Camera-Lidar Consistent Neural Radiance Fields

Chao Hou[1], Fu Zhang[1†]

***Abstract*— Neural Radiance Fields (NeRFs) have become a leading technique for novel view synthesis, with promising applications in robotics. However, due to shape-radiance ambiguity, NeRFs often require additional depth inputs for regularization in outdoor scenarios. LiDAR provides accurate depth measurements, but current methods typically combine only a few frames, resulting in sparse depth maps and discrepancies with camera images. The asynchronous nature of LiDAR, where each point is captured at a different timestamp, introduces depth inaccuracies when treated as simultaneous. These errors, along with inherent LiDAR noise, create inconsistencies that hinder reconstruction accuracy. To address these challenges, we propose a continuous-time framework for joint Camera-LiDAR optimization, enabling more consistent radiance field reconstruction and improving both view synthesis and geometric accuracy.**

I. Introduction

3D reconstruction is fundamental in robotics, providing essential mapping information for tasks like localization and visualization. Recently, implicit neural representations have gained popularity for their continuity and storage efficiency compared to traditional methods like point clouds, voxels, and meshes. Among these, Neural Radiance Fields (NeRFs) [17] stand out, enabling high-quality reconstructions from a set of posed images. Research [37] [32] [29] [11] has extended NeRF's application from object-centric settings to complex outdoor scenes. However, when applied to scenes captured with handheld devices, such as in SLAM datasets, NeRF-based reconstructions face challenges. These datasets often feature long trajectories with little overlap, leading to artifacts like blurs and "floaters" in RGB-only NeRF, resulting in less accurate and noisier geometries. Notably, NeRF's reliance on photometric errors, akin to direct visual odometry [7] [8], makes it struggle in low-texture areas or where viewpoint overlap is insufficient.

For robotics datasets (e.g., autonomous driving or SLAM datasets), collinear camera motions and sparser viewpoints exacerbate the issue, resulting in insufficient constraints for accurate geometry. In such scenes, LiDAR depth is often introduced for regularization [3] [32]. A common approach is to project the LiDAR points onto the camera image plane, as shown in Figure 1(a). However, due to LiDAR's sparsity, this supervision is incomplete. Additionally, the different viewpoints of LiDAR and the camera cause occlusion problems when converting point clouds to depth maps, meaning the projected depth may exceed the actual depth, reducing the model's performance. Previous methods often assume that

†Corresponding author
[1]The University of Hong Kong, China, houchao@connect.hku.hk, fuzhang@hku.hk.

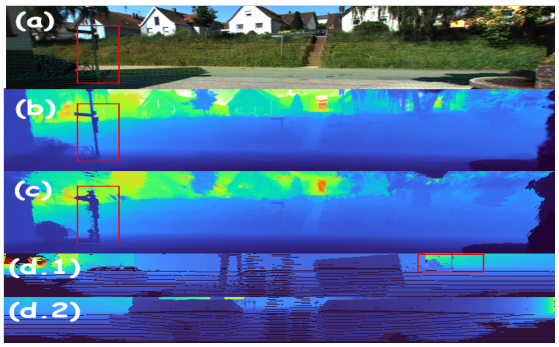

Fig. 1: Illustrations of LiDAR measurement noise and projection error: (a) Projected depth image from a single LiDAR frame. (b) Depth rendering using only RGB training. (c) Depth rendering using only LiDAR training. (d) Range image from a single LiDAR frame.

LiDAR points are located at the true surface, thus constraining the corresponding density distribution [19]. Figure 1(b) and 1(c) show the depth maps learned using only RGB and LiDAR data, respectively. LiDAR depth maps exhibit more noise for slender objects such as poles. Therefore, due to the misalignment, there is a trade-off between the modalities, making it challenging to improve both LiDAR and camera metrics simultaneously.

To address this issue, we do not directly impose depth constraints on the RGB rays; instead, we perform additional learning on the LiDAR rays. Rather than treating LiDAR as a static range image [24], we utilize a method inspired by continuous LiDAR odometry [5] to obtain undistorted LiDAR rays, referred to as the Continuous Time Formula(CTF). Previous approaches have mitigated the adverse effects of LiDAR noise either by designing depth filters [22] to exclude unreliable depth measurements or by preparing a separate set of features for LiDAR [25]. Our solution aligns with the latter. We designed the Masked Attention Fusion Module (MAFM) to integrate features from both LiDAR and the camera, using an attention mechanism to select features that are beneficial for RGB reconstruction.

In summary, our contributions are outlined below:

- We develop an octree-based sparse data structure for efficient retrieval of dense depth and normals from aggregated LiDAR point clouds.
- We improve the consistency between camera and LiDAR data at both the data and feature levels, reducing discrepancies and noise that could impair model performance.
- We introduce CLC-NeRF that efficiently combines the training of LiDAR point clouds and RGB images,

