construction of Large-Scale Nerfs for Virtual Fly-Throughs." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12922–12931.

Turki, H., J. Y. Zhang, F. Ferroni, and D. Ramanan. 2023. "Suds: Scalable Urban Dynamic Scenes." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12375–12385.

Wang, P., L. Liu, Y. Liu, C. Theobalt, T. Komura, and W. Wang. 2021. *Neus: Learning Neural Implicit Surfaces by Volume Rendering for Multi-View Reconstruction*. NeurIPS.

Wang, Z., A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. 2004. "Image Quality Assessment: From Error Visibility to Structural Similarity." *IEEE Transactions on Image Processing* 13, no. 4: 600–612.

Xie, Z., X. Yang, Y. Yang, et al. 2023. "S3im: Stochastic Structural Similarity and Its Unreasonable Effectiveness for Neural Fields." In *International Conference on Computer Vision*.

Xu, L., Y. Xiangli, S. Peng, et al. 2023. "Grid-Guided Neural Radiance Fields for Large Urban Scenes." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8296–8306.

Xu, Q., W. Kong, W. Tao, and M. Pollefeys. 2022. "Multi-Scale Geometric Consistency Guided and Planar Prior Assisted Multi-View Stereo." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45: 1–18.

Yang, Z., Y. Chen, J. Wang, et al. 2023. "Unisim: A Neural Closed-Loop Sensor Simulator." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1389–1399.

Yariv, L., J. Gu, Y. Kasten, and Y. Lipman. 2021. "Volume Rendering of Neural Implicit Surfaces." In *Thirty-Fifth Conference on Neural Information Processing Systems*.

Yariv, L., Y. Kasten, D. Moran, et al. 2020. "Multiview Neural Surface Reconstruction by Disentangling Geometry and Appearance." In *Advances in Neural Information Processing Systems. vol. 33*, edited by H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, 2492–2502. Curran Associates. https://proceedings.neurips.cc/paper_files/paper/2020/file/1a77befc3b608d6ed363567685f70e1e-Paper.pdf.

Zhang, K., G. Riegler, N. Snavely, and V. Koltun. 2020. "Nerf++: Analyzing and Improving Neural Radiance Fields." arXiv preprint arXiv:2010.07492.

Zhang, R., P. Isola, A. A. Efros, E. Shechtman, and O. Wang. 2018. "The Unreasonable Effectiveness of Deep Features as a Perceptual Metric." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 586–595.

Zhang, Y., G. Chen, and S. Cui. 2023. "Efficient Large-Scale Scene Representation With a Hybrid of High-Resolution Grid and Plane Features." arXiv preprint arXiv:2303.03003.

Zhenxing, M., and D. Xu. 2022. "Switch-Nerf: Learning Scene Decomposition With Mixture of Experts for Large-Scale Neural Radiance Fields." In *The Eleventh International Conference on Learning Representations*.