Para-Lane: Multi-Lane Dataset Registering Parallel Scans for Benchmarking Novel View Synthesis

Ziqian Ni[1] Sicong Du[1] Zhenghua Hou[1] Chenming Wu[2] Sheng Yang[1]✉
[1]Autonomous Driving Lab, CaiNiao Inc., Alibaba Group [2]Baidu Research
{niziqian.nzq,dusicong.dsc,houzhenghua.houzhe,shengyang}@alibaba-inc.com

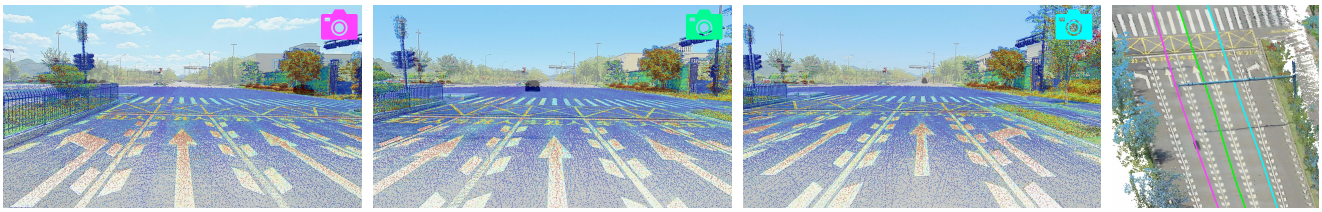

Figure 1. Our work introduces the first real-world multi-lane dataset for evaluating the novel view synthesis capabilities of recent reconstruction approaches for autonomous driving. Public urban roads are scanned using multi-pass trajectories with three laser scanners, a front-view camera, and four surround-view cameras. Frame-wise poses are accurately aligned through LiDAR mapping and multi-modal Structure-from-Motion techniques. Here, we present example images captured from close positions in three aligned cross-lane sequences, with a shared point cloud projected onto the images based on our optimized camera-LiDAR poses.

Abstract

To evaluate end-to-end autonomous driving systems, a simulation environment based on Novel View Synthesis (NVS) techniques is essential, which synthesizes photo-realistic images and point clouds from previously recorded sequences under new vehicle poses, particularly in cross-lane scenarios. Therefore, the development of a multi-lane dataset and benchmark is necessary. While recent synthetic scene-based NVS datasets have been prepared for cross-lane benchmarking, they still lack the realism of captured images and point clouds. To further assess the performance of existing methods based on NeRF and 3DGS, we present the first multi-lane dataset registering parallel scans specifically for novel driving view synthesis dataset derived from real-world scans, comprising 25 groups of associated sequences, including 16,000 front-view images, 64,000 surround-view images, and 16,000 LiDAR frames. All frames are labeled to differentiate moving objects from static elements. Using this dataset, we evaluate the performance of existing approaches in various testing scenarios at different lanes and distances. Additionally, our method provides the solution for solving and assessing the quality of multi-sensor poses for multi-modal data alignment for curating such a dataset in real-world. We plan to continually add new sequences to test the generalization of existing methods across different scenarios. The dataset is released publicly at the project page: https://nizqleo.github.io/paralane-dataset/.

1. Introduction

As a widely investigated technique in 3D vision, novel view synthesis (NVS) [17, 27] is utilized in two primary ways in the development of autonomous driving systems: (1) It facilitates the transfer of data for trained perception or end-to-end models across different vehicle products and sensor configurations [4]; (2) It generates sensor frames with realistic geometry and appearance from various viewpoints for closed-loop simulations, particularly in sensor-to-control scenarios [48].

However, most existing NVS methods in autonomous driving, such as [17, 44], primarily focus on evaluating novel views based on interpolation quality rather than lateral viewpoint shifts, i.e., cross-lane NVS. This is due to the lack of datasets and benchmarks specifically designed for this purpose. As a result, the full potential impact of these new algorithms has not been fully demonstrated. Consequently, current simulation platforms have primarily validated the effectiveness of testing strategic changes in longitudinal (speed) behavior and motion planning, while evalu-

✉: Corresponding author.

ations of lateral (path) planning remain less convincing.

Unfortunately, collecting multi-lane ground truth data within a real-world scene is intrinsically difficult. As a result, XLD [19] chooses to use a simulation platform, Carla [9], to render synthetic data with perfect parameters such as intrinsic, extrinsics, rolling-shutter appearances, and sensor frame poses. They have two primary limitations for further use: First, creating synthetic scenes with high-definition mesh models and materials requires meticulous adjustments for artistic and coherent shading. This process is expensive and must be completed before efficiently generating photo-realistic frames for evaluating driving failure cases on-board. Second, there are still challenging photo-realistic issues, such as inherent noise of real-sensors, and fog-wind-fluid caused dynamics, causing artifacts and domain transfer costs [49]. Therefore, even though sensor data obtained from real recordings may have issues with imprecise parameters, they are an indispensable part of evaluating cross-lane NVS quality. If we aim to collect the data in a single pass, then a super large structure to rigidly mount multiple cameras is required, however, a typical width of a round lane is about 3-4 meters, which is intricate to design and manufacture such a structure with stable dynamics. To tackle this difficulty, our method opts for multi-pass data collection.

This paper focuses on addressing the challenges of creating a real-world dataset using a multi-pass collection scheme for cross-lane NVS benchmarks by tackling the following issues. First, commonly used inertial navigation systems (INS) for obtaining pseudo ground-truth vehicle trajectories in most datasets [11] are insufficient for aligning temporally adjacent sensor frames. This is because the data association between consecutive frames of exterceptive sensors does not directly participate in the maximum-a-posterior pose estimation. Instead, temporal alignment is achieved through dual filtering of the RTK-IMU trajectory, and then composite with extrinsic parameters between IMU and exteroceptive sensors calibrated during end-of-line. Second, pixel-to-point mapping from camera frames to LiDAR frames is imprecise if we rely solely on the trajectory and LiDAR-IMU-cameras extrinsics after pose estimation, which becomes worse when we need to remap across multiple scans. Without establishing and resolving cross-modal feature correspondences between exteroceptive sensors [30, 35], the mapping lacks accuracy. We have addressed these two issues through our two-phase pose optimization (Sec. 3.3).

We develop a unified framework to construct the Para-Lane dataset, featuring a two-phase pose optimization mechanism for aligning data from exteroceptive sensors both temporally and spatially. Additionally, we implemented an autonomous system equipped with LiDAR and camera sensors to capture data, which is then processed using our proposed framework for cross-modal alignment. As the first real-world dataset for cross-lane scenarios, we benchmark mainstream methods for driving NVS based on either NeRF or 3DGS. Our findings could inspire further research and enhance end-to-end driving simulations, and the dataset will be released publicly, ultimately and hopefully accelerating the research and development of autonomous driving products. In summary, our work offers three key contributions:

- We curate the first real-world cross-lane dataset, dubbed Para-Lane, for evaluating NVS capabilities. It includes ample LiDAR, front-view, and surround-view camera data. Our dataset ensures that all sequences are annotated and grouped for easier benchmarking.
- We propose a two-stage framework to precisely align exteroceptive sensors using cross-modal correspondences, demonstrating effectiveness in alignment metrics.
- We evaluate recent NeRF and 3DGS methods, including those designed for autonomous driving scenes, on our curated dataset, offering insights into NVS performance with lateral viewpoint shifts.