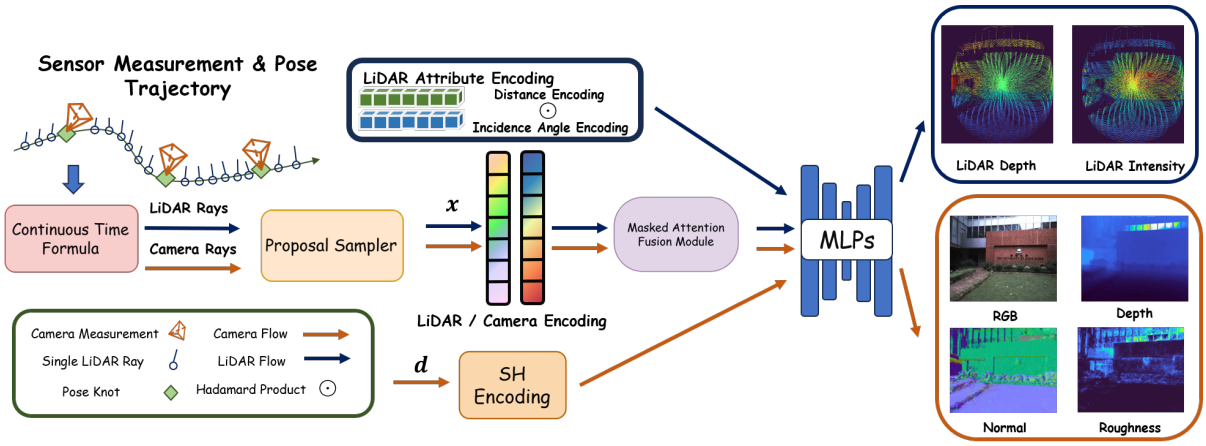

Fig. 2: **System Overview.** Our method takes RGB Image together with Lidar stream as input.

achieving high performance across both modalities.

II. Related Works

A. Neural Implicit Scene Representations

Neural implicit representations have swiftly risen as a pioneering technique in 3D geometry and radiance modeling. Leading this wave is the Neural Radiance Field (NeRF [17]), which employs multi-layer perceptrons to map 5D coordinates(position and direction) into density and radiance, providing a continuous and compact scene representation. However, the vanilla NeRF model demands extensive training to accurately capture local details. Recent methodologies [18] [21] [4] [34] have accelerated training process, albeit at the cost of heightened storage requirements, by substituting the singular, massive MLP with a combination of a voxel grid and more light-weight MLPs. With memory efficiency and fast convergency, multi-resolution hash grids [18] has become essential components for fast NeRF training.

B. Depth-Supervised NeRF

A key challenge in applying NeRF to large outdoor scenes is its shape-radiance ambiguity [37], leading to artifacts in sparse views and inaccuracies in areas with strong radiance variations. To address this, DS-NeRF [6] uses structure-from-motion (SfM) point clouds for training with sparse inputs, while DDP-NeRF [20] applies depth completion to refine these points into detailed depth maps. MonoSDF [36] directly integrates monocular depth cues. In outdoor settings, LiDAR measurements are more common. URF [19] uses point clouds to predefine rendering weights along rays. CLONeR [2] employs an occupancy grid, updating it with depth loss, and LightningNeRF [1] initializes the grid with accumulated point clouds, capturing coarse geometry for efficient ray sampling.

A comprehensive study by [28] investigates depth prior selection criteria while comparatively analyzing their relative merits, affirming the effectiveness of depth supervision in settings with sparse viewpoints and highlighting the importance of dense depth. However, the introduction of depth does not always yield positive results. It has also been observed that incorporating depth can lead to a decline in image quality metrics on training views, suggesting that in areas with dense camera coverage, the inherent noise in depth and inconsistencies across modalities may degrade model performance. S-NeRF [32] utilizes LiDAR depth completion to derive dense depth data and integrates geometric consistencies across depth, optical flow, and RGB to formulate a trainable confidence map for weighting depth loss. LidaRF [22] has developed an occlusion-aware robust supervision scheme that progressively learns reliable depth data from closer to further distances, addressing the occlusion issues inherent in aggregated point clouds. AlignMiF [25] utilizes separate decomposed hash encodings for LiDAR and camera, maintaining geometric consistency at the coarse level while ensuring independence at the fine level, thus preserving the unique details of each modality.

III. PRELIMINARIES

Neural Radiance Field (NeRF) fits a coordinate-based neural network $\mathcal{F}(\vartheta) : (\mathbf{x}, \mathbf{d}) \longmapsto (\mathbf{c}, \sigma)$ to map position $\mathbf{x}$ and ray direction $\mathbf{d}$ to an emitted color $\mathbf{c}$ and density σ. To render the color of a single pixel $\hat{\mathbf{C}}$ in a certain camera view, NeRF samples the volumetric radiance field via ray marching and integrates the sampled density σ_i and color $\mathbf{c_i}$ via numerical integration of the volume rendering equation [17].

$$\hat{\mathbf{C}} = \sum_i \exp\left(-\sum_{j<i} \sigma_j \delta_j\right) \left(1 - \exp\left(-\tau_i \delta_i\right)\right) \mathbf{c}_i \quad (1)$$

where t_i is sample distance at point i, $\delta_i = t_i - t_{i-1}$ is sample interval, $\alpha_i = 1 - \exp\left(-\sigma_i \delta_i\right)$ is alpha value, $T_i = \sum_i \exp\left(-\sum_{j<i} \sigma_j \delta_j\right)$ is accumulated transmittance.

We could also define the ray terminated depth $\hat{D}$ as the weighted accumulated sample distances.

$$\hat{D} = \sum_{i=1}^{N} T_i \alpha_i t_i \quad (2)$$

LiDAR Point Clouds represent 3D spatial information along with intensity measurements. Unlike passive sensors like cameras, LiDAR actively emits laser pulses to measure distances by timing the reflection from surfaces. Ideally, each LiDAR point corresponds to a spot on a 3D surface, but in practice, non-zero laser divergence and waveform discretization [12] can introduce errors such as discretization

and biases in distance estimation. These inaccuracies are influenced by factors like incident angle and distance.

IV. Method

A. Data Preprocessing

To construct a global spatial structure, we then aggreate lidar points and filter out invalid points accoording to intensity and tag [15]. We downsample the global pointcloud and calculate the normal vectors from Singular Value Decomposition(SVD). The normal vector $\mathbf{d}$ of the point corresponds to the eigenvector of the smallest eigenvalue λ_3 of the covariance matrix $\mathbf{A}$ of a set of nearest points.

