

Crowd-Sourced NeRF: Collecting Data from Production Vehicles for 3D Street View Reconstruction

Tong Qin, Changze Li, Haoyang Ye, Shaowei Wan, Minzhen Li, Hongwei Liu, and Ming Yang

arXiv:2406.16289v1 [cs.CV] 24 Jun 2024

Abstract—Recently, Neural Radiance Fields (NeRF) achieved impressive results in novel view synthesis. Block-NeRF showed the capability of leveraging NeRF to build large city-scale models. For large-scale modeling, a mass of image data is necessary. Collecting images from specially designed data-collection vehicles can not support large-scale applications. How to acquire massive high-quality data remains an opening problem. Noting that the automotive industry has a huge amount of image data, crowd-sourcing is a convenient way for large-scale data collection. In this paper, we present a crowd-sourced framework, which utilizes substantial data captured by production vehicles to reconstruct the scene with the NeRF model. This approach solves the key problem of large-scale reconstruction, that is where the data comes from and how to use them. Firstly, the crowd-sourced massive data is filtered to remove redundancy and keep a balanced distribution in terms of time and space. Then a structure-from-motion module is performed to refine camera poses. Finally, images, as well as poses, are used to train the NeRF model in a certain block. We highlight that we presents a comprehensive framework that integrates multiple modules, including data selection, sparse 3D reconstruction, sequence appearance embedding, depth supervision of ground surface, and occlusion completion. The complete system is capable of effectively processing and reconstructing high-quality 3D scenes from crowd-sourced data. Extensive quantitative and qualitative experiments were conducted to validate the performance of our system. Moreover, we proposed an application, named first-view navigation, which leveraged the NeRF model to generate 3D street view and guide the driver with a synthesized video.

Index Terms—Crowd-sourced system, intelligent vehicles, scene reconstruction, navigation, NeRF.

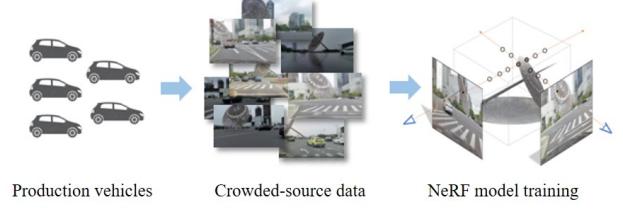
I. INTRODUCTION

SCENE reconstruction has been a long-standing topic over the last decades. Classical Structure-from-Motion(SfM)-based method, such as COLMAP [1, 2], reconstructed a 3D model from a set of 2D images taken from different angles. The basic idea was to find the 3D coordinates of points in the scene by triangulating corresponding points in multiple images. The resulting 3D points can then be used to estimate the camera poses and reconstruct the 3D geometry of the scene. Besides the sparse 3D points, the dense surface could be stuck piece by piece. However, the canonical SfM approaches

This work is supported in part by the National Natural Science Foundation of China under Grants U22A20100 and 62173228 (Corresponding author: Ming Yang; e-mail: MingYANG@sjtu.edu.cn).

Tong Qin, Changze Li, and Ming Yang are with the Global Institute of Future Technology, Shanghai Jiao Tong University, Shanghai, China.

Haoyang Ye, Shaowei Wan, Minzhen Li, and Hongwei Liu are with IAS BU, Huawei Technologies, Shanghai, China.



(a) The basic idea of crowd-sourced NeRF



(b) Synthetic view with navigation information.

Fig. 1. (a) shows the basic idea of crowd-sourced NeRF, which is collecting data from production vehicles to train the NeRF model for large-scale reconstruction. (b) shows an application, first-view navigation. A reference line (in yellow) is rendered with the realistic scene, which provides the driver with clearer experience. The video can be found at: <https://youtu.be/oVUC634R1zw>.

focused on 3D structures, ignoring photo-realistic textures. Recently, a learning-based approach, named NeRF [3], represented the scene in the neural model implicitly. NeRF assumed that the scene can be represented as a continuous function that maps a 3D point to a color and opacity value. The model was trained using a set of images and corresponding camera poses, and the continuous function can be queried to render novel views of the scene from any desired viewpoint. NeRF can achieve high-fidelity and realistic image synthesis, which can be widely applied to Virtual Reality(VR), Augmented Reality(AR), and simulation in autonomous driving industries.

While NeRF is an impressive technology, it requires a large amount of high-quality data to generate accurate and realistic 3D models. Earlier works tended to focus on object-centric reconstruction within a small size, e.g. a table and a room. Block-NeRF [4] proposed a partitioning strategy that separated the city into multiple small blocks so that NeRF could be

extended into city-scale modeling. Block-NeRF is indeed a groundbreaking approach in leveraging Neural Radiance Fields for large-scale city modeling. However, it inherently relies on data collected via specialized vehicles equipped with high-resolution cameras and precision localization hardware, which is not scalable for extensive urban environments due to high costs and logistical complexities. How to acquire massive and high-quality data to support NeRF training remains an open question for large-scale reconstruction.

Crowded sourcing is an efficient way to collect massive data in a short time and at a relatively low cost. Nowadays, thousands of vehicles equipped with multiple cameras run everywhere and every day, which forms an ideal platform for crowd-sourced data collection. In addition, the data stream of crowd-sourcing is stable and constant, which can support the daily updating of the model. Therefore, Crowd-sourced data is cost-effective and has good real-time performance, which is crucial for large-scale urban 3D reconstruction and updates. However, how to use crowd-sourced data efficiently is a challenging issue. On the one hand, with the continued growth of the crowd-sourced database, there is a lot of redundant data inside, which brings a huge burden to the storage and computation resources. On the other hand, due to lacking a high-accurate positioning system, the pose of the image is inaccurate, which brings great difficulties to fusing them together.

To address the above-mentioned challenges, we propose CS-NeRF (Crowd-Sourced NeRF), which uses data captured by countless production vehicles to reconstruct the scene with the NeRF model. Firstly, the massive crowd-sourced data is filtered by a spatial-temporal selector, which removes the redundancy and keeps a balanced spatial and temporal distribution. The image is further segmented into multiple classes. The dynamic scene is masked to improve accuracy in the SfM and NeRF training procedure. We use inverse projection to estimate the depth of ground pixels, which serves as the depth for the supervision. Due to lacking a high-accurate localization device on vehicles, the pose of data is inaccurate and noisy. We perform SfM to precisely localize each image. The NeRF model is trained with depth supervision, occlusion completion, as well as sequence appearance embedding. Last but not least, we apply the NeRF model in a real application, first-view navigation, which synthesizes the scene with navigation information together for clear guidance. The contributions of this paper are summarized as follows:

- We propose a comprehensive framework for large-scale NeRF reconstruction, that integrates multiple modules, effectively processing and reconstructing high-quality 3D scenes from crowd-sourced data.
- We propose three improvements within our CS-NeRF framework: sequence appearance embedding, ground surface supervision, and occlusion completion.
- Real-world experiments with massive crowd-sourced data are conducted to validate the performance of the proposed system.

The crowd-sourced data used in the experiment is open-

sourced for the benefit of the community.^{1 2}

II. RELATED WORK

A. Implicit Neural Model

Traditional geometry-based 3D scene representations usually represent the scene as explicit models, such as point clouds [5], textured meshes [6] or voxels [7]. The sparsity and discretization attributes of these models make it difficult to render high-quality novel views. NeRF [3] and its subsequent works [8, 9] parameterized the positions and directions by multilayer perceptron (MLP) and composited the novel views using volumetric rendering. This structure ensured the model in a smooth and continuous representation, which helped to generate more photo-realistic and higher quality results [10].

Combining image segmentation supervision, Semantic-NeRF [11] jointly encoded density and radiance along with scene semantics. Semantic information in images can be further used to mask dynamic objects for static scene modeling, as mentioned in [4, 12]. Panoptic-NeRF [13] further involved segmentations to decouple background and foreground as different networks to produce implicit panoptic models. PanopticLifting [14] lifts 2D panoptic labels to an implicit 3D volumetric representation, which is robust to label noise and has the potential to enhance semantic temporal stability.