2. Related Work

2.1. Autonomous Driving Datasets for NVS

In autonomous driving, there exists a number of available datasets, such as KITTI [11], KITTI-360 [24], CityScapes [7], Waymo Open Dataset [37], nuScenes [3], LiDAR-CS [10] and WayveScenes101 [53]. In decades, they are regarded as the foundation of numerous autonomous driving solutions and algorithms [4, 13, 23]. However, the community lacks datasets that are specifically or compatible to evaluate NVS tasks, due to the booming requirements of end-to-end autonomous driving research. A very recent work, Open MARS Dataset [21], also features multiple laser scanners and cameras for driving scenes. However, their focus is performing collaboratively multi-agent and multi-traversal data collection, with an emphasis on obtaining spatially nearby sequences, instead of multi-lane sequences. Moreover, the multi-pass frame registration algorithm is not disclosed in their work. The XLD [19] dataset, introduced earlier this year, serves as a reliable resource based on synthetic scenes and rendering. However, as discussed in Sec. 1, we argue that real-world datasets are more essential and reliable for comprehensive evaluation, though there are many challenges to curate such a dataset.

2.2. Multi-Sensor Data Alignment

We categorize the multi-sensor dataset alignment into single-modal (camera or LiDAR) and multi-modal (camera and LiDAR) alignments, both of which are crucial for creating a high-quality real-world dataset for NVS evaluation.

Ensuring precise alignment of data from multiple sensors

is essential. While a unified multi-sensor SLAM framework can achieve this, the industrial community typically handles these steps separately for large-scale scene production. For Level-4 unmanned vehicles, during High-definition Map (HDMap) production, LiDAR mapping has been a well-established process [47] and is now standard for mass production and vehicle deployment. For cameras, Structure-from-Motion (SfM) [31, 34] and multi-view stereo [29, 43] are commonly used for solving frame poses and performing dense geometric reconstruction.

To address the cross-modal data association issue—linking cameras and LiDAR—we need to establish explicit correspondences between LiDAR points and camera pixels for a densely coupled pose estimation. There are two categories of methods based on their inputs: the first category operates between LiDAR frames and camera frames [16], which is inefficient for generating large sequences. Therefore, assuming LiDAR mapping provides a sufficiently accurate trajectory, the second category—our chosen approach—works between LiDAR sequences and camera frames [22, 52]. While some methods primarily use ray-casted depth information, we found that incorporating the intensity channel from LiDAR measurements is beneficial. This allows for the use of both sparse and dense photometric loss alongside geometric loss, as demonstrated in various RGB-D reconstruction methods [8, 41]. To establish photometric loss, we identified Normalized Information Distance (NID) [20, 30] as the most effective metric for linking these two types of channels, which can assess the quality of multi-modal registration.

2.3. NeRF and 3DGS Approaches for NVS

Beyond the classical dense reconstruction methods that utilize Truncated Signed Distance Function (TSDF) [8] and Surfels [41] to represent large-scale scenes. NeRF [27] and 3DGS [17] methods are profound innovations in the enhancement of representing geometric and appearance details, respectively. For instance, Block-NeRF [38] is a pioneering work that addresses the reconstruction of large-scale urban scenes through division. MARS [42] is a modular, instance-aware simulator built on NeRF, which separately models dynamic foreground instances and static background environments. UniSim [46] converts recorded logs into realistic closed-loop multi-sensor simulations, incorporating dynamic object priors and using a convolutional network to address unseen regions. Rather than implicitly representing scenes (NeRF) that lack of flexibility of editing and labeling those reconstructed assets, explicitly representing scenes (3DGS) has aroused the interest of many industrial autonomous driving teams [44] to operate on their domain-specific database. Our dataset and benchmark scanned from the real world are specifically designed to evaluate the performance of cross-lane NVS methods.

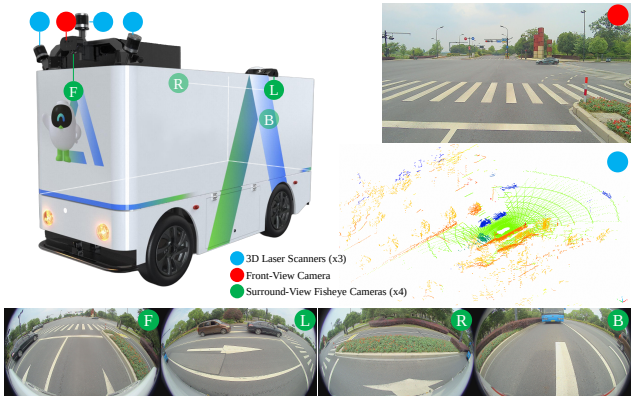

Figure 2. Sensor assembly and sample frames of our data collection unmanned vehicle, the right fisheye camera is mounted symmetrically opposite to the left fisheye, and the back fisheye is located at the center of the back-side.

Some of these methods [5, 6, 14, 17, 26, 44], considering the availability of their code, are used as baselines in our evaluation.

3. Curation of Para-Lane Dataset

3.1. Sensor Setup

To collect and process raw data for our Para-Lane dataset, we implemented an autonomous system equipped with various sensors for street operation. Specifically, to meet the input data specifications for NVS approaches, we installed one front-view camera with a 90° Field-of-View (FOV) capturing 1920×1080 images at 10Hz, along with four surround-view cameras featuring 190° fisheye lenses capturing 1920×1080 images at 10Hz. Additionally, we used three 3D laser scanners with 32 LiDAR channels from $+15^\circ$ to -55° on the vertical FOV capturing at 10Hz. All frame timestamps are synchronized at the hardware level, and we combine points from the three laser scanners into a single LiDAR frame after motion compensation.

Fig. 2 illustrates our autonomous hardware and its assembly, and we refer readers to our accompanying calibration parameters in the dataset for more details. Besides the sensors depicted in the figure, we have additional sensors inside, such as an Inertial Navigation System (INS) for obtaining a high-quality initial trajectory prior to data alignment. In our dataset, we provide all necessary relative transformations between grouped sequences and a reference coordinate system.

3.2. Multi-pass Data Acquisition and Processing

We selected clear sunny days with uncongested road conditions to drive through each parallel lane in the same direction. The data was collected in an anonymous city (details will be disclosed after the review period), consisting of

75 sequences grouped into 25 scenes. Each scene includes three sequences from different lanes, sharing the same start and end positions orthogonal to the road direction, covering approximately 150 meters. As our collection vehicle traveled on public roads, its speed fluctuated, resulting in sequence durations ranging from 10 to 45 seconds, with a median of 20 seconds.