We use the eigenvalue λ to compute curvature κ and evaluate the quality of the normal vector q. We reckon that the normal vectors estimated near planar regions of the point cloud are more reliable.

$$\overline{\mathbf{p}} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{p}_i; \mathbf{A} = \frac{1}{N}\sum_{i=1}^{N}\left(\mathbf{p}_i - \overline{\mathbf{p}}\right)\left(\mathbf{p}_i - \overline{\mathbf{p}}\right)^T \quad (3)$$

$$q \ = 1 - 3\kappa = 1 - 3\frac{\lambda_3}{\lambda_1+\lambda_2+\lambda_3} \quad (4)$$

We use a sparse spatial structure from [9] to build an octree that supports fast node attribute queries from ray intersections. During training, we first gather a batch of rays and perform ray intersections with the octree. We then query the normal and its quality from the tree nodes and acquire the intersection distances, which provide sampler guidance for faster convergence.

B. Continuous Time Formula

Lidar continuously emits beams at a high frequency, collecting points that are subsequently grouped into a frame over a designated scan cycle or period. As a result, these 3D points are sampled at different times and from varying poses due to the ongoing motion of the LiDAR system. A prevalent deskewing technique in LiDAR SLAM [33] involves projecting all points within a frame, using an initial motion estimate (often sourced from IMU measurements) to align them to the end of that frame. Drawing inspiration from continuous SLAM methods [5] [27], We represent the pose as a function of time. For each LiDAR point, based on its timestamp τ_i, we select the two temporally closest optimizable camera poses for interpolation.

$$\mathbf{R}_{\tau_i} = \operatorname{slerp}\left(\mathbf{R}_k, \mathbf{R}_{k+1}, \alpha_{\tau_i}\right) \quad (5)$$

$$\mathbf{t}_{\tau_i} = (1-\alpha)\,\mathbf{t}_k + \alpha_i \mathbf{t}_{k+1} \quad (6)$$

$$\alpha_{\tau_i} = \left(\tau_i - \tau_k\right) / \left(\tau_{k+1} - \tau_k\right) \quad (7)$$

We obtain the estimated camera pose $(\mathbf{R}_{\tau_i}, \mathbf{t}_{\tau_i})$ at time τ_i. Then, we get the lidar transformations to the world frame by applying the extrinsic camera-to-lidar transformations.

C. Consistent Radiance Field Reconstruction

We present a unified neural field

$$F_\phi : \mathbf{x}, \mathbf{d} \mapsto (\sigma, \rho, \mathbf{n}, \mathbf{c}, \varsigma),$$

where each 3D location $\mathbf{x} \in \mathbb{R}^3$ with view direction $\mathbf{d} \in \mathbb{R}^3$ is mapped to various color $\mathbf{c}$, volume density σ, LiDAR intensity ς, normal $\mathbf{n}$ and roughness ρ.

The neural radiance field is comprised of:

- Two multi-resolution hash feature grids for Lidar and camera encoding Φ_L, Φ_C with total L levels.
- Masked Attentional Fusion Module(MAFM) Φ_θ.
- LiDAR Attribute Encoding(LAE): Two 1D Grids, Distance Encoding Φ_D, Incidence Angle Encoding Φ_I.
- Spatial MLPs ($\gamma(\mathbf{x})$ refers to positional embedding):

$$\begin{aligned} f_{\text{density}} &: \left(\Phi_\theta\left(\Phi_L(\mathbf{x}), \Phi_C(\mathbf{x})\right)\right) \longmapsto (\mathbf{b}, \sigma), \\ f_{\text{norm}} &: (\mathbf{b}, \gamma(\mathbf{x})) \longmapsto \mathbf{n}, \\ f_{\text{roughness}} &: (\mathbf{b}) \longmapsto \rho \end{aligned}$$

- Directional MLPs:

$$\begin{aligned} f_{\text{color}} &: \left(\mathbf{b}, \operatorname{IDE}\left(\hat{\boldsymbol{\omega}}_r, 1/\rho\right), \hat{\mathbf{n}} \cdot \mathbf{d}\right) \longmapsto (\mathbf{b}, \mathbf{c}), \\ f_{\text{intensity}} &: \left(\mathbf{b}, \operatorname{LAE}\left(t, \hat{\mathbf{n}} \cdot \mathbf{d}\right)\right) \longmapsto \varsigma \end{aligned}$$

We use two multi-resolution hash feature grids to capture the details of different modalities. Our Masked Attention Fusion Module works as follows: based on a predefined mask level, we divide the features into two parts, high-resolution features and low-resolution features. High-resolution features tend to overfit the modalities and are therefore more susceptible to noise. Consequently, when learning different modalities, we can mask out each other's high-resolution features to prevent interference in the details, while ensuring consistency at a coarse scale.

$$\Phi_\theta\left(\Phi_L(\mathbf{x}), \Phi_C(\mathbf{x})\right) = \begin{cases} \left(\Phi_L(\mathbf{x}), \Phi_{C\text{-low}}(\mathbf{x}), \mathbf{0}\right), & \mathbf{x} \sim \mathbf{r}_L \\ \left(\Phi_{L\text{-low}}(\mathbf{x}), \mathbf{0}, \Phi_C(\mathbf{x})\right), & \mathbf{x} \sim \mathbf{r}_C \end{cases} \quad (8)$$

However, this approach reduces the utilization rate of LiDAR information and may require manually adjusting the mask level to find the optimal resolution. Therefore, we adopt a method inspired by multi-head attention to compute the similarity between LiDAR and camera features, enabling the adaptive reweighting of high-resolution LiDAR features. Unlike conventional attention mechanisms that use softmax to normalize weights across a sequence of features, our approach operates on individual features to better align LiDAR representations with the camera modality. Hence, we use a sigmoid activation on the scaled dot product, generating outputs in the range of 0 to 1 for each spatial resolution individually. This results in a dynamic mask that effectively filters out inconsistent LiDAR features.