Accurate surface reconstruction was difficult for NeRF models to achieve since NeRF mainly focused on learning view synthesis results rather than scene geometry. Implicit signed distance function (SDF) methods [15, 16] are proposed to better disentangle the geometry and radiance, and ensure high-quality surface geometry. In these approaches, volume density functions in the original NeRF were replaced by transformed learnable SDF functions. However, for texture-less scenes, depth supervision was necessary to ensure good surface geometry. For example, DS-NeRF [17] and URF [12] supervised volume densities to fit the true depth distributions from sparse point clouds. On the other hand, these models could provide denser coverage than raw point clouds, thanks to the image observations utilized during training. DS-NeRF uses sparse point clouds obtained from SfM for depth supervision, but this method is unable to triangulate a large number of reliable 3D points in texture-less structures such as the ground surface, which results in the geometry not being able to be improved effectively. URF uses LiDAR for depth supervision, which may not be available in customer vehicles. In contrast, Our proposed method uses inverse mapping to avoid the problems of the above methods.

Instead of modeling synthetic or small indoor scenes, Block-NeRF [4], Urban Radiance Fields (URF) [12] targeted to real-world outdoor mapping applications. These approaches took street-view images in individual blocks, and depths if available, as inputs to generate large-scale NeRF models. Similarly, Mega-NeRF [18] and BungeeNeRF [19] made use of aerial images or multi-scale satellite data. Although these

¹https://drive.google.com/drive/folders/1AGxYzPJceb4xEs3CeEwC5h4J_cVq_AX?usp=sharing

²<https://pan.baidu.com/s/1fECS4ue8HKUclq3Zqi3UiQ?pwd=gzs5>

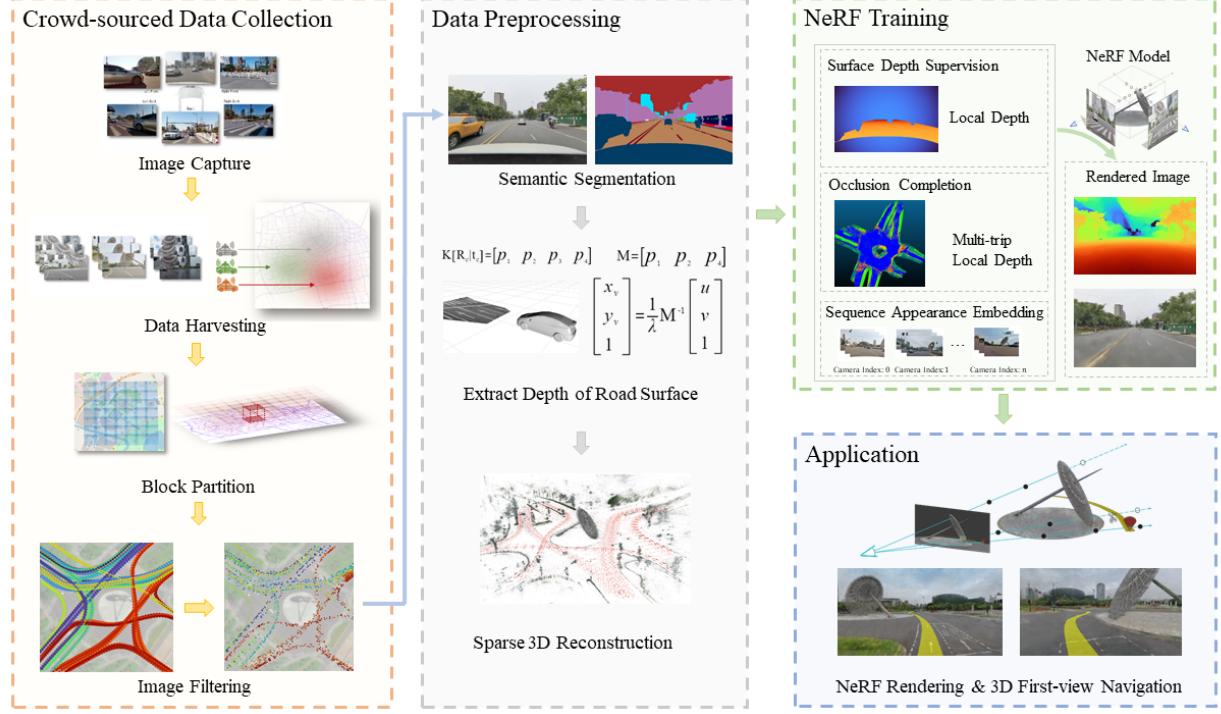


Fig. 2. The structure of the proposed crowd-sourcing system. The strategy of crowd-sourced data collection is elaborated in Sec. IV, which collects massive data and filters them with a balanced spatial and temporal distribution. Then, the data pre-process model, Sec. V, segments images semantically, extracts the depth of the ground surface, and refines the camera pose by SfM. The NeRF training procedure is illustrated in Sec. VI, which trains the NeRF model with three improvements, which are sequence appearance embedding, surface depth supervision, and occlusion completion.

methods can generate urban-level NeRF models, their data are collected using professional high-precision acquisition equipment, which reduces the complexity of the problem. In contrast, the data collected in our method comes from crowd-sourced data, which is of low quality. In order to get good reconstruction results, it is necessary to perform data cleaning and pose optimization, as well as add some constraints for crowd-sourced scenarios, such as scene appearance embedding. By adding additional transient and appearance embeddings, NeRF-W [20] addressed the appearance mismatches across different images. To deal with unbounded environments, different scene parameterizations [9, 18, 21] were also used in these methods.

B. Camera Pose Optimization

Accurate camera poses are essential for NeRF models to converge and obtain consistent color and occupancy. Classical Structure-from-Motion (SfM) [1, 5, 22] was an effective offline way to guarantee the accuracy of camera poses, as well as the sparse scene structure. As proposed in [4, 23]–[26], jointly optimizing camera poses and NeRF models during training can further improve the model consistency, meanwhile reducing the requirements for very accurate camera poses. NoPe-NeRF [27] has utilized monocular depth estimation to generate dense point clouds and then used the constraints of the point clouds to optimize both the neural radiance field and the camera pose. However, due to the accuracy of monocular depth estimation and the problem of local optima, it is difficult to achieve good results in large-scale autonomous driving scenarios.

In our method, pose optimization is considered to compensate for calibration errors and noisy camera poses from customer vehicles.

C. Crowd-sourced Mapping

Data collection by laboratory sensor settings, such as the one used in Block-NeRF [4], were usually not affordable for large-scale applications. For instance, high-resolution camera with 2K resolution and high-accurate positioning system with centimeter-level accuracy. There exist several works related to crowd-sourced mapping for geometric models. Crowd trajectory data was utilized in [28] for automated intersection mapping. In [29, 30], a crowd-sourced process was used for the new HD map feature layer generation. For NeRF applications, NeRF-W [20] was one of the earliest works related to crowdsourcing. It took internet photo collections as inputs and succeeded to produce a static model without transient elements. To produce driving-view models, our approach intended to use data from production vehicles. We will introduce the crowd-sourced NeRF system dealing with low-fidelity and noisy data in the following sections.

III. FRAMEWORK OVERVIEW

An overview of our method is shown in Fig. 2. The strategy of crowd-sourced data collection is elaborated in Sec. IV, which collects massive data and filters them with a balanced spatial and temporal distribution. Then, the data pre-process model, Sec. V, segments images semantically, extracts the

depth of the ground surface, and refines the camera pose by SfM. The NeRF training procedure is illustrated in Sec. VI, which trains the NeRF model with three improvements, which are sequence appearance embedding, surface depth supervision, and occlusion completion.

IV. CROWD-SOURCED DATA COLLECTION

A. Data Collection Platform

The data is collected in a crowd-sourcing way by multiple mass-produced vehicles. Only an integrated positioning system and cameras with intrinsic and extrinsic calibrations are required. A large number of production vehicles, equipped with ADAS (Advanced Driver-Assistance Systems) or ADS (Automated Driving Systems), meet the hardware requirements and can be served as the data collection source. Images, along with meter-level global poses, are uploaded to the data collection platform.

B. Data Selection

Since substantial data aggregates continuously, the data selection pipeline is designed to efficiently select images and reduce redundancy. The data selection pipeline ensures the spatial and temporal distribution of the data by block partitioning and image filtering.

1) *Block Partition*: Following Block-NeRF [4], the scene is divided into small blocks. The crowd-sourced data is assigned to blocks according to the pose. The adjacent blocks are overlapped by 20% to ensure consistency. Subsequently, a separate neural network can be trained for each block, enabling parallel processing and reducing overall training time.