After data collection, we anonymized the images by obscuring vehicle license plates and pedestrian faces. We employed SAM [18], followed by a manual quality inspection, to segment dynamic elements from static foregrounds and backgrounds, ensuring the dataset is free of ethical issues.

3.3. Two-phase Pose Optimization

Phase 1: LiDAR mapping. Given the initial trajectory from the RTK/INS sensor, we apply [47] to construct a LiDAR map $\mathcal{L}^{\mathbb{G}}$ in a reference coordinate $\mathbb{G}$, which involves two components: odometry and loop refinement. We focus on the first component—offline LiDAR odometry—by implementing an enhanced offline LIO system [33] that utilizes both LOAM features [50] and dense scan-to-submap point cloud registration [32] for robustness across various mapping scenarios. For the second component, loop closure and pose graph optimization, we begin with the coarse matching of submaps using Predator [15] and SuperLine3D [51]. We then evaluate the matching scores to select candidate submap pairs for fine registration [32], establishing precise relative constraints for the pose graph optimization problem. Relative poses between sequences are determined through multi-pass loop closure associations and joint optimization.

To verify the quality of LiDAR mapping, we use the thickness of structural objects as a reference metric. We first perform mesh reconstruction using VDBFusion [39] and Marching Cubes [25] with the solved LiDAR frame poses (both voxel size and truncation distance set to 5 cm) to create a triangle mesh representing the maximum-a-posteriori locations of watertight object surfaces. We then calculate the Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) [28] by comparing the mesh with the stitched point cloud to assess thickness. Tab. 1 and Fig. 3 demonstrate the effectiveness of achieving a thinner LiDAR map after the combination of multiple scaned sequences.

Phase 2: Registering images to the LiDAR map. After obtaining the LiDAR map $\mathcal{L}^{\mathbb{G}}$ and the pose of each LiDAR frame, we register the images $\bigcup_i\{\mathbf{C}_i\}$ captured by our camera sensors to $\mathcal{L}^{\mathbb{G}}$. This involves determining the pose of each camera frame $\mathbf{C}_i$ relative to the reference coordinate, denoted as $\mathbf{T}^{\mathbb{G}}_{\mathbb{C}_i}$. We begin by coarsely initializing these poses using linear-slerp interpolation [2] between the two adjacent LiDAR frames and an extrinsic parameter.

We then refine the coarse camera poses by formulating the following factor graph optimization problem [12]

Table 1. Quantitative metrics for LiDAR mapping. We choose to sample and evaluate the MAE and RMSE of stitched LiDAR frames (in centimeters).

	MAE ↓				RMSE ↓			
Metrics	Avg.	90^{th}	95^{th}	99^{th}	Avg.	90^{th}	95^{th}	99^{th}
Before	4.5	5.8	5.9	7.3	7.4	9.8	10.3	12.6
After	**1.3**	**1.4**	**1.4**	**1.7**	**2.9**	**3.6**	**4.1**	**7.1**

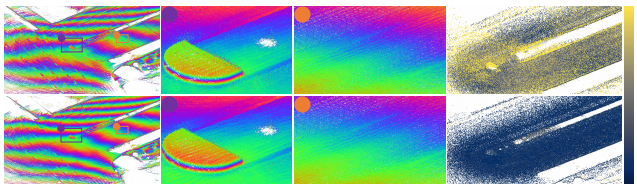

Figure 3. LiDAR map stitching quality visualized in both 20cm periodical height ramp in rainbow (left columns) and 10cm cividis colormap reflecting distance with their reconstructed mesh (the right column). Both the error map and zoomed-in views reflect that these refined LiDAR frame poses (the second row), compared to the initial RTK trajectory (the first row), have achieved a thinner stitched cloud with fewer hovering noisy points due to better frame poses.

to optimize all camera poses $\mathcal{C}^{\mathbb{G}} \triangleq \bigcup_i\{\mathbf{T}^{\mathbb{G}}_{\mathbb{C}_i}\}$ and the 3D positions of visual landmarks $\mathcal{P}^{\mathbb{G}} \triangleq \bigcup_j\{\mathbf{p}^{\mathbb{G}}_j\}$ using an expectation-maximization (EM) strategy [1]:

$$\min_{\mathcal{C}^{\mathbb{G}},\mathcal{P}^{\mathbb{G}}} \mathbf{E}^1(\mathbf{C}_i,\mathbf{p}_j) + \mathbf{E}^2(\mathbf{p}_j,\mathcal{L}^{\mathbb{G}}) + \mathbf{E}^3(\mathbf{C}_i,\mathbf{p}_j,\mathbf{L}_i), \quad (1)$$

where $\mathbf{E}(a,b[,c]) = \sum -\log(\mathbf{f}_{a,b[,c]})$ is the sum of negative log-likelihood of a type of valid factor constraints $\mathbf{f}$, making their scale factors become irrelevant constants. $\mathbf{L}_i \triangleq \pi((\mathbf{T}^{\mathbb{G}}_{\mathbb{C}_i})^{-1}\cdot\mathcal{L}^{\mathbb{G}})$ is the intensity image ray-casted from $\mathcal{L}^{\mathbb{G}}$ at $\mathbf{T}^{\mathbb{G}}_{\mathbb{C}_i}$. Specifically, these factors $\{\mathbf{f}\}$ are designed as:

$\mathbf{f}^1$: Structure-from-motion (SfM) for cameras. We use SuperGlue [40] to associate multiple camera frames with joint observations on sparse features. We formulate this factor using the reprojection error [31], which quantifies the error by projecting the 3D positions of landmarks $\mathbf{p}^{\mathbb{G}}_j$ onto the image plane of the corresponding frame pose $\mathbf{T}^{\mathbb{G}}_{\mathbb{C}_i}$:

$$\mathbf{f}^1_{\mathbf{C}_i,\mathbf{p}_j} \propto \exp(-\frac{1}{2}\|\hat{v}_j - \mathbf{K}((\mathbf{T}^{\mathbb{G}}_{\mathbb{C}_i})^{-1}\cdot\mathbf{p}^{\mathbb{G}}_j)\|^2_{\Omega_1}), \quad (2)$$

where $\|\mathbf{e}\|^2_\Omega \triangleq \mathbf{e}^\top\Omega^{-1}\mathbf{e}$ represents the squared Mahalanobis distance with the covariance matrix Ω. Here, $\hat{v}_j$ is the corresponding pixel observed in $\mathbf{C}_i$, and $\mathbf{K}(\cdot)$ denotes the perspective projection and rectification function with respect to the intrinsic and distortion parameters.