$$\mathbf{F} = \operatorname{sigmoid}\left(\mathbf{K}\mathbf{Q}^T / \sqrt{d_k}\right)\Phi_{L\text{-high}} \quad (9)$$

where $\mathbf{K}$ and $\mathbf{Q}$ are the linear projections of $\Phi_L(\mathbf{x})$ and $\Phi_C(\mathbf{x})$, respectively. $\sqrt{d_k}$ is the feature dimension of

the above projections. Figure 3 demonstrates the detailed calculation. Hence, our fused feature is as follows.

$$\Phi_\theta\left(\Phi_L(\mathbf{x}), \Phi_C(\mathbf{x})\right) = \begin{cases} \left(\Phi_L(\mathbf{x}), \Phi_{C\text{-low}}(\mathbf{x}), \mathbf{0}\right), & \mathbf{x} \sim \mathbf{r}_L \\ \left(\Phi_{L\text{-low}}(\mathbf{x}), \mathbf{F}, \Phi_C(\mathbf{x})\right), & \mathbf{x} \sim \mathbf{r}_C \end{cases} \tag{10}$$

We feed the fused feature into spatial MLPs to obtain spatial attributes. The spatial MLP outputs a bottleneck vector $\mathbf{b}$, which is passed to the following MLPs to convey geometric information.

Similar with [26], we simulate view-dependent effects for color using the rendered normal to compute the reflection direction.

$$\hat{\boldsymbol{\omega}}_r = 2\left(\hat{\boldsymbol{\omega}}_o \cdot \hat{\mathbf{n}}\right)\hat{\mathbf{n}} - \hat{\boldsymbol{\omega}}_o; \hat{\boldsymbol{\omega}}_o = -\mathbf{d} \tag{11}$$

We use roughness ρ to represent the concentration of the von Mises-Fisher (vMF) distribution as introduced by Integrated Directional Encoding (IDE $(\hat{\boldsymbol{\omega}}_r, 1/\rho)$), which can be seen as a form of weighted spherical harmonics encoding.

Unlike color, intensity values also vary with distance. Therefore, we apply LiDAR Attribute Encoding (LAE) to account for this. By normalizing the distance to the viewpoint and the incidence angle, we can query 1D feature grids. LAE is the product of Distance Encoding Φ_D and Incidence Angle Encoding Φ_I.

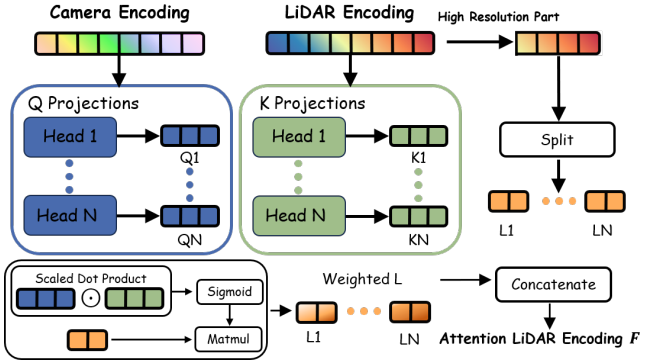

Fig. 3: Calculation of Attention Feature.

D. Optimization

We optimize the consisent neural radiance field end-to-end by minimizing the loss:

$$\begin{aligned} \mathcal{L} = \mathcal{L}_{\text{rgb}} + \lambda_{\text{depth}}\mathcal{L}_{\text{depth}} + \\ \lambda_{\text{intensity}}\mathcal{L}_{\text{intensity}} + \lambda_{\text{normal}}\mathcal{L}_{\text{normal}} \end{aligned} \tag{12}$$

RGB Loss: We optimize the camera radiance field by minimizing a MSE loss between the ground truth $\mathbf{C}$ and the reconstructed color $\hat{\mathbf{C}}$ from Eq.(1) averaged over all training rays.

$$\mathcal{L}_{\text{rgb}} = \sum_{\mathbf{r}\in\mathcal{R}_\mathcal{C}} \|\hat{\mathbf{C}}(\mathbf{r}) - \mathbf{C}(\mathbf{r})\|^2 \tag{13}$$

Lidar Depth Loss: Similarly, for LiDAR rays, we minimize the L1 loss between the LiDAR depth and the expected depth obtained from volume rendering. Furthermore, as in [6], we employ KL divergence to constrain the density distribution $h(t)$ to be focused around the LiDAR points, modeled as a normal distribution $\mathcal{N}(D, \hat{\sigma}^2)$.

$$\mathcal{L}_{\text{depth}} = \sum_{\mathbf{r}\in\mathcal{R}_\mathcal{L}} \|\hat{D}(\mathbf{r}) - D(\mathbf{r})\|_1 + \text{KL}\left[\mathcal{N}(D, \hat{\sigma}) \| h(t)\right] \tag{14}$$

Intensity Loss: .We reconstruct the lidar radiance field by minimizing the L2 loss between reconstructed intensity and the actual intensity readings from the LiDAR.

$$\mathcal{L}_{\text{intensity}} = \sum_{\mathbf{r}\in\mathcal{R}_L} \|\hat{\varsigma}(\mathbf{r}) - \varsigma(\mathbf{r})\|^2 \tag{15}$$