2) *Image Filtering*: In each block, images are filtered automatically according to following principles:

- Small proportion of moving objects: We identify moving objects through semantic segmentation. Images with a proportion of moving objects greater than 40 percent are filtered out.
- Stable pose estimation: Theoretically, the prior pose should not deviate too far after optimization. Therefore, if the pose of a specific image changes a lot after the pose optimization, the image is considered untrustworthy and is filtered out.
- Data diversity: Images captured with similar times, positions, and viewpoints can be clustered by utilizing optimized poses and timestamps. Only few images are retained in each clustering category.

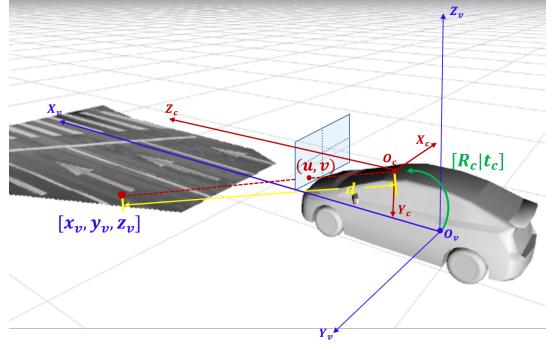
V. DATA PREPROCESSING

A. Semantic Segmentation

Semantic segmentation is a mature technology, which has been widely used in visual applications. Typical CNN-based methods include FCN [31], U-Net [32], SegNet [33], BiSeNet V2 [34], etc. In this project, we adopted BiSeNet V2 [34], which had a good trade-off between speed and accuracy. In BiSeNet V2, the network was separated into two branches. The detail branch captured low-level details and generated



(a) The example of semantic segmentation result.



(b) The illustration of ground surface projection.

Fig. 3. In (a), the image is segmented into multiple semantic groups, such as lane, crosswalk, vehicle, tree, road, stop lines, etc. (b) is the diagram of the inverse projection process. The pixel is inversely projected to the ground ($z_v = 0$), so that the depth d of the ray can be obtained.

high-resolution features, while the semantic branch with deep layers, obtained high-level semantic context.

Semantic segmentation was performed on each image. As shown in Fig. 3(a), the image was segmented into multiple semantic groups, such as lane, crosswalk, vehicle, tree, road, stop lines, etc. There are two usages for semantic segmentation results:

- Mask moving object. The moving objects, such as cars and pedestrians, severely impact the accuracy of the following 3D reconstruction and NeRF model training.
- Road surface extraction. Through segmentation, the road surface can be detected on the image plane, which will be used for surface depth generation in the next section.

B. Depth of Ground Surface

Since the data is captured on the open street, we assume that the road surface is a roughly flat plane. Therefore, the road surface can provide the depth for supervision in the NeRF training process. The depth of the road surface can be inferred by inverse projection. A diagram of the inverse projection process is shown in Fig. 3(b). The pixel is inversely projected to the ground ($z_v = 0$), so that the depth d of the ray can be obtained.

Specifically, the formula for projecting a 3D point under the vehicle coordinates $[x_v, y_v, z_v]$ onto the 2D image plane $[u,$

$v]$ is:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \lambda \mathbf{K} [\mathbf{R}_c | \mathbf{t}_c] \begin{bmatrix} x_v \\ y_v \\ z_v \\ 1 \end{bmatrix}, \quad (1)$$

where \mathbf{R}_c , \mathbf{t}_c is the extrinsic calibration of the camera with respect to the vehicle's center, $\mathbf{R}_c \in SO(3)$. \mathbf{K} is the intrinsic parameter of the camera, $\mathbf{K} = \begin{bmatrix} f_x, 0, c_x \\ 0, f_y, c_y \\ 0, 0, 1 \end{bmatrix}$, where f_x, f_y are focal lengths, and c_x, c_y represent the camera principal point. This equation projects the 3D point to 2D pixel $[u, v]$ on the image plane. λ is a scaling factor. We define the origin of the vehicle coordinates as located on the ground, so z_v is 0 for points on the ground. x_v and y_v can be solved given \mathbf{K} , \mathbf{R}_c , \mathbf{t}_c , $[u, v]$ and $z_v = 0$. Therefore, we can get the depth d of the ray, which is the distance between the 3D surface point $[x_v, y_v, 0]$ and the camera's center. The depth will be used for depth supervision in Sec. VI-C.

C. Sparse 3D Reconstruction

The crowd-sourced data is in meter-level positioning accuracy, which is insufficient for NeRF training. To address this issue, we perform Structure-from-motion (SfM) to further improve localization accuracy. We adopt COLMAP [1], which refines the geometric structure of the scene by feature extraction, feature matching, and bundle adjustment (optimizing camera poses and feature positions). Some improvements are made to improve accuracy in the feature matching step:

- Since invalid features from dynamic objects (car, pedestrian, bicycle, etc.) affect the accuracy of the 3D scene's reconstruction, we remove features on dynamic objects based on the segmentation.
- We use semantic labels to further filter out mismatched pairs. Only pairs with the same semantic label are kept. For example, features belonging to the road, or the building are treated as inliers.

In addition, some improvements are made to improve the effectiveness of the SfM pose optimization process:

- Since we have the meter-level position for each image, the prior pose can guide feature matching instead of exhaustively searching. We perform feature matching only with candidate images within the neighborhood.
- In the bundle adjustment, the prior poses are adopted to guarantee the correctness of optimization and accelerate convergence.

Therefore, we have a centimeter-level position for crowd-sourced images, which can be used for further NeRF training.

VI. NERF TRAINING

A. NeRF Preliminaries

Neural Radiance Field (NeRF) represents the scene as a continuous function, whose input is a 5D vector, including a 3D position $\mathbf{x} = (x, y, z)$ and a view direction $\mathbf{d} = (\theta, \phi)$, and whose output is an emitted color $\mathbf{c} = (r, g, b)$ and volume

density σ . In order to improve the high-frequency detail of the rendered images, NeRF incorporates positional encoding:

$$\gamma(\mathbf{x}) = [\sin(\mathbf{x}), \cos(\mathbf{x}), \dots, \sin(2^{L-1}\mathbf{x}), \cos(2^{L-1}\mathbf{x})]^T \quad (2)$$

where L is a hyperparameter. The function can be described as an MLP network $\mathbf{F}_{\Theta} : (\gamma(\mathbf{x}), \mathbf{d}) \rightarrow (\mathbf{c}, \sigma)$ [3]. It is worth noting that \mathbf{x} in the NeRF network is inputted at the first layer, while the \mathbf{d} is injected into the network closer to the end of the MLP to alleviate the "shape-radiance ambiguity" [21]. Colors can be obtained by volume rendering of any ray passing through the scene, the expected color $C(\mathbf{r})$ of the ray $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$ within near and far bounds $[t_n, t_f]$ is:

$$C(\mathbf{r}) = \int_{t_n}^{t_f} w(t) \mathbf{c}(\mathbf{r}(t), \mathbf{d}) dt, \quad (3)$$

where $w(t) = \exp\left(-\int_{t_n}^t \sigma(\mathbf{r}(s)) ds\right) \cdot \sigma(\mathbf{r}(t))$. The training process is supervised by an $L2$ photometric loss:

$$L_{rgb} = \|C_i(\mathbf{r}) - C_i^{gt}(\mathbf{r})\|_2^2, \quad (4)$$

where $C_i^{gt}(\mathbf{r})$ is the ground truth color of a pixel i in the image.

We build on the method, Nerfacto [35], which is an open-sourced toolbox for NeRF development, integrating a lot of new features. Nerfacto integrated pose refinement from NeRF- [23], proposal network sampler and scene contraction from Mip-NeRF 360 [9], appearance embedding from NeRF-W [20], hash coding and fused MLP from Instant-NGP [36]. In the following sections, we only elaborate on our modifications and improvements based on the Nerfacto.

B. Sequential Appearance Embedding

Crowd-sourced data is captured by the different vehicles at different times. The inconsistency of image styles may lead to poor reconstruction performance. Traditional algorithms, such as NeRF-W [20], assigned every image with an appearance embedding vector. Although the appearance embedding solved the ambiguity of reconstruction, the high degree of freedom causes the model to learn some randomness in a single image, such as shadows. Instead of assigning appearance embedding vectors for each image, we assign appearance embedding vectors for each sequence. In other words, the same sequence (images captured by the same camera on the same trip) shares the same embedding vector instead of using one appearance embedding vector per image. Therefore, we reduce the dimension of appearance embedding and reduce the randomness. The model learns the common style for a series of images, instead of learning a special pattern for a certain image.