$\mathbf{f}^2$: SfM points and LiDAR map registration factors. Longitudinal driving sequences often lack sufficient parallax to accurately estimate depth for the sparse landmarks

$\mathcal{P}^{\mathbb{G}}$. Therefore, we constrain the absolute positions of these landmarks once they are effectively associated with nearby LiDAR points. The unary prior factor for each landmark is defined as:

$$\mathbf{f}^2_{\mathbf{p}_j,\mathcal{L}^{\mathbb{G}}} \propto \exp(-\frac{1}{2}\|\mathbf{p}_j^{\mathbb{G}} - \Psi(\mathcal{L}^{\mathbb{G}}, \mathbf{p}_j^{\mathbb{G}})\|^2_{\Omega_2}). \tag{3}$$

We found that the choice between point-to-point and point-to-plane formulations is not critically important; however, the point-picking strategy $\Psi(\cdot)$ is significant. Specifically, we select a group of candidate 3D LiDAR points based on the current nearest search for $\mathbf{p}_j^{\mathbb{G}}$, and calculate the tangential distance between these points and the rays emitted from multiple camera observations $\mathcal{R}(\mathbb{C}_i, \hat{v}_j^{\mathbb{C}_i})$ as a joint weight for the final prior position.

$\mathbf{f}^3$: Cross-modal sparse and dense factors. Constructing linkage among sparse 3D points through the previous factor $\mathbf{f}^2$, in our scenario, is not sufficient enough for a camera-pixel to LiDAR-point level data integration. Inspired by the tightly-coupled multi-modal registration used in RGB-D reconstruction [8], we use a combined sparse-and-dense factor, with SuperGlue [40] again for providing sparse constraints, and we define cross-modal photometric error [36] for dense constraints as:

$$\mathbf{f}^3_{\mathbf{C}_i,\mathbf{p}_j,\mathbf{L}_i} \propto \exp(-\frac{1}{2}\|\hat{w}_j - \mathbf{K}((\mathbf{T}^{\mathbb{G}}_{\mathbb{C}_i})^{-1} \cdot \mathbf{p}_j^{\mathbb{G}})\|^2_{\Omega_3^s} \tag{4}$$

$$-\frac{1}{2}\|\mathbf{C}_i \ominus \mathbf{L}_i\|^2_{\Omega_3^d}), \tag{5}$$

where the first half focuses on maintaining sparse relationships between input and ray-casted frames, while the second half addresses dense matching. To establish sparse correspondences between ray-casted feature points $\hat{w}_j$ and SfM points $\mathbf{p}_j^{\mathbb{G}}$, we utilize feature points extracted and matched in $\mathbf{C}_i$ to connect these 2D-3D relationships. The operator $\ominus$ represents $E(\xi)$ as defined in Equation 11 of [36], which employs image gradients to slightly adjust the camera pose.

Parameters. We set the number of Superglue correspondences to 500 for both $\mathbf{f}^1$ and $\mathbf{f}^3$. We use parameters $\Omega_1 = 1$, $\Omega_2 = 0.3 \cdot \mathbf{I}_3$, $\Omega_3^s = 1$, and $\Omega_3^d = 0.01$ to perform the maximization step using Levenberg-Marquardt (LM) with 50 iterations. During optimization, we recalculate the point picking strategy $\Psi(\cdot)$ in $\mathbf{f}^2$ and the ray-casted image $\mathbf{L}_i$ from $\pi(\cdot)$ in $\mathbf{f}^2$ once every 3 iterations.

Evaluation and Ablation Study. To evaluate the quality of multi-camera alignment, we utilize the Normalized Information Distance (NID) proposed in Equation 3 of [30] as a quantitative metric. For each pair of original grayscale images and LiDAR intensity images rendered from a given pose, we compute an initial NID by discretizing the continuous pixel values into 64 bins. We then slightly shift the original image along the UV coordinates to identify a position that yields a local minimum of the NID. The offset used to find this local minimum is defined as NID-Loss.

Table 2. Quantitative metrics for pose estimation. We choose the reprojection error (in pixel) as a common SfM metric [31] to evaluate coherency between camera frames, and the NID-loss [30] to evaluate coherency between camera frames and the LiDAR map.

Metric	Init. $\mathbf{T}^{\mathbb{G}}_{\mathbb{C}_i}$	Opt. $\mathbf{f}^1$	Opt. $\mathbf{f}^{1-2}$	Opt. $\mathbf{f}^{1-3}$
Reproj. Err. ↓	7.959	**1.342**	1.344	1.343
NID-Loss ↓	5.32	10.52	4.47	**4.10**

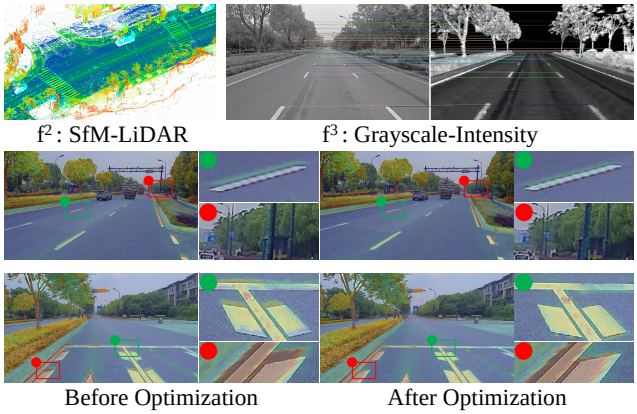

Figure 4. Factors used in our cross-modal pose optimization framework, and we visualize LiDAR-camera alignment quality through an alpha-blending of the colorized intensity map onto its corresponding camera frame. We refer readers to our supplementary video for the data alignment quality of our LiDAR map and multiple camera frames.

We present the results of the ablation study in Tab. 2 and Fig. 4. These results demonstrate that the classical SfM framework ($\mathbf{f}^1$) can jointly solve poses between camera frames to ensure single-modal coherency. However, the lack of correspondence to the LiDAR map $\mathcal{L}^{\mathbb{G}}$ undermines multi-modal coherency. Therefore, an effective approach within the SfM framework is to construct the paired relationship between Visual-SfM points and their corresponding LiDAR points, providing a strong depth prior for triangulation—especially for landmarks established with a narrow baseline (e.g., landmarks on the road or curb while vehicles drive straight). Moreover, photometric factors based on 2D input and ray-casted images can further enhance multi-modal coherency, as our experiments indicate. This is because pixel-level correspondences and gradients operating on grayscale and intensity images, similar to depth map contours used for texture mapping [52], directly utilize raw input channels from different sensors.

4. Benchmark

Based on our scanners, scanning-pattern, and pose optimization procedures discussed in Sec. 3, we have prepared grouped sequences for benchmarking NVS methods on multi-lane scenarios. First of all, we discuss on implementation details of our selected NVS methods in Sec. 4.1, and then share our protocols and metrics chosen for cross-lane NVS application in Sec. 4.2.