Normal Loss: We obtain the normal at a specific location $\mathbf{x}$ by computing the gradient of the volume density $\tilde{\mathbf{n}}_\mathbf{x} = -\frac{\nabla_\mathbf{x}\sigma(\mathbf{x})}{\|\nabla_\mathbf{x}\sigma(\mathbf{x})\|}$. However, this formulation tends to produce noisy results [31]. Consequently, we use the normal predicted by the network f_{norm}, which is smoother. We then ensure consistency between the predicted normal $\hat{\mathbf{n}}_\mathbf{x}$ and the gradient normal samples along each camera ray $\tilde{\mathbf{n}}_\mathbf{x}$.

$$L_{\text{norm-grad}} = \sum_{\mathbf{r}\in\mathcal{R}_\mathcal{C}} \sum_i w_i \|\tilde{\mathbf{n}}_{\mathbf{x_i}} - \hat{\mathbf{n}}_{\mathbf{x_i}}\|^2 \tag{16}$$

Additionally, we used normals computed from the point cloud $\bar{\mathbf{n}}$ to provide more accurate supervision. Since the normals are retrieved from ray-octree intersections, they may encounter occlusion problems. Thus, we employ a depth mask $\mathcal{M}$ to mask out rays where the difference between the octree depth and the reconstructed LiDAR depth is greater than a threshold.

$$\mathcal{L}_{\text{norm-lidar}} = \sum_{\mathbf{r}\in\mathcal{R}_C\&\mathcal{M}} \|\hat{\mathbf{n}}(\mathbf{r}) - \bar{\mathbf{n}}(\mathbf{r})\|_1 + \left\|1 - \hat{\mathbf{n}}(\mathbf{r})^\top \bar{\mathbf{n}}(\mathbf{r})\right\|_1 \tag{17}$$

$$\mathcal{L}_{\text{normal}} = \mathcal{L}_{\text{norm-grad}} + \mathcal{L}_{\text{norm-lidar}} \tag{18}$$

V. Experiments and Results

A. Experimental Setup

Datasets. We conducted experiments on two datasets: KITTI-360 [13] and R3LIVE [16]. From KITTI-360, we selected five scenes with primarily static objects, each containing around 80 to 100 frames captured at 10Hz, including stereo images and LiDAR point clouds. The R3LIVE dataset, collected using a handheld device for LiDAR SLAM, features solid-state LiDAR with significant overlap between the camera and LiDAR fields of view. We selected scenes from two sequences, hkust_campus_02 and hku_campus_00, each with approximately 1000 frames captured at 8Hz. We used 10 percent of the frames at fixed intervals as the test set, with the rest for training.

Implementation Details. Our code is built on SDFStudio [35]. We train all methods for 30,000 iterations without pose optimization unless specified otherwise. The hash encoding's finest resolution is 8192, with a 16-level hash grid and a mask level set to 8. We sample points near the octree depth and combine them with points from a uniform sampler for the proposal sampler. After two rounds of resampling, each ray has 64 sampling points.

Evaluation Metrics. We assess image quality using three common metrics: PSNR, SSIM [30], and LPIPS [38], following prior works [17], [18]. For geometric quality in LiDAR

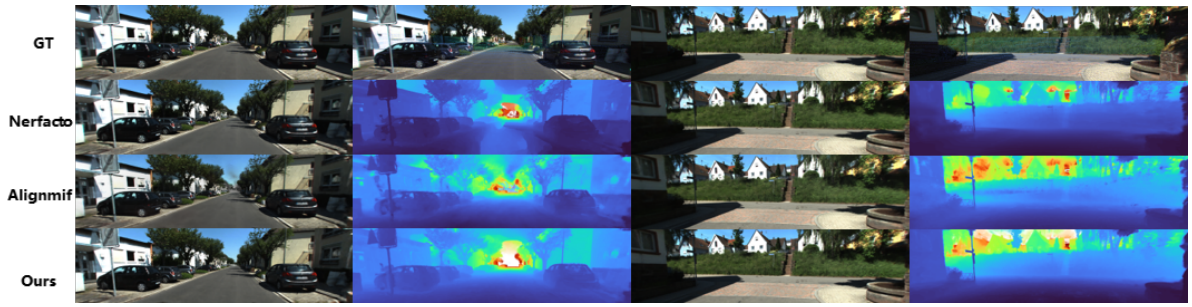

Fig. 4: **Qualitative results on the KITTI-360 Dataset.** Better viewed zoomed-in and in-color.

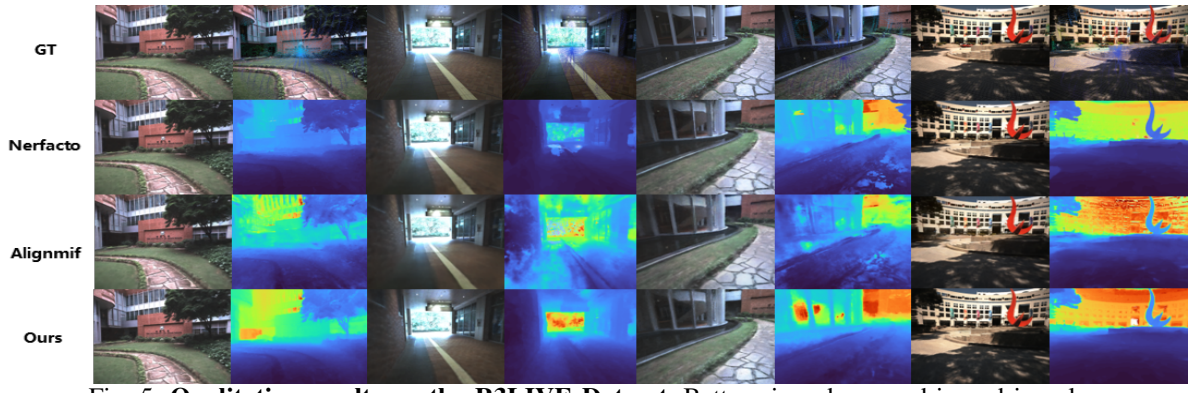

Fig. 5: **Qualitative results on the R3LIVE Dataset.** Better viewed zoomed-in and in-color.