C. Depth Supervision of Ground Surface

In the canonical NeRF training procedure, the color of pixels is supervised while the geometry of the scene is neglected. Due to the lack of geometric supervision, there are often many floating artifacts in the novel view [12, 17]. Depth supervision contributes to improving NeRF's geometric quality when the

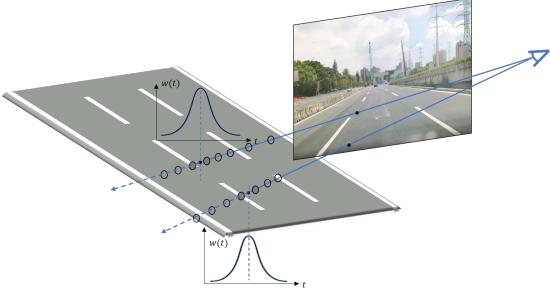


Fig. 4. The illustration of depth supervision. The density distribution of sample points along a ray is supervised by the Dirac function.

number of input views is limited. As illustrated in Sec. V-B, the depth of the ray, which hits on the ground surface, is generated. This depth is used for supervision to improve the geometric quality. We follow the depth supervision method proposed in Urban Radiance Fields (URF) [12], by supervising the density distribution of a ray looking like the Dirac function, as shown in Fig. 4. The depth loss function is defined as:

$$L_d = \int_{t_n}^{t_f} (w(t) - \delta(z))^2 dt, \quad (5)$$

where $\delta(z)$ is the Dirac function. The final loss is the sum of color loss for all rays and depth loss for rays s , which hits the road surface:

$$L = \sum_i L_{rgb} + \sum_{s \in S} L_d. \quad (6)$$

After adding the depth loss, the rendered depth is smoother than the one without depth supervision, as shown in Fig. 8.

D. Occlusion Completion

Although semantic segmentation is used to mask dynamic objects (e.g. cats, pedestrians, bicyclists), masked areas may cause black holes due to a lack of efficient supervision. As shown in Fig. 10, black shadows appear in the area shaded by vehicles in the rendered view. To this end, we fill depth of the ground in the mask area with the nearby ground. Therefore, the masked area is supervised effectively, and the quality of ground reconstruction is improved, as shown in Fig. 10.

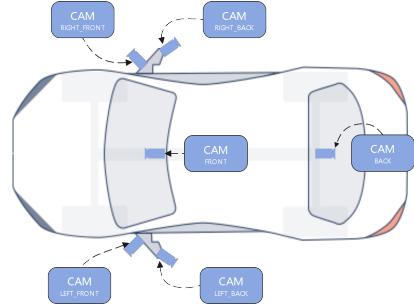
Overall, the comparison of proposed method against typical NeRF methods, such as Mip-NeRF [8], Instant-NGP [36], Nerfacto [35], Block-NeRF [4], is shown in Table I.

VII. EXPERIMENTS

In this section, we performed real-world experiments to evaluate the proposed system. The data was captured in a crowd-sourcing way by multiple production vehicles. We focused on the qualitative and quantitative results compared against other state-of-the-art algorithms.



(a) Commercial vehicle: ArcFox Alpha S



(b) Original sensor configuration

Fig. 5. (a) shows the vehicle we used for crowd-sourced data collection. (b) shows the sensor setup we used for experiments. (The vehicle contains more sensors than we used.)

A. Experimental Data Collection

1) Hardware Configuration: The dataset was collected on public urban roads by production vehicles, which were not specifically designed for data collection. By cooperating with the automobile company, *ArcFox Alpha S*, were used for data collection. Our collaboration with the company operating these vehicles allowed us access to raw data directly from the fleet, encompassing a wide variety of users, locations, and environmental conditions. The vehicle was equipped with multiple cameras. Among these cameras, 6 cameras were used in our project: one located at the front window shield, one located at the back, and four located at the right and left side respectively, as shown in Fig. 5. Specifically, cameras operate at 20Hz frequency with 640x480 resolution and JPEG compression.

An example of captured images from one vehicle was shown in Fig. 6(a). All cameras were factory-calibrated. An integrated positioning system was incorporated with GPS, IMU, and wheel speedometer, which achieved meter-level localization accuracy. Specifically, in open area, where GPS signals are well received, the localization accuracy can reach 1 meter, while in complex region, where GPS signals are occluded and reflected, the localization error is over 10 meters. One vehicle was specially equipped with Lidar to provide accurate depth only for the evaluation purpose. The images were captured on-board and uploaded to the cloud platform. The proposed system run on the cloud service offline.

2) Century Avenue Dataset: In the following experiment, we focused on one block, the Century Avenue Intersection in Shanghai, China. Driving in this area was challenging for

TABLE I
COMPARISON FOR DIFFERENT NERF METHODS

	Scene Contraction	Hash Encoding	Appearance Embedding	Block Partition	Depth Supervision	Occlusion Completion
Mip-NeRF [8]	✓					
Instant-NGP [36]		✓				
Nerfacto [35]	✓	✓				
Block-NeRF [4]	✓		✓		✓	
CS-NeRF(ours)	✓	✓	✓		✓	✓

*Our method was built on the top of Nerfacto [35].



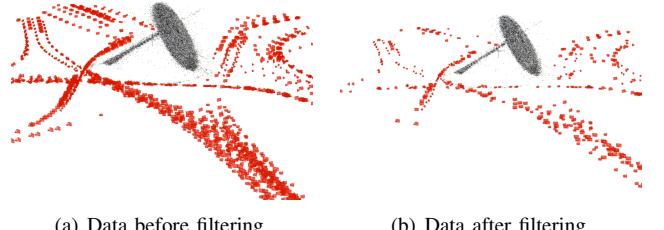
(a) Images captured by the vehicle from six cameras.



(b) Sample images from crowd-sourced dataset.

Fig. 6. (a) shows images captured by the data collection platform. (b) shows the diversity of our crowd-sourced dataset.

human drivers due to its complex road structure. The block was covered $150m \times 150m$. The crowd-sourced data contained 58 trips passing through the area from February 2022 to August 2022, including winter, spring, and summer seasons. The local time spans from 9:00 am to 7:00 pm. The total data collection time was 1.06 hours, with 29,732 images. Sample images from the crowd-sourced dataset were shown in Fig. 6(b), which illustrated the diversity of the data. As described before, data needed to be filtered to reduce redundancy and ensure the efficiency of further processing. Ultimately, 2,896 images with balanced spatial and temporal distribution were selected. An illustration of the dataset before and after filtering was shown in Fig. 7.



(a) Data before filtering.

(b) Data after filtering.

Fig. 7. The data selection procedure efficiently reduced the redundancy of the crowd-sourced data, while keeping a balanced spatial distribution.

TABLE II
METRIC COMPARISON FOR DIFFERENT NERF METHODS

	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	RMSE[m] \downarrow @ 1σ	RMSE[m] \downarrow @ 2σ
Mip-NeRF [8]	17.33	0.543	0.629	10.70	14.28
Instant-NGP [36]	16.25	0.599	0.496	15.79	25.19
Nerfacto [35]	21.33	0.748	0.193	14.13	23.84
CS-NeRF(ours)	22.54	0.754	0.185	6.11	11.34

*Comparisons of different NeRF method in the century avenue dataset.

*Our method was built on the top of Nerfacto [35].

B. Reconstruction Comparison

We reported PSNR, SSIM [37], and LPIPS [38] metrics for novel view rendering. These three metrics evaluate the rendering quality of NeRF from different perspectives, which include image reconstruction errors and perceptual differences. The calculation method can be found in [37] and [38]. Not only evaluating appearance, but we also used RMSE (Root Mean Square Error) of depth to evaluate the geometrical reconstruction quality. The ground truth of depth came from Lidar point clouds. We compared our proposed method against Mip-NeRF [8], Instant-NGP [36], and Nerfacto [35]. Nerfacto [35] is an open-source implementation, which integrated multiple advanced features, such as pose refinement from NeRF- - [23], proposal network sampler and scene contraction from Mip-NeRF 360 [9], appearance embedding from NeRF-W [20], hash coding and fused MLP from Instant-NGP [36]. Our system was built on the top of Nerfacto. Since Nerfacto optimized the appearance embedding vector for each image, there was no appearance embedding vector for test images. For a fair comparison, we used the average of embedding vectors from the same camera as the embedding vector. For each block, our training was performed on a single GPU (Nvidia Quadro P5000) with a training duration of approximately two hours.