4.1. Benchmarking Methods

We select a range of methods, specifically those designed for autonomous driving datasets and based on either NeRF or 3DGS, as our benchmarking methods. For all methods discussed below, we use the combination of LiDAR $\mathcal{L}^{\mathbb{G}}$ and SfM $\mathcal{P}^{\mathbb{G}}$ points to initialize the Gaussians. In order to eliminate the influence of dynamic objects, we filter out all the cars and pedestrians in point clouds and images based on semantic labels. Since the original 3DGS and most of its extensions only support focal points at the absolute center, we rectify the focal point to the center of corresponding images during the post-processing of our dataset.

3DGS [17]: We utilize the implementation from the official release to evaluate our dataset and employ the AdamW optimizer with a learning rate of 10^{-3}. The model is trained for 30,000 steps.

GaussianPro [6]: We train the models for 30,000 iterations across all scenes, adhering to the original training schedule and hyperparameters. The interval step for the progressive propagation strategy is set to 20, with propagation performed three times.

Scaffold-GS [26]: Both the appearance and feature dimensions in the MLP are set to 32, and voxel size is set to 0.005. We adjust the initial and hierarchy factors for anchor growing to 16 and 4, respectively. The model is trained for 30,000 steps.

2DGS [14]: We keep most parameters consistent with the original implementation. For densification, we adjust the gradient threshold to 3×10^{-4} and set the final densification iterations to 13,000. The model is trained for 30,000 steps.

Street Gaussians [44]: To ensure a fair comparison with other methods, we omit the handling of dynamic objects from the original scene graph method. Additionally, we do not utilize the sky mask for an equitable comparison. The parameters remain consistent with those in the official implementation.

PVG [5]: We utilize the Adam optimizer and keep a comparable learning rate for most parameters, consistent with the original implementation. We adjust the gradient threshold to 3×10^{-4} and set the final densification iterations to 13,000. The maximum number of Gaussian spheres is configured to 10^6. We omit multi-resolution downsampling, using images at their original resolution for training.

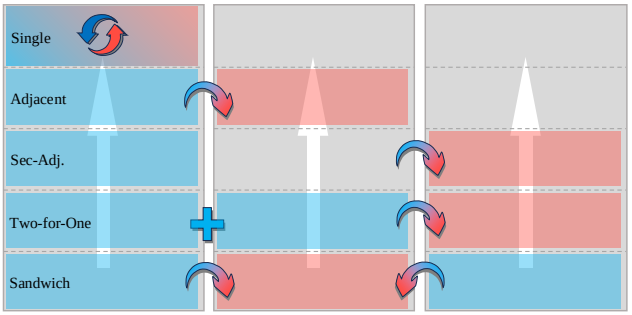

Figure 5. Five evaluation tracks using different combinations of lanes for training (colored in blue) and testing (colored in red).

EmerNeRF [45]: We train EmerNeRF for 30,000 iterations using its original parameters. The flow branch and temporal interpolation are activated, with both the feature levels of the hash encoder for the static and dynamic branches set to 4.

4.2. Experimental Protocols and Metrics

To perform a comprehensive benchmark of all the aforementioned methods on our proposed dataset, we meticulously group all sequences and organize the benchmarks across five different tracks for each method. Specifically, the tracks are categorized as follows: (1) Single lane regression, (2) Adjacent lane prediction, (3) Second-adjacent lane prediction, (4) Adjacent lane prediction (trained from two lanes), and (5) Sandwich lane prediction (trained from two side lanes). A figure illustrating the experimental protocols is provided in Fig. 5. For each track, we uniformly sample 200 frames from training sequences for model learning and 25 frames from test sequences as the ground truth.

We adhere to the widely used metrics for evaluating the performance of NVS as outlined in [19], which includes Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM), and Learned Perceptual Image Patch Similarity (LPIPS).

4.3. Experimental Results and Notes

We benchmark the methods described in Sec. 4.1 according to the framework outlined in Sec. 4.2 to evaluate performance for cross-lane NVS. A series of experiments were conducted using an NVIDIA GeForce RTX 3090 24GB GPU. For quantitative metrics across all tested methods and different designs, refer to Tab. 3. Throughout the experiments, we discovered several interesting insights:

The quality of NVS is significantly affected by the view distribution of the training set. From Tab. 3, we found *exactly* the same conclusion for all methods: the performance gradually decreases in the following sequence: Single $>$ Sandwich $>$ Two-for-One $>$ Adjacent $>$ Second-Adjacent. When the training and testing views are on the same tra-

Table 3. Quantitative results of different Gaussian reconstruction methods on our proposed Para-Lane dataset.

Method	Single			Adjacent			Sec-Adj.			Two-for-One			Sandwich		
Metrics	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
3DGS	**22.99**	**0.689**	0.344	17.05	0.524	**0.446**	16.26	0.505	0.472	17.85	0.551	0.440	18.74	0.563	0.424
GaussianPro	22.93	0.687	**0.343**	17.01	0.521	**0.446**	16.29	0.505	0.472	17.83	0.551	0.439	18.66	0.562	0.424
Scaffold-GS	22.96	0.675	0.364	**17.59**	**0.538**	0.450	**17.09**	**0.525**	**0.470**	**18.62**	**0.565**	**0.437**	**19.20**	**0.574**	**0.423**
2DGS	22.29	0.651	0.395	16.79	0.523	0.469	16.01	0.510	0.494	17.46	0.548	0.466	19.04	0.572	0.451
Street Gaussians	22.56	0.643	0.353	17.50	0.510	0.456	16.16	0.496	0.480	17.87	0.555	0.453	18.91	0.561	0.443