TABLE I: Novel view synthesis results on KITTI-360 dataset and R3LIVE dataset.

		KITTI-360 Dataset						R3LIVE Dataset					
		Camera Metric			LiDAR Metric			Camera Metric			LiDAR Metric		
Method	M	PSNR↑	SSIM↑	LPIPS↓	C-D↓	F-score↑	PSNR↑	PSNR↑	SSIM↑	LPIPS↓	C-D↓	F-score↑	PNSR↑
nerfstudio [23]	C	**24.844**	**0.731**	**0.154**	10.544	0.634	-	**24.371**	**0.673**	0.273	24.406	0.465	-
nerfstudio(D) [23]	LC	24.801	0.725	0.162	10.832	0.545	-	24.359	0.670	0.260	2.569	0.538	-
LightingNeRF [1]	LC	22.171	0.628	0.446	12.449	0.493	-	20.224	0.541	0.571	26.127	0.324	-
AlignMiF [25]	LC	23.733	0.710	0.184	0.145	0.918	19.255	23.031	0.633	0.316	0.492	0.800	17.638
CLC-NeRF (ours)	LC	24.615	0.718	0.168	**0.0661**	**0.935**	**20.025**	24.135	0.649	**0.273**	**0.116**	**0.887**	**18.438**

M, L, C, D denotes modality, LiDAR, camera, Depth-supervised respectively. In the context of LiDAR metrics, PSNR denotes the Intensity metric.

view synthesis, we report Chamfer Distance (C-D) and F-Score (5cm threshold) as in [24], [39]. Additionally, we evaluate the intensity values of the predicted point cloud using PSNR.

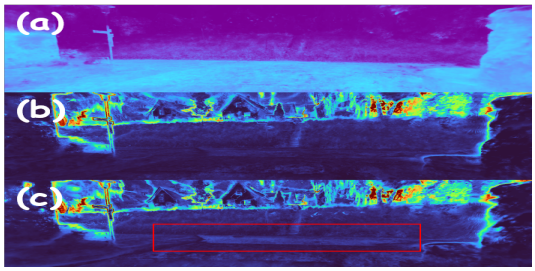

Fig. 6: (a) Attention Map: Brighter areas indicate higher weights for high-resolution LiDAR features. (b) and (c) show Depth Uncertainty with and without the attention mechanism.

B. Novel View Synthesis Results

In Table I, we report quantitative evaluations on both KITTI-360 and R3LIVE datasets. The evaluation includes four cutting-edge NeRF approaches, including Nerfstudio, its variant with single-frame LiDAR depth supervision, AlignMiF, and LightningNeRF. LightningNeRF relies on point cloud initialization for the occupancy grid and explicit grid modeling for density, which limits its expressive capacity and makes it vulnerable to LiDAR noise, particularly in areas lacking point cloud coverage. While Nerfstudio achieved the best RGB metrics, it struggles with precise geometric reconstruction due to its reliance on camera images alone

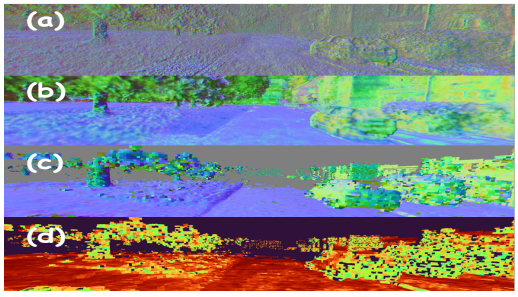

Fig. 7: Visualization of normals. (a) The gradient of volume density. (b) Smoother normals from network prediction. (c) LiDAR normals stored in octree nodes. (d) LiDAR normal quality (Reder is better).

(for fairness, we use median depth for visualization and evaluation since Nerfstudio lacks depth supervision). Sparse depth supervision also contributes minimally to geometric accuracy. Our method delivers image quality on par with Nerfstudio and outperforms AlignMiF in multimodal performance.

Qualitative results in Figure 4, 5, demonstrate that our method improves geometric accuracy, reduces floaters, and eliminates ground holes while maintaining RGB quality. Additionally, in areas with significant LiDAR noise, such as around pillars or poles, our method effectively filters the noise, revealing sharp geometric structures.

C. Ablation Study

Analysis on Network Design. To investigate the contributions of different components, we present LiDAR and camera metrics in Table II. Single-modal features show a significant drop in RGB metrics due to LiDAR noise. By mutually masking high-resolution features, both LiDAR and camera results improve. The MAFM further enhances performance by leveraging consistent high-resolution features between LiDAR and the camera. Figure 6(a) visualizes the attention results, showing higher fusion weights for accurate LiDAR points (e.g., roads) and lower weights for areas with more errors (e.g., grass). As noted in [10], shape-radiance ambiguity stems from a null space in photometric loss optimization. We illustrate this uncertainty using renderer depth variance in Figure 6(a)(b). The reduced uncertainty with MAFM indicates more effective use of LiDAR's geometric information. Incorporating LAE allows accurate modeling of intensity changes across different viewpoints, preventing overfitting.

TABLE II: Ablation Studies of Network Design

Method	Camera Metric		LiDAR Metric		
	PSNR↑	LPIPS↓	C-D↓	F-score↑	PSNR↑
Single Hash Encoding	24.588	0.214	0.0522	0.931	20.137
Masked Hybrid Encoding	25.167	0.199	**0.0445**	**0.937**	**20.170**
wo LAE	25.129	0.206	0.0499	0.933	20.043
Ours(Full)	**25.219**	**0.193**	0.0446	**0.937**	20.135

Analysis on Lidar Supervision. In Table III, we present an ablation study examining the impact of different LiDAR supervision methods. Our approach involves learning both normals and intensity, which adds complexity to the optimization objective. Without LiDAR depth supervision, RGB metrics significantly decline due to overfitting to intensity, resulting in inaccurate scene geometry. Figure 7 shows that the volume density gradient is quite noisy, emphasizing the need for normal supervision.