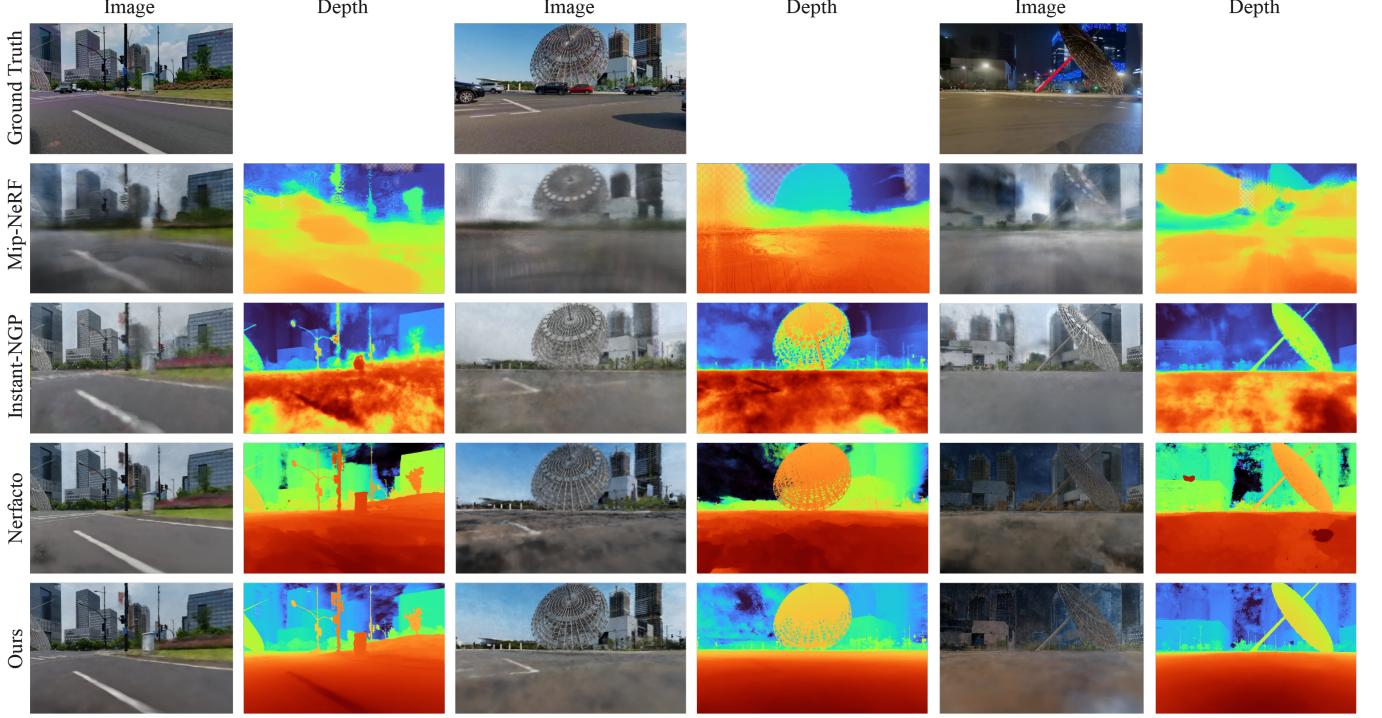


Fig. 8. Figures shows qualitative results from experiments on the crowd-sourced dataset. Compared against Mip-NeRF [8], Instant-NGP [36], Nerfacto [35], our method added three improvements, ground surface supervision, occlusion completion, and sequence appearance embedding. It can be seen that the depth of our approach is more accurate, the ground is smoother.

TABLE III
ABLATION STUDY: METRIC COMPARISON FOR DIFFERENT COMPONENTS

	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	RMSE \downarrow [m] @1 σ	RMSE \downarrow [m] @2 σ
baseline(Nerfacto [35])	21.33	0.748	0.193	14.13	23.84
baseline+S	<u>22.47</u>	<u>0.753</u>	<u>0.186</u>	13.56	22.95
baseline+D	21.54	0.750	0.191	<u>6.89</u>	12.76
baseline+D+O	21.42	0.748	0.193	<u>7.34</u>	<u>12.56</u>
baseline+S+D+O(Ours)	22.54	0.754	0.185	6.11	11.34

*Comparisons of different components. S refers to sequence appearance embedding, D refers to ground surface supervision, O refers to occlusion completion.

Multiple blocks were processed in parallel. The quantitative metric results were shown in Table. II. It could be seen that our method outperforms others in terms of appearance and depth estimation. Straightforwardly, figures of qualitative comparisons were shown in Fig. 8. More detailed examples can be found in Fig. 9 and Fig. 10. It can be seen that the rendered image from our method was more clear than others. Furthermore, the depth from our method was smoother and more accurate than others. Detailed comparisons of different components contributing to the overall performance were discussed in the next section, Ablation Study VII-C.

C. Ablation Study

Besides the engineering contribution of the crowd-sourcing framework, three theoretical novel components were proposed: sequence appearance embedding, ground surface supervision,

and occlusion completion. We performed an ablation study to investigate the individual contribution of these components and evaluated their combined effects. Nerfacto [35] was the baseline method, which different components were added on. The detailed metrics were shown in Table. III. It can be seen that the appearance performance (PSNR, SSIM, and LPIPS) improved by adding sequence appearance embedding. The depth metric (RMSE) improved a lot by adding ground surface supervision. The depth error value still seemed large, since only the nearby ground surface was optimized, there was still a big depth error in the distant scene. Occlusion completion was not evaluated separately, as it was based on ground surface supervision. It seemed that occlusion completion had no effect on the quantitative metrics, however, it contributed a lot to the rendered image, as shown in the following qualitative analysis. In the following, we gave specific examples to illustrate the effect of these components qualitatively:

1) **Sequence Appearance Embedding:** As shown in Fig. 8, lacking sequence appearance embedding, the appearance of rendered images from Mip-NeRF and Instant-NGP was far away from the ground truth. However, the rendered image from our method was similar to ground truth in terms of weather and daytime.

2) **Ground Surface Supervision:** To visualize depth reconstruction quality intuitively, we explicitly extracted the mesh and point cloud from the implicit NeRF model, as shown in Fig. 9. We used two formats to represent geometry, which were mesh and point cloud. By querying the density of every 3D space point brute-forcely and setting an occupied threshold, the dense point cloud was generated. Then, Poisson surface

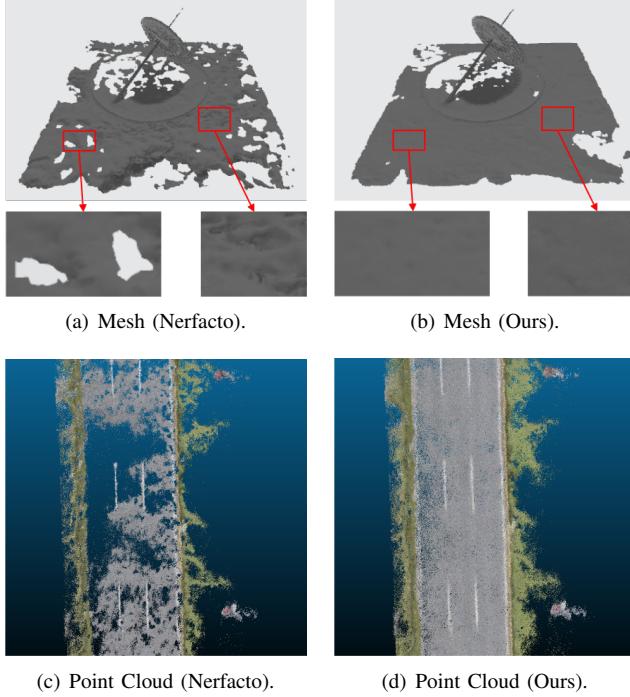


Fig. 9. Comparisons on surface reconstruction. The road surface of our method was more complete and flatter due to depth supervision and occlusion completion.

reconstruction [39] outputs high-quality meshes. It can be seen that there were a lot of holes from Nerfacto, and the surface was irregular, in Fig. 9(a) and Fig. 9(c). It was the common phenomenon for the NeRF-based method, which achieved poor geometric structure. However, the ground surface was completer and flatter with the ground depth supervision, as shown in Fig. 9(b) and Fig. 9(d).