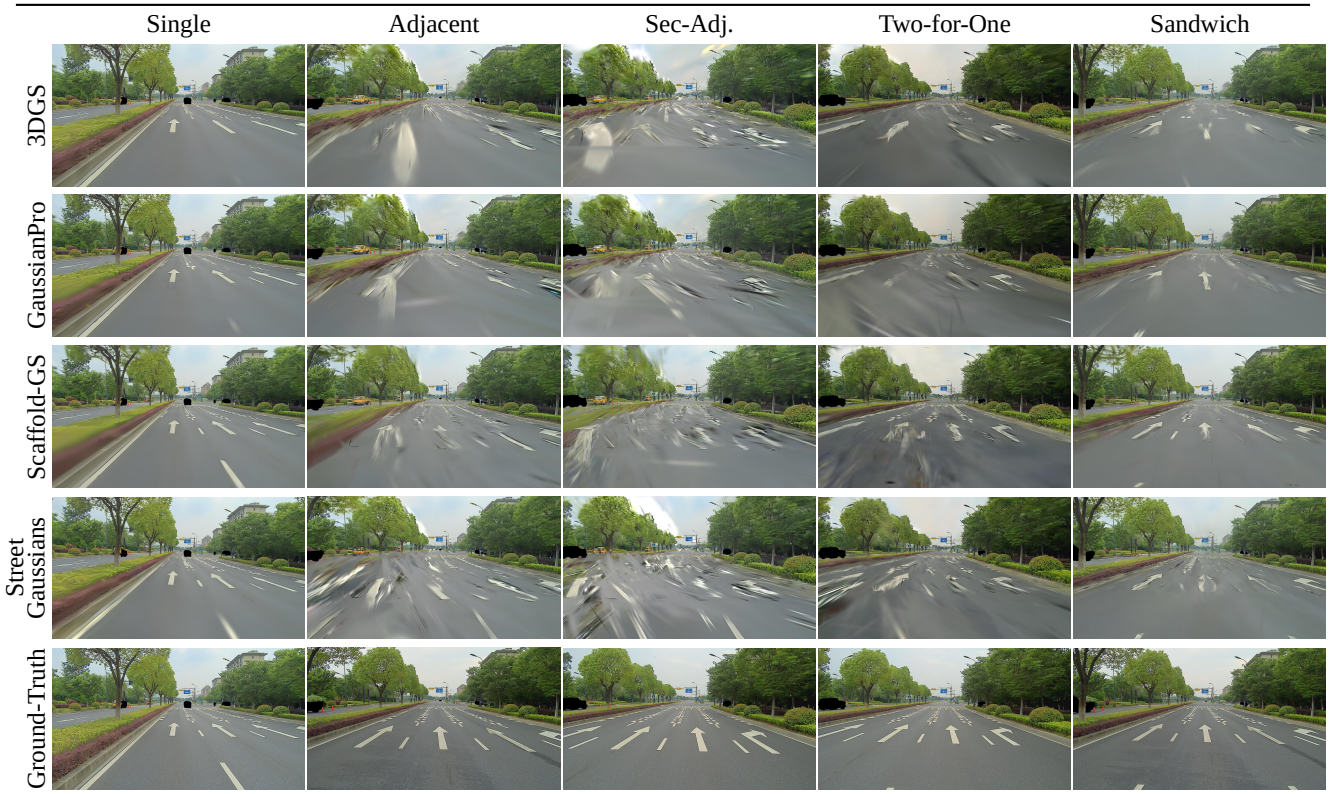

Figure 6. Comparisons for NVS quality between different methods and designs, see our supplementary video for results on more sequences.

jectory, all methods achieve the best NVS results. However, when the testing viewpoint undergoes lateral shifts, the results are compromised to varying degrees. Besides, although the number of images used for training is the same, our Sandwich track, which evenly distributes training views on both sides of the test views, consistently achieves superior rendering quality compared to the Two-for-One track, where training views are located only on one side of the testing views. This can be attributed to the fact that when training data is more evenly distributed and closer to the target render pose, the learned radiance functions are less likely to overfit to a specific viewpoint. Therefore, from an application perspective, to generate images of a target scene from arbitrary viewpoints, it is advisable to utilize multiple passes of data to reconstruct the target scene, thereby minimizing potential artifacts during novel view synthesis. Fig. 6 presents a visual comparison of novel view synthesis results under different designs, and our supplementary video provides additional examples.

Domain gap between synthetic [19] and real datasets. Most of the methods tested here were also evaluated on the XLD dataset [19], a synthetic cross-lane dataset; however, they generally did not perform as well on our dataset as they did on XLD. We attribute this difference to the domain gaps between synthetic and real-world data for several reasons. Firstly, synthetic data perfectly ensures the accuracy of all parameters, including extrinsic, intrinsic, and timestamps. However, in the real world, despite the optimizations discussed in Sec. 3.3, there inevitably remains a gap between our final estimations and the ground truth values. Secondly, the sequences in a group were collected over different time intervals, leading to minor variations in brightness, water stain shapes, and other trivial factors among the cross-lane sequences. This also brings errors that require NVS ap-

proaches to handle. Finally, in real-world data, special attention must be given to dynamic objects. Unlike synthetic data, we cannot capture observations of the same dynamic object simultaneously across different locations. Although we used SAM to mask dynamic objects, the model's output is not always precise, and some noise remains.

Scaffold-GS achieves the best NVS performance on our benchmark. Unlike other methods, Scaffold-GS [26] utilizes anchor points to distribute 3D Gaussians. Each anchor point is associated with multiple neural Gaussians. The attributes of these neural Gaussians—such as position, opacity, quaternion, scaling, and color—are determined by a multi-layer perceptron (MLP). The input features for the MLP include the relative poses from the anchor points to the camera views. Our cross-lane experimental results demonstrate that this method, which establishes correlations between view poses and rendering outcomes, is effective.

4.4. Handling Dynamic Objects

Besides the main experiment in Sec. 4.3 that reflects performance of methods in masked static scenes, we perform another experiment on those experiments capable of handling dynamic objects.

EmerNeRF [45] and PVG [5] are two representative methods that contain procedures on handling dynamic objects in a self-supervised manner, we compare the two methods using the Single track. We perform reconstruction through both methods with and without automatically labeled mask [18] The dataset used in this part includes 6 groups, which are part of the entire 25 groups dataset.

Results and analysis. From Tab. 4, we can find that EmerNeRF achieves better PSNR and LPIPS scores. This is probably because of its novel and effective approach of static-dynamic decomposition. On the other hand, PVG excels in SSIM for both tests. This exceptional performance is likely to due to the adaptable design for Gaussian points. Qualitative results are shown in Fig. 7. We can find that the two methods are comparable for nearby scenes. However, PVG results are inferior for distant locations, such as buildings and trees. This is likely due to that PVG advocates to utilize the larger points for faraway scenes as described in its draft [5], which results in inadequate expression of details and cause blur.

4.5. Limitations

This section describes the limitations of our current dataset and benchmark. The dynamic object masks provided in the dataset are not manually labeled, so there are a small number of omissions and mislabeling. We also do not yet have 3D bounding box and tracking labels for all dynamic objects. The diversity of the dataset can also be enhanced by collecting more data in the future. Given the fact that current works rarely support dynamic/static decomposition, we only tested EmerNeRF [45] and PVG [5] in handling dynamic objects. A more comprehensive benchmark for dynamic scenes would be beneficial as more dynamic methods are developed in the future.

Method	Single		
Metrics	PSNR↑	SSIM↑	LPIPS↓
EmerNeRF	**23.67**	0.668	**0.350**
PVG	22.08	**0.672**	0.408
EmerNeRF (static only)	**23.76**	0.678	**0.346**
PVG (static only)	22.77	**0.684**	0.368

Table 4. Quantitative results on our proposed Para-Lane dataset with dynamic objects. We perform reconstruction with and without mask for ablation study.