We enhance the proposal sampler by incorporating LiDAR depth or ray-octree intersection distance to add initial sampling points. Finally, the continuous time formula enables better utilization of LiDAR data, reducing errors and enhancing RGB quality.

Analysis on Continuous Formula. Tables IV and Figure 8 validate the effectiveness of our proposed CTF. In earlier experiments, pose optimization was not employed due to the infeasibility of verifying test set poses. Instead, optimized poses from the training set were used to generate the metrics in Table IV. The results indicate that direct pose optimization without CTF degrades performance due to misalignments between LiDAR and RGB data. While Cubic B-splines offer clearer results than linear interpolation, they require additional computation for fitting initial values, as they do not pass directly through control points. We found that increasing the number of optimizable poses with linear interpolation also yielded favorable results. The "5 knots" refer to five optimizable poses between two camera measurements. Given the complexity of NeRF pose optimization, which may require specific strategies (e.g., coarse-to-fine [14]) for convergence, we suggest further exploration of this topic.

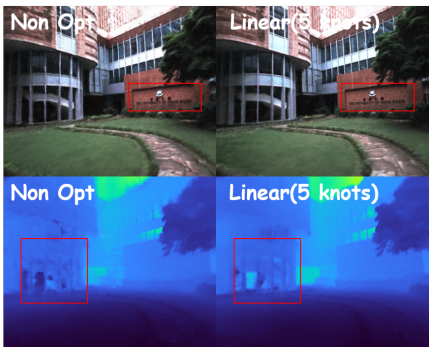

Fig. 8: **Qualitative Comparison of Pose Optimization.**

TABLE III: Ablation Studies of Lidar Supervision

Method	Camera Metric		LiDAR Metric		
	PSNR↑	LPIPS↓	C-D↓	F-score↑	PSNR↑
wo LiDAR Normal Loss	25.202	0.197	0.0500	0.933	20.095
wo Depth Loss	24.781	0.239	76.806	0.104	**20.784**
wo Sample Guide	25.082	0.202	0.0495	0.933	20.121
wo CTF	24.852	0.214	0.0649	0.922	20.012
Ours(Full)	**25.219**	**0.193**	**0.0446**	**0.937**	20.135

TABLE IV: Continuous Time Formulation for Pose Optimization

Method	Camera Metric		LiDAR Metric		
	PSNR↑	LPIPS↓	C-D↓	F-score↑	PSNR↑
Non Optimization	26.329	0.267	0.123	0.918	18.921
Direct Optimization	22.219	0.439	0.126	0.912	18.827
Cubic B-Spline	**26.607**	0.253	0.0291	0.978	20.154
Linear(2 Knots)	24.412	0.306	0.0402	0.971	20.011
Linear(5 Knots)	26.576	**0.246**	**0.0242**	**0.985**	**20.718**

VI. Conclusion

In conclusion, we have developed a NeRF training pipeline that ensures consistent reconstruction across different modalities while preserving their unique characteristics. By leveraging an octree structure, we fully exploit LiDAR data and mitigate inherent noise through our Continuous Time Formula and Masked Attention Fusion Module. Currently, due to hash capacity limitations, our method is restricted to moderately sized scenes. However, we believe the insights from this work can be extended to large-scale environments, enabling realistic simulations in robotic systems.

REFERENCES

[1] Junyi Cao, Zhichao Li, Naiyan Wang, and Chao Ma. Lightning nerf: Efficient hybrid scene representation for autonomous driving. *arXiv preprint arXiv:2403.05907*, 2024.

[2] Alexandra Carlson, Manikandasriram S. Ramanagopal, Nathan Tseng, Matthew Johnson-Roberson, Ram Vasudevan, and Katherine A. Skinner. Cloner: Camera-lidar fusion for occupancy grid-aided neural representations. *IEEE Robotics and Automation Letters*, 8(5):2812–2819, 2023.

[3] MingFang Chang, Akash Sharma, Michael Kaess, and Simon Lucey. Neural radiance field with lidar maps. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 17914–17923, October 2023.

[4] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *European Conference on Computer Vision (ECCV)*, 2022.

[5] Pierre Dellenbach, Jean-Emmanuel Deschaud, Bastien Jacquet, and François Goulette. Ct-icp: Real-time elastic lidar odometry with loop closure. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 5580–5586. IEEE, 2022.

[6] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised nerf: Fewer views and faster training for free. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12872–12881, 2022.

[7] Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct sparse odometry. *IEEE transactions on pattern analysis and machine intelligence*, 40(3):611–625, 2017.

[8] Christian Forster, Matia Pizzoli, and Davide Scaramuzza. Svo: Fast semi-direct monocular visual odometry. In *2014 IEEE international conference on robotics and automation (ICRA)*, pages 15–22. IEEE, 2014.

[9] Clement Fuji Tsang, Maria Shugrina, Jean Francois Lafleche, Towaki Takikawa, Jiehan Wang, Charles Loop, Wenzheng Chen, Krishna Murthy Jatavallabhula, Edward Smith, Artem Rozantsev, Or Perel, Tianchang Shen, Jun Gao, Sanja Fidler, Gavriel State, Jason Gorski, Tommy Xiang, Jianing Li, Michael Li, and Rev Lebaredian. Kaolin: A pytorch library for accelerating 3d deep learning research. `https://github.com/NVIDIAGameWorks/kaolin`, 2022.