3) Occlusion Completion: Although the metric of rendering quality seemed no improvement in Table III, the occluded area (masked by moving objects) was effectively restored in the image view. Examples were shown in Fig. 10. The mask of moving objects resulted in the black hole on the synthetic view in Nerfacto, without occlusion completion. However, the shaded area on the ground was effectively recovered by our occlusion completion. Since the masked area was interpolated from its nearby road surface in the training process, our neural network was able to predict the blocked area. Unfortunately, due to the lack of ground truth of the masked region, the metric number was not improved.

Overall, the combination of these three components, sequence appearance embedding, ground surface supervision, and occlusion completion, contributed to the overall performance.

D. Numbers of Trips Comparison

Since our system is a crowd-sourced framework, the result is affected by the amount of data directly. With the continuous increase of the crowd-sourced data, the model becomes more

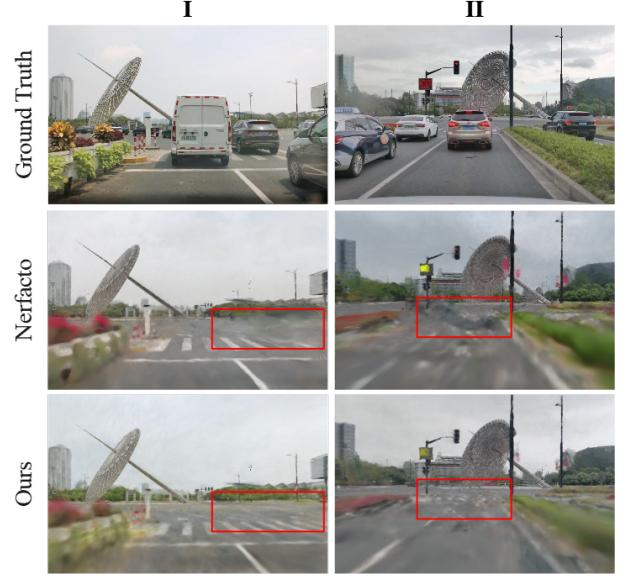


Fig. 10. Comparison on render quality near traffic flow. Due to the occlusion completion, the shaded area was predict well in our system.

TABLE IV
METRIC COMPARISON FOR DIFFERENT NUMBERS OF TRIPS

trips	training images	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	RMSE[m] @ $1\sigma \downarrow$	RMSE[m] @ $2\sigma \downarrow$
5	539	16.73	0.676	0.272	7.33	11.90
10	1104	16.89	0.686	0.262	8.18	17.38
15	1582	17.10	0.701	0.249	6.74	12.63
30	2104	17.24	0.705	0.244	6.60	11.86
45	2474	17.58	0.718	0.229	6.66	11.92
58	2845	17.79	0.718	0.226	6.13	11.54

and more complete and accurate. Objectively speaking, the proposed system had great growth potential. We conducted experiments with different numbers of trips and training images to illustrate this point. Increasing trips in one scene can improve the coverage, and reduce the ambiguities caused by unseen views. As shown in Table IV, the proposed system, CS-NeRF, obtained higher performance with more training images from different trips. We also showed qualitative results of this experiments in Fig. 11. Fewer floaters in rendered images and more precise rendered depth could be obtained, with the increase of trips.

VIII. APPLICATION: 3D NAVIGATION

By leveraging the NeRF model, we could render novel views with any virtual elements. In this section, we rendered the image with a guideline in this complex traffic intersection, which could provide users with an first-view navigation experience. As shown in Fig. 1(b), a reference line is rendered on the view to guide the driver vividly. We elaborate on how to render a novel view with guidance markers to achieve first-view navigation.

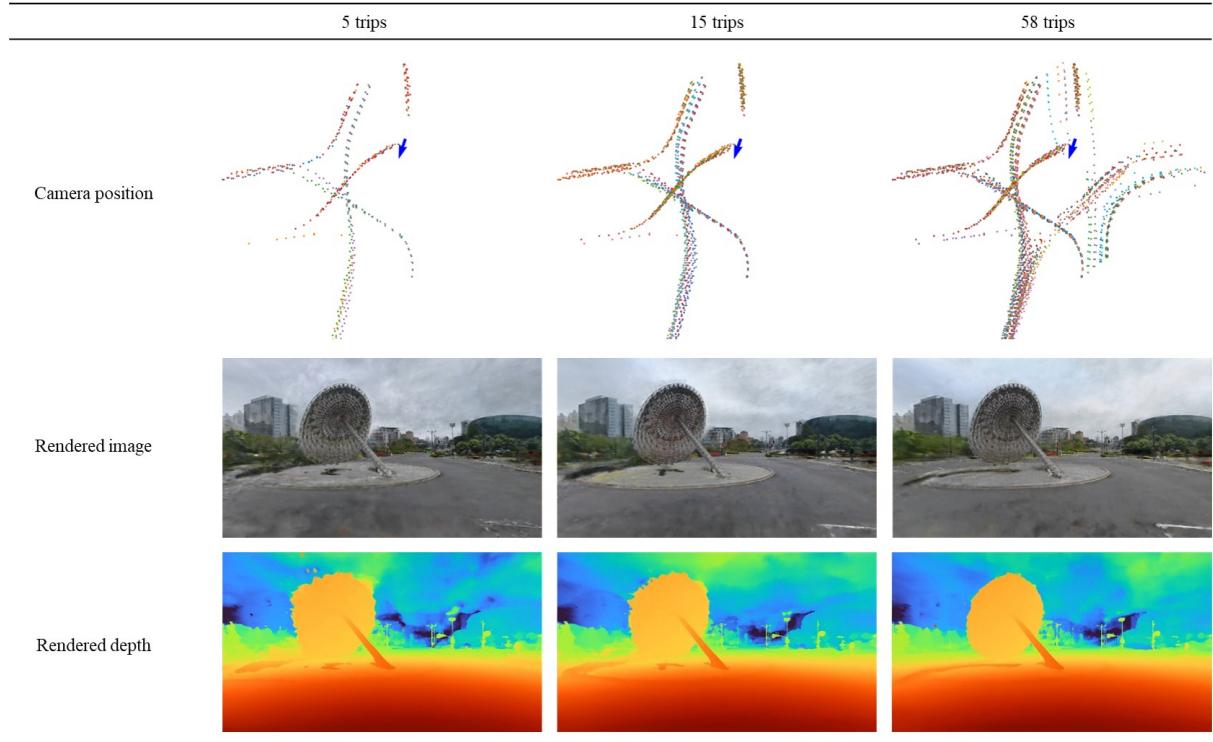


Fig. 11. Qualitative comparison on different trips. In the first row, the colorful dots stands for the view point in the X-Y plane. The blue arrow indicates the view of rendering. The rendered images and depths are shown in the second and third rows, respectively. It can be seen that the scene reconstruction becomes more complete and precise with more trips.

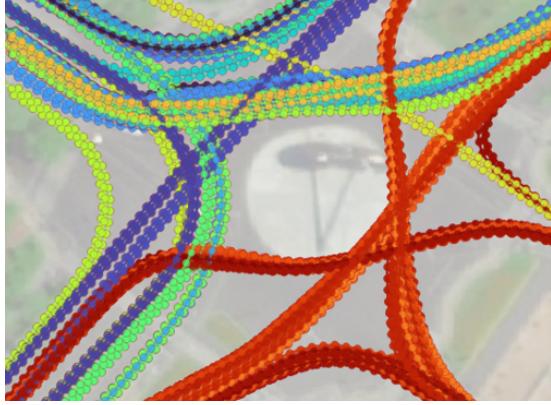


Fig. 12. The illustration of crowd-sourcing human's driving paths within one intersection. These driving paths can be used to guide followers.

A. Guide Marker Generation

In addition to images, trajectories of vehicles were collected through the crowd-sourcing platform. As shown in Fig. 12, multiple human's driving paths were collected within one intersection. The driving path from experienced drivers can be used to guide new-hand drivers. Therefore, we picked up some smooth trajectories, and treated them as guiding markers of the route in the complicated crossroad.