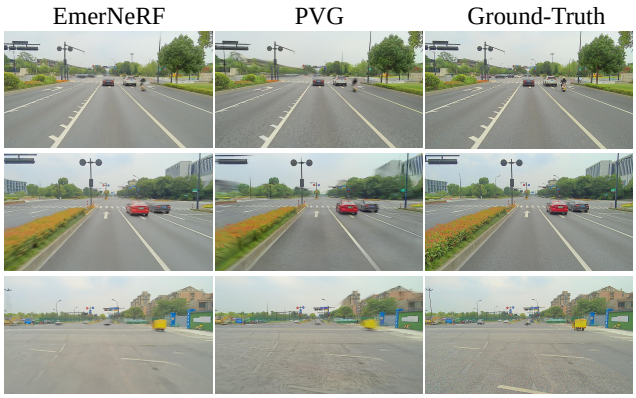

Figure 7. Comparisons for NVS quality in Single lane test between EmerNeRF and PVG.

5. Conclusion and Future Work

In conclusion, we provide a two-stage framework that first registers multi-pass LiDAR frames to form a coherent map, and then registers camera frames to the LiDAR map for the multi-modal pose estimation. We use the provided method to produce reliable poses for multiple grouped parallel lane sequences, and test the performance of recent approaches for synthesizing novel views through longitudinal and lateral viewpoint shifts.

In the future, we plan to expand our dataset with a variety of sequences and incorporate the latest approaches for thorough benchmarking. While we have successfully aligned fisheye camera frames to the LiDAR map, we have also identified limitations in recent works in jointly reconstructing with them. Handling such an issue presents an opportunity for the application of these methods in the industrial community, particularly for cost-effective labeling and mass production for closed-loop simulation scenarios. We hope that the dataset we have released will facilitate future research in this area.

Acknowledgements

We thank the reviewers for the valuable discussions and our colleagues for preparing the proposed dataset. This research was supported by the Zhejiang Provincial Natural Science Foundation of China under Grant No. LD24F030001.

References

[1] Sean L. Bowman, Nikolay Atanasov, Kostas Daniilidis, and George J. Pappas. Probabilistic data association for semantic SLAM. In *ICRA*, pages 1722–1729, 2017. 4

[2] Samuel R. Buss and Jay P. Fillmore. Spherical averages and applications to spherical splines and interpolation. *ACM TOG*, 20(2):95–126, 2001. 4

[3] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. NuScenes: A multimodal dataset for autonomous driving. In *CVPR*, pages 11621–11631, 2020. 2

[4] Li Chen, Penghao Wu, Kashyap Chitta, Bernhard Jaeger, Andreas Geiger, and Hongyang Li. End-to-end autonomous driving: Challenges and frontiers. *IEEE TPAMI*, pages 1–20, 2024. 1, 2

[5] Yurui Chen, Chun Gu, Junzhe Jiang, Xiatian Zhu, and Li Zhang. Periodic vibration gaussian: Dynamic urban scene reconstruction and real-time rendering. *arXiv:2311.18561*, 2023. 3, 6, 8

[6] Kai Cheng, Xiaoxiao Long, Kaizhi Yang, Yao Yao, Wei Yin, Yuexin Ma, Wenping Wang, and Xuejin Chen. GaussianPro: 3D gaussian splatting with progressive propagation. *arXiv:2402.14650*, 2024. 3, 6

[7] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, pages 3213–3223, 2016. 2

[8] Angela Dai, Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Christian Theobalt. BundleFusion: Real-time globally consistent 3D reconstruction using on-the-fly surface reintegration. *ACM TOG*, 36(3):24:1–24:18, 2017. 3, 5

[9] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, pages 1–16, 2017. 2

[10] Jin Fang, Dingfu Zhou, Jingjing Zhao, Chenming Wu, Chulin Tang, Cheng-Zhong Xu, and Liangjun Zhang. Lidar-cs dataset: Lidar point cloud dataset with cross-sensors for 3d object detection. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 14822–14829. IEEE, 2024. 2

[11] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, pages 3354–3361, 2012. 2

[12] Giorgio Grisetti, Rainer Kummerle, Cyrill Stachniss, and Wolfram Burgard. A tutorial on graph-based SLAM. *IEEE Intelligent Transportation Systems Magazine*, 2(4):31–43, 2010. 4

[13] Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, et al. Planning-oriented autonomous driving. In *CVPR*, pages 17853–17862, 2023. 2

[14] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2D gaussian splatting for geometrically accurate radiance fields. In *ACM SIGGRAPH Conference Papers*, pages 32:1–32:11, 2024. 3, 6

[15] Shengyu Huang, Zan Gojcic, Mikhail Usvyatsov, Andreas Wieser, and Konrad Schindler. Predator: Registration of 3D point clouds with low overlap. In *CVPR*, pages 4267–4276, 2021. 4

[16] Xin Jing, Xiaqing Ding, Rong Xiong, Huanjun Deng, and Yue Wang. DXQ-Net: Differentiable LiDAR-Camera extrinsic calibration using quality-aware flow. In *IROS*, pages 6235–6241, 2022. 3

[17] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkuehler, and George Drettakis. 3D gaussian splatting for real-time radiance field rendering. *ACM TOG*, 42(4):139:1–139:14, 2023. 1, 3, 6

[18] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. In *ICCV*, pages 3992–4003, 2023. 4, 8

[19] Hao Li, Chenming Wu, Ming Yuan, Yan Zhang, Chen Zhao, Chunyu Song, Haocheng Feng, Errui Ding, Dingwen Zhang, and Jingdong Wang. XLD: A cross-lane dataset for benchmarking novel driving view synthesis. In *Proc. of the Intl. Conf. on 3D Vision (3DV)*, 2025. 2, 6, 7

[20] Ming Li, Xin Chen, Xin Li, Bin Ma, and P.M.B. Vitanyi. The similarity metric. *IEEE Transactions on Information Theory*, 50(12):3250–3264, 2004. 3

[21] Yiming Li, Zhiheng Li, Nuo Chen, Moonjun Gong, Zonglin Lyu, Zehong Wang, Peili Jiang, and Chen Feng. Multiagent multitraversal multimodal self-driving: Open MARS dataset. In *CVPR*, pages 22041–22051, 2024. 2

[22] Yun-Jin Li, Mariia Gladkova, Yan Xia, Rui Wang, and Daniel Cremers. VXP: Voxel-cross-pixel large-scale image-lidar place recognition. *arXiv:2403.14594*, 2024. 3

[23] Bencheng Liao, Shaoyu Chen, Xinggang Wang, Tianheng Cheng, Qian Zhang, Wenyu Liu, and Chang Huang. MapTR: Structured modeling and learning for online vectorized HD map construction. In *ICLR*, 2023. 2

[24] Yiyi Liao, Jun Xie, and Andreas Geiger. Kitti-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d. *IEEE TPAMI*, 45(3):3292–3310, 2022. 2

[25] William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3D surface construction algorithm. *TOG*, 21(4):163—-169, 1987. 4