[10] Lily Goli, Cody Reading, Silvia Sellán, Alec Jacobson, and Andrea Tagliasacchi. Bayes' rays: Uncertainty quantification for neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20061–20070, 2024.

[11] Jianfei Guo, Nianchen Deng, Xinyang Li, Yeqi Bai, Botian Shi, Chiyu Wang, Chenjing Ding, Dongliang Wang, and Yikang Li. Streetsurf: Extending multi-view implicit surface reconstruction to street views. *arXiv preprint arXiv:2306.04988*, 2023.

[12] Shengyu Huang, Zan Gojcic, Zian Wang, Francis Williams, Yoni Kasten, Sanja Fidler, Konrad Schindler, and Or Litany. Neural lidar fields for novel view synthesis. *arXiv preprint arXiv:2305.01643*, 2023.

[13] 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*, 45(3):3292–3310, 2022.

[14] Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. Barf: Bundle-adjusting neural radiance fields. In *IEEE International Conference on Computer Vision (ICCV)*, 2021.

[15] Jiarong Lin and Fu Zhang. Loam livox: A fast, robust, high-precision lidar odometry and mapping package for lidars of small fov. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3126–3131. IEEE, 2020.

[16] Jiarong Lin and Fu Zhang. R ̂3 live++: A robust, real-time, radiance reconstruction package with a tightly-coupled lidar-inertial-visual state estimator. *arXiv preprint arXiv:2209.03666*, 2022.

[17] 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. *ECCV*, pages 405–421, 2020.

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

[19] 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 (CVPR)*, pages 12932–12942, June 2022.

[20] Barbara Roessle, Jonathan T. Barron, Ben Mildenhall, Pratul P. Srinivasan, and Matthias Nießner. Dense depth priors for neural radiance fields from sparse input views. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022.

[21] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *CVPR*, 2022.

[22] Shanlin Sun, Bingbing Zhuang, Ziyu Jiang, Buyu Liu, Xiaohui Xie, and Manmohan Chandraker. Lidarf: Delving into lidar for neural radiance field on street scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19563–19572, 2024.

[23] Matthew Tancik, Ethan Weber, Evonne Ng, Ruilong Li, Brent Yi, Terrance Wang, Alexander Kristoffersen, Jake Austin, Kamyar Salahi, Abhik Ahuja, et al. Nerfstudio: A modular framework for neural radiance field development. In *ACM SIGGRAPH 2023 Conference Proceedings*, pages 1–12, 2023.

[24] Tang Tao, Longfei Gao, Guangrun Wang, Yixing Lao, Peng Chen, Hengshuang Zhao, Dayang Hao, Xiaodan Liang, Mathieu Salzmann, and Kaicheng Yu. Lidar-nerf: Novel lidar view synthesis via neural radiance fields, 2023.

[25] Tang Tao, Guangrun Wang, Yixing Lao, Peng Chen, Jie Liu, Liang Lin, Kaicheng Yu, and Xiaodan Liang. Alignmif: Geometry-aligned multimodal implicit field for lidar-camera joint synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21230–21240, 2024.

[26] Dor Verbin, Peter Hedman, Ben Mildenhall, Todd Zickler, Jonathan T Barron, and Pratul P Srinivasan. Ref-nerf: Structured view-dependent appearance for neural radiance fields. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5481–5490. IEEE, 2022.

[27] Ignacio Vizzo, Tiziano Guadagnino, Benedikt Mersch, Louis Wiesmann, Jens Behley, and Cyrill Stachniss. Kiss-icp: In defense of point-to-point icp–simple, accurate, and robust registration if done the right way. *IEEE Robotics and Automation Letters*, 8(2):1029–1036, 2023.

[28] Chen Wang, Jiadai Sun, Lina Liu, Chenming Wu, Zhelun Shen, Dayan Wu, Yuchao Dai, and Liangjun Zhang. Digging into depth priors for outdoor neural radiance fields, 2023.

[29] Peng Wang, Yuan Liu, Zhaoxi Chen, Lingjie Liu, Ziwei Liu, Taku Komura, Christian Theobalt, and Wenping Wang. F2-nerf: Fast neural radiance field training with free camera trajectories. *CVPR*, 2023.

[30] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[31] Zian Wang, Tianchang Shen, Jun Gao, Shengyu Huang, Jacob Munkberg, Jon Hasselgren, Zan Gojcic, Wenzheng Chen, and Sanja Fidler. Neural fields meet explicit geometric representations for inverse rendering of urban scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8370–8380, 2023.

[32] Ziyang Xie, Junge Zhang, Wenye Li, Feihu Zhang, and Li Zhang. S-nerf: Neural radiance fields for street views. In *ICLR 2023*, 2023.

[33] Wei Xu and Fu Zhang. Fast-lio: A fast, robust lidar-inertial odometry package by tightly-coupled iterated kalman filter. *IEEE Robotics and Automation Letters*, 6(2):3317–3324, 2021.

[34] Han Yan, Celong Liu, Chao Ma, and Xing Mei. Plenvdb: Memory efficient vdb-based radiance fields for fast training and rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 88–96, 2023.

[35] Zehao Yu, Anpei Chen, Bozidar Antic, Songyou Peng, Apratim Bhattacharyya, Michael Niemeyer, Siyu Tang, Torsten Sattler, and Andreas Geiger. Sdfstudio: A unified framework for surface reconstruction, 2022.

[36] 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.

[37] Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. Nerf++: Analyzing and improving neural radiance fields, 2020.

[38] 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.

[39] Zehan Zheng, Fan Lu, Weiyi Xue, Guang Chen, and Changjun Jiang. Lidar4d: Dynamic neural fields for novel space-time view lidar synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5145–5154, 2024.