Algorithm 1 Novel Image Rendering with Guidance Markers

Input: Trained model \mathbb{M} , Guidance Trajectory \mathbb{P}

Output: Navigation Image \mathbb{I}

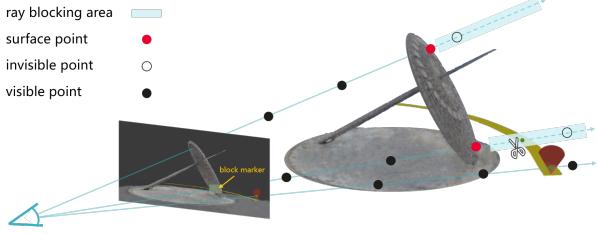
```

1: Generate the polygon area of guidance markers  $E = \{E_1, E_2, \dots, E_k\}$  from trajectory  $\mathbb{P}$ ;
2: for each ray  $r$  in  $\mathbb{I}$  do
3:   Color  $c_r$ , depth  $d_r \leftarrow$  Query  $r$  in  $\mathbb{M}$ ;
4:   for  $j = 1$  to  $k$  do
5:     if  $r$  intersects with  $E_j$  then
6:        $D \leftarrow$  Add the depth of intersection point;
7:     end if
8:   end for
9:   if  $\min(D) < d_r$  then
10:     $c \leftarrow$  Compose scene color  $c_r$  with marker color  $c_m$ ;
11:   else
12:     $c \leftarrow$  Scene color  $c_r$ ;
13:   end if
14:   Update color  $c$  to navigation image  $\mathbb{I}$ ;
15: end for

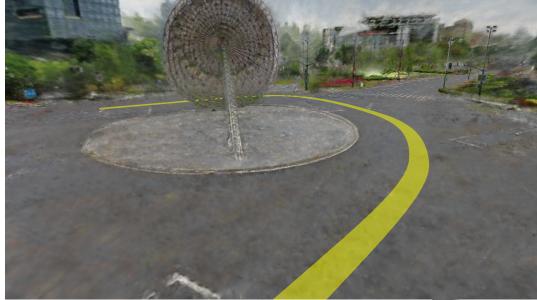
```

B. Novel View Rendering

In the following, we elaborate on how to render the novel view image with guidance markers. The whole procedures are detailed in pseudo-code in Algorithm 1. Firstly, as same with the normal rendering process, eq.(3) is performed on every ray. Secondly, we add the marker's color on this pixel if the



(a) The illustration of rendering novel view with navigation information.



(b) The Example of rendered view with occlusion.

Fig. 13. (a) The illustration of rendering novel view with navigation information. (b) The Example of rendered view with occlusion.

ray intersects with the marker. The overlaid color is,

$$C'(\mathbf{r}) = \alpha C(\mathbf{r}) + (1 - \alpha) \mathbf{c} \quad (7)$$

where \mathbf{c} is the constant color value of the marker, α is a constant scale for alpha composition. We use $\alpha = 0.3$ in the implementation to achieve a transparent appearance.

However, in some places, the guidance marker is occluded by the scene. Therefore, we need to deal with occlusion specially. As shown in Fig. 13(a), if the ray is terminated before the marker, in other words, the depth of the ray is shorter than the marker, we determine occlusion happens. The ray will keep the original color. Otherwise, the ray will be overlaid with the marker's color. An example of a novel view with an occluded marker was shown in Fig. 13(b). In this way, we can achieve an first-view navigation experience, meanwhile, the driver can change view angles online freely.

As shown in Fig. 14, we compared it with the traditional navigation tool, Amap. Amap only provided users with the 2D picture at a fixed angle view. However, our system could provide users with a realistic picture, which was similar to the real scene. In addition, the viewing angle could be changed arbitrarily, such as the front view, and top-down view, to give the user an clearer experience.

IX. CONCLUSION

In this paper, we proposed a crowd-sourced framework that trained the NeRF model from the data captured by multiple production vehicles. This approach solved a key problem of large-scale reconstruction, that was where the data came from. We incorporated multiple improvements, such as ground surface supervision, occlusion completion, and sequence appearance embedding, to enhance the performance. Finally, the

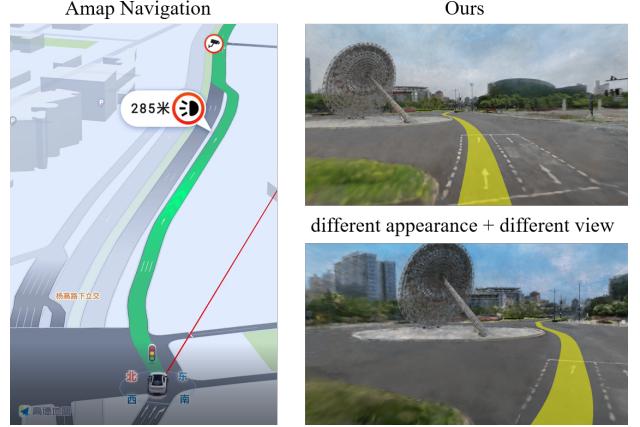


Fig. 14. The navigation application of the proposed system compared against Amap. Traditional navigation tools only provide 2D navigation pictures, while our approach can provide 3D images with different appearance styles and with different angles of view.

3D first-view navigation based on the NeRF model was applied to real-world scenarios.

Although the result from the proposed CS-NeRF framework seems great and promising, there are still several limitations and future works that are worth discussing:

- 1) Realistic Changes: Handling realistic changes (temporal inconsistency) is challenging during NeRF reconstruction. Temporary changes (e.g., lane redrawing) and long-term changes (e.g., road expansions) need to be accurately identified and incorporated into the reconstruction process.
- 2) Privacy Concerns: Due to the nature of user data, privacy concerns need to be addressed. Appropriate data processing and anonymization measures are implemented to ensure user privacy and data security.
- 3) Data Quality and Sensor Variations: Variations in data quality and sensor types across different platforms impact the accuracy of NeRF reconstruction. Data preprocessing, standardization, and calibration techniques are employed to mitigate these variations and ensure consistent data quality.

In summary, the paper proved the concept that crowd-sourcing way can be applied to the real world. In the future, the work will be extended to the automotive industry, where it can obtain a mass of driving data. Meanwhile, the NeRF model will contribute to the development of autonomous driving.

REFERENCES

- [1] J. L. Schönberger and J.-M. Frahm, "Structure-from-motion revisited," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [2] J. L. Schönberger, E. Zheng, M. Pollefeys, and J.-M. Frahm, "Pixel-wise view selection for unstructured multi-view stereo," in *European Conference on Computer Vision (ECCV)*, 2016.
- [3] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," in *ECCV*, 2020.