[26] Tao Lu, Mulin Yu, Linning Xu, Yuanbo Xiangli, Limin Wang, Dahua Lin, and Bo Dai. Scaffold-GS: Structured 3D gaussians for view-adaptive rendering. In *CVPR*, pages 20654–20664, 2024. 3, 6, 8

[27] 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. *Commun. ACM*, 65(1):99—106, 2021. 1, 3

[28] Alexander Millane, Zachary Taylor, Helen Oleynikova, Juan Nieto, Roland Siegwart, and César Cadena. C-blox: A scalable and consistent TSDF-based dense mapping approach. In *IROS*, 2018. 4

[29] Pierre Moulon, Pascal Monasse, Romuald Perrot, and Renaud Marlet. OpenMVG: Open multiple view geometry. In *International Workshop on Reproducible Research in Pattern Recognition*, pages 60–74, 2016. 3

[30] Geoffrey Pascoe, William P Maddern, and Paul Newman. Robust direct visual localisation using normalised information distance. In *BMVC*, pages 70:1–70:13, 2015. 2, 3, 5

[31] Johannes L. Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, pages 4104–4113, 2016. 3, 4, 5

[32] Aleksandr Segal, Dirk Haehnel, and Sebastian Thrun. Generalized-ICP. In *Robotics: Science and Systems*, pages 21:1–21:8, 2009. 4

[33] Tixiao Shan, Brendan Englot, Drew Meyers, Wei Wang, Carlo Ratti, and Daniela Rus. LIO-SAM: Tightly-coupled lidar inertial odometry via smoothing and mapping. In *IROS*, pages 5135–5142, 2020. 4

[34] Noah Snavely, Steven M. Seitz, and Richard Szeliski. Photo tourism: exploring photo collections in 3d. *ACM TOG*, 25 (3):835—-846, 2006. 3

[35] Haryong Song, Wonsub Choi, and Haedong Kim. Robust vision-based relative-localization approach using an rgb-depth camera and lidar sensor fusion. *IEEE Trans. Industrial Electronics*, 63(6):3725–3736, 2016. 2

[36] Frank Steinbrücker, Jürgen Sturm, and Daniel Cremers. Real-time visual odometry from dense RGB-D images. In *ICCVW*, pages 719–722, 2011. 5

[37] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *CVPR*, pages 2446–2454, 2020. 2

[38] Matthew Tancik, Vincent Casser, Xinchen Yan, Sabeek Pradhan, Ben Mildenhall, Pratul P Srinivasan, Jonathan T Barron, and Henrik Kretzschmar. Block-nerf: Scalable large scene neural view synthesis. In *CVPR*, pages 8248–8258, 2022. 3

[39] Ignacio Vizzo, Tiziano Guadagnino, Jens Behley, and Cyrill Stachniss. VDBFusion: Flexible and efficient TSDF integration of range sensor data. *Sensors*, 22(3), 2022. 4

[40] Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. Superglue: A stickier benchmark for general-purpose language understanding systems. In *NeurIPS*, 2019. 4, 5

[41] T. Whelan, S. Leutenegger, R. F. Salas-Moreno, B. Glocker, and A. J. Davison. ElasticFusion: Dense SLAM without a pose graph. In *Robotics: Science and Systems*, pages 1:1–1:9, 2015. 3

[42] Zirui Wu, Tianyu Liu, Liyi Luo, Zhide Zhong, Jianteng Chen, Hongmin Xiao, Chao Hou, Haozhe Lou, Yuantao Chen, Runyi Yang, Yuxin Huang, Xiaoyu Ye, Zike Yan, Yongliang Shi, Yiyi Liao, and Hao Zhao. Mars: An instance-aware, modular and realistic simulator for autonomous driving. In *CICAI*, 2023. 3

[43] Qingshan Xu and Wenbing Tao. Planar prior assisted patchmatch multi-view stereo. *AAAI*, 2020. 3

[44] Yunzhi Yan, Haotong Lin, Chenxu Zhou, Weijie Wang, Haiyang Sun, Kun Zhan, Xianpeng Lang, Xiaowei Zhou, and Sida Peng. Street gaussians for modeling dynamic urban scenes. In *ECCV*, 2024. 1, 3, 6

[45] Jiawei Yang, Boris Ivanovic, Or Litany, Xinshuo Weng, Seung Wook Kim, Boyi Li, Tong Che, Danfei Xu, Sanja Fidler, Marco Pavone, and Yue Wang. EmerNeRF: Emergent spatial-temporal scene decomposition via self-supervision. In *ICLR*, 2024. 6, 8

[46] Mengjiao Yang, Yilun Du, Kamyar Ghasemipour, Jonathan Tompson, Dale Schuurmans, and Pieter Abbeel. Learning interactive real-world simulators. *arXiv:2310.06114*, 2023. 3

[47] Sheng Yang, Xiaoling Zhu, Xing Nian, Lu Feng, Xiaozhi Qu, and Teng Ma. A robust pose graph approach for city scale LiDAR mapping. In *IROS*, pages 1175–1182, 2018. 3, 4

[48] Xuemeng Yang, Licheng Wen, Yukai Ma, Jianbiao Mei, Xin Li, Tiantian Wei, Wenjie Lei, Daocheng Fu, Pinlong Cai, Min Dou, Botian Shi, Liang He, Yong Liu, and Yu Qiao. DriveArena: A closed-loop generative simulation platform for autonomous driving. *arXiv:2408.00415*, 2024. 1

[49] Bo Zhang, Xinyu Cai, Jiakang Yuan, Donglin Yang, Jianfei Guo, Xiangchao Yan, Renqiu Xia, Botian Shi, Min Dou, Tao Chen, Si Liu, Junchi Yan, and Yu Qiao. Resimad: Zero-shot 3d domain transfer for autonomous driving with source reconstruction and target simulation. In *ICLR*, 2024. 2

[50] Ji Zhang, Sanjiv Singh, et al. LOAM: Lidar odometry and mapping in real-time. In *Robotics: Science and Systems*, pages 7:1–7:9, 2014. 4

[51] Xiangrui Zhao, Sheng Yang, Tianxin Huang, Jun Chen, Teng Ma, Mingyang Li, and Yong Liu. SuperLine3D: Self-supervised line segmentation and description for lidar point cloud. In *ECCV*, pages 263–279, 2022. 4

[52] Qian-Yi Zhou and Vladlen Koltun. Color map optimization for 3D reconstruction with consumer depth cameras. *ACM TOG*, 33(4):155:1–155:10, 2014. 3, 5

[53] Jannik Zürn, Paul Gladkov, Sofía Dudas, Fergal Cotter, Sofi Toteva, Jamie Shotton, Vasiliki Simaiaki, and Nikhil Mohan. WayveScenes101: A dataset and benchmark for novel view synthesis in autonomous driving. *arXiv:2407.08280*, 2024. 2