- [4] M. Tancik, V. Casser, X. Yan, S. Pradhan, B. Mildenhall, P. P. Srinivasan, J. T. Barron, and H. Kretzschmar, “Block-nerf: Scalable large scene neural view synthesis,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 8248–8258.
- [5] S. Agarwal, Y. Furukawa, N. Snavely, I. Simon, B. Curless, S. M. Seitz, and R. Szeliski, “Building rome in a day,” *Communications of the ACM*, vol. 54, no. 10, pp. 105–112, 2011.
- [6] A. Romanoni, D. Fiorenti, and M. Matteucci, “Mesh-based 3d textured urban mapping,” in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 3460–3466.
- [7] S. M. Seitz and C. R. Dyer, “Photorealistic scene reconstruction by voxel coloring,” in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, 1997, pp. 1067–1073.
- [8] J. T. Barron, B. Mildenhall, M. Tancik, P. Hedman, R. Martin-Brualla, and P. P. Srinivasan, “Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 5855–5864.
- [9] J. T. Barron, B. Mildenhall, D. Verbin, P. P. Srinivasan, and P. Hedman, “Mip-nerf 360: Unbounded anti-aliased neural radiance fields,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5470–5479.
- [10] K. Gao, Y. Gao, H. He, D. Lu, L. Xu, and J. Li, “Nerf: Neural radiance field in 3d vision, a comprehensive review,” *arXiv preprint arXiv:2210.00379*, 2022.
- [11] S. Zhi, T. Laidlow, S. Leutenegger, and A. J. Davison, “In-place scene labelling and understanding with implicit scene representation,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 15 838–15 847.
- [12] K. Rematas, A. Liu, P. P. Srinivasan, J. T. Barron, A. Tagliasacchi, T. Funkhouser, and V. Ferrari, “Urban radiance fields,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 12 932–12 942.
- [13] A. Kundu, K. Genova, X. Yin, A. Fathi, C. Pantofaru, L. J. Guibas, A. Tagliasacchi, F. Dellaert, and T. Funkhouser, “Panoptic neural fields: A semantic object-aware neural scene representation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 12 871–12 881.
- [14] Y. Siddiqui, L. Porzi, S. R. Bulò, N. Müller, M. Nießner, A. Dai, and P. Kortscheder, “Panoptic lifting for 3d scene understanding with neural fields,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 9043–9052.
- [15] P. Wang, L. Liu, Y. Liu, C. Theobalt, T. Komura, and W. Wang, “Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction,” *NeurIPS*, 2021.
- [16] L. Yariv, J. Gu, Y. Kasten, and Y. Lipman, “Volume rendering of neural implicit surfaces,” in *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021.
- [17] K. Deng, A. Liu, J.-Y. Zhu, and D. Ramanan, “Depth-supervised nerf: Fewer views and faster training for free,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 12 882–12 891.
- [18] H. Turki, D. Ramanan, and M. Satyanarayanan, “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*, 2022, pp. 12 922–12 931.
- [19] Y. Xiangli, L. Xu, X. Pan, N. Zhao, A. Rao, C. Theobalt, B. Dai, and D. Lin, “Bungeenerf: Progressive neural radiance field for extreme multi-scale scene rendering,” in *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXII*. Springer, 2022, pp. 106–122.
- [20] R. Martin-Brualla, N. Radwan, M. S. M. Sajjadi, J. T. Barron, A. Dosovitskiy, and D. Duckworth, “Nerf in the wild: Neural radiance fields for unconstrained photo collections,” in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 7206–7215.
- [21] K. Zhang, G. Riegler, N. Snavely, and V. Koltun, “Nerf++: Analyzing and improving neural radiance fields,” *arXiv preprint arXiv:2010.07492*, 2020.
- [22] C. Wu, “Towards linear-time incremental structure from motion,” in *2013 International Conference on 3D Vision-3DV 2013*. IEEE, 2013, pp. 127–134.
- [23] Z. Wang, S. Wu, W. Xie, M. Chen, and V. A. Prisacariu, “NeRF—: Neural radiance fields without known camera parameters,” *arXiv preprint arXiv:2102.07064*, 2021.
- [24] C.-H. Lin, W.-C. Ma, A. Torralba, and S. Lucey, “Barf: Bundle-adjusting neural radiance fields,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 5741–5751.
- [25] E. Sucar, S. Liu, J. Ortiz, and A. J. Davison, “imap: Implicit mapping and positioning in real-time,” in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 6209–6218.
- [26] Z. Zhu, S. Peng, V. Larsson, W. Xu, H. Bao, Z. Cui, M. R. Oswald, and M. Pollefeys, “Nice-slam: Neural implicit scalable encoding for slam,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.
- [27] W. Bian, Z. Wang, K. Li, J.-W. Bian, and V. A. Prisacariu, “Nope-nerf: Optimising neural radiance field with no pose prior,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 4160–4169.
- [28] C. Ruhhammer, M. Baumann, V. Protschky, H. Kloeden, F. Klanner, and C. Stiller, “Automated intersection mapping from crowd trajectory data,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 3, pp. 666–677, 2016.
- [29] C. Kim, S. Cho, M. Sunwoo, and K. Jo, “Crowd-sourced mapping of new feature layer for high-definition map,” *Sensors*, vol. 18, no. 12, p. 4172, 2018.
- [30] T. Qin, Y. Zheng, T. Chen, Y. Chen, and Q. Su, “A light-weight semantic map for visual localization towards autonomous driving,” in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 11 248–11 254.
- [31] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.
- [32] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.
- [33] V. Badrinarayanan, A. Kendall, and R. Cipolla, “Segnet: A deep convolutional encoder-decoder architecture for image segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.
- [34] C. Yu, C. Gao, J. Wang, G. Yu, C. Shen, and N. Sang, “Bisenet v2: Bilateral network with guided aggregation for real-time semantic segmentation,” *International Journal of Computer Vision*, vol. 129, pp. 3051–3068, 2021.
- [35] M. Tancik, E. Weber, E. Ng, R. Li, B. Yi, J. Kerr, T. Wang, A. Kristoffersen, J. Austin, K. Salahi, A. Ahuja, D. McAllister, and A. Kanazawa, “Nerfstudio: A modular framework for neural radiance field development,” *arXiv preprint arXiv:2302.04264*, 2023.
- [36] T. Müller, A. Evans, C. Schied, and A. Keller, “Instant neural graphics primitives with a multiresolution hash encoding,” *ACM Trans. Graph.*, vol. 41, no. 4, pp. 102:1–102:15, Jul. 2022.
- [37] Z. Wang, E. Simoncelli, and A. Bovik, “Multiscale structural similarity for image quality assessment,” in *The Thirty-Seventh Asilomar Conference on Signals, Systems & Computers*, 2003, vol. 2, 2003, pp. 1398–1402 Vol.2.
- [38] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, “The unreasonable effectiveness of deep features as a perceptual metric,” in *CVPR*, 2018.
- [39] M. Kazhdan, M. Bolitho, and H. Hoppe, “Poisson surface reconstruction,” in *Proceedings of the fourth Eurographics symposium on Geometry processing*, vol. 7, 2006, p. 0.



Tong Qin received the B.Eng degree in control science and engineering from the Zhejiang University, China, in 2015, and the Ph.D. degree in the Department of Electronic and Computer Engineering, the Hong Kong University of Science and Technology, HongKong, in 2019. He worked as a staff research scientist in the department of Advanced Driving Solution, Huawei from 2019 to 2023. He is currently working as an associate professor in Global Institute of Future Technology, Shanghai Jiao Tong University. His research interests include SLAM, NeRF, machine learning, and autonomous driving.



Changze Li received the B.Eng. degree in control science and engineering from Xidian University, China, in 2019, and the M.S. degree in Navigation Guidance and Control, the Northwestern Polytechnical University, China, in 2022. He worked as a research engineer in the department of Advanced Driving Solution, Huawei. He is currently a PhD candidate in Shanghai Jiao Tong University. His research interests include NeRF, computer vision in the field of autonomous driving.



Ming Yang received his Master's and Ph.D. degrees from Tsinghua University, Beijing, China, in 1999 and 2003, respectively. Presently, he holds the position of Distinguished Professor at Shanghai Jiao Tong University, also serving as the Director of the Innovation Center of Intelligent Connected Vehicles. Dr. Yang has been engaged in the research of intelligent vehicles for more than 25 years.



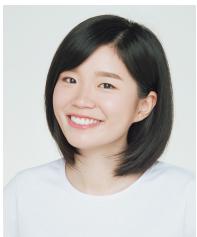
Haoyang Ye received the B.Eng. degree in automation from the College of Control Science and Engineering, Zhejiang University, China, in 2016, and the Ph.D. degree from the Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology, Hong Kong, in 2020.

He is currently working as a Research Engineer with Shanghai Huawei Technologies Co., Ltd., Shanghai, China. His research interests include state estimation for robotics, SLAM, sensor fusion, and computer vision.



Shaowei Wan received the B.Eng. degree in Robot Engineering from Northeastern University, China, in 2020, and the M.S. degree in Control Engineering, Huazhong University of Science and Technology, China, in 2022.

He is currently working as a research engineer in the department of Advanced Driving Solution, Huawei. His research interests include NeRF and autonomous driving.



Minzhen Li received her B.Eng. degree in Geographic Information Systems from Tongji University in 2013, followed by her M.S. degree in Surveying Engineering from the same university in 2016.

She is a research engineer working at Huawei in the field of autonomous driving since 2016. Her research interests include Mapping & Localization, Auto labelling in the field of autonomous driving.



Hongwei Liu received the B.Eng. degree in Automobile Engineering from Chongqing Jiaotong University, China, in 2017, and the M.S. degree in Automobile Engineering, Tongji University, China, in 2020.

She is currently working as a research engineer in the department of Advanced Driving Solution, Huawei. Her research interests include SLAM and autonomous driving.