This ACCV 2024 paper, provided here by the Computer Vision Foundation, is the author-created version.
The content of this paper is identical to the content of the officially published ACCV 2024
LNCS version of the paper as available on SpringerLink: https://link.springer.com/conference/accv

EDeRF: Updating Local Scenes and Editing Across Fields for Real-Time Dynamic Reconstruction of Road Scene

Zhaoxiang Liang[1], Wenjun Guo[1], Yi Yang[1⋆], and Tong Liu[1]

School of Automation, Beijing Institute of Technology, Beijing, China

Abstract. NeRF provides high reconstruction accuracy but is slow for dynamic scenes. Editable NeRF speeds up dynamics by editing static scenes, reducing retraining and succeeding in autonomous driving simulation. However, the lack of depth cameras and the difficulty in obtaining precise vehicle poses make real-time dynamic road scene reconstruction challenging, particularly in swiftly and accurately reconstructing new vehicles entering the scene and their trajectories. We propose EDeRF, a method for real-time dynamic road scene reconstruction from fixed cameras such as traffic surveillance through collaboration of sub-NeRFs and cross-field editing. We decompose the scene space and select key areas to update new vehicles by sharing parameters and local training with sub-fields. These vehicles are then integrated into the complete scene and achieve dynamic motion by warping the sampling rays across different fields, where vehicles' six degrees of freedom(6-DOF) is estimated based on inter-frame displacement and rigid body contact constraints. We have conducted physical experiments simulating traffic monitoring scenes. Results show that EDeRF outperforms comparative methods in efficiency and accuracy in reconstructing the appearance and movement of newly entered vehicles.

Keywords: Real-time 3D Reconstruction · Editable Radiance Fields · Intelligent Traffic Monitoring

1 Introduction

Real-time dynamic 3D reconstruction within a pre-built road environment is crucial for robotics, intelligent traffic systems and autonomous driving simulations. A typical application is intelligent surveillance of roads. Performing real-time dynamic reconstruction of vehicles within heavily trafficked scenes can significantly improve capabilities of vehicular forensics and traffic management.

However, achieving this task in real-world road scenarios without extensive depth cameras is challenging, due to the need for fast and accurate reconstruction of new vehicles entering the environment and their moving trajectories. Most traditional methods based on mesh or point cloud for real-time dynamic

⋆ Correspondence: yang_yi@bit.edu.cn

3D reconstruction rely heavily on depth cameras [2, 26]. Due to their lack for reconstruction speed, some work [1] use generic templates to represent moving vehicles, preventing them from capturing the true appearance and details of the vehicles. NeRF [23] has high-fidelity reconstruction capabilities, but it's very time-consuming to train dynamic scenes. Although Instant NGP(INGP) [24] have greatly reduced training time of static to tens of seconds, directly applying NeRF to the mentioned scenarios still presents difficulties: 1) Existing methods require a large number of cameras to cover the complete 360° scene and results in heavy data transmission volumes [15, 32, 39] , which is difficult to ensure in real road. 2) Dynamic NeRF methods lack real-time capabilities [4, 15, 28]. For example. Hex-plane [3] decomposes 4D scenes to accelerate, but still needs 4 hours to reconstruct a 10-second dynamic scene on a kitchen bar. D-NeRF and some works [28, 30, 32] treat dynamics as deformations and train to map each frame from a canonical field to the deformation field, which is also time-consuming.

Some works [12, 18, 37, 40] propose editable fields, which efficiently achieve object motion in implicit space by warping sampling rays instead of retraining. Some of them use proxy meshes for editing [29,40], ideal for traffic scenes as they separate dynamic vehicles within 3D bounding boxes from the static background. This feature is also applied in autonomous driving simulations such as Mars [39]. Additionally, GPS, IMU, and roadside BEV data can guide dynamic vehicles as rigid bodies, reducing the need for densely sampled video viewpoints.

However, editable NeRF methods face several challenges in our real-time vehicle road monitoring task: First, most methods [12, 18, 29, 37, 40] are limited to dealing with objects that already exist within the reconstructed scene. But in traffic monitoring tasks, new vehicles enter the scene from various entry points. Introducing new objects requires retraining the entire NeRF, which is a time-consuming process and significantly degrades real-time performance. Second, the dynamic reconstruction results of editable NeRF heavily depend on the input movement information of the target object [39]. In reality, inputs from sensors or visual localization are often imprecise, causing clipping, floating, and misalignment. Moreover, temporary input absences can result in reconstruction failures.

To address these challenges, we propose a real-time reconstruction method based on fast local updating and editable NeRF for dynamic road scenes. First, we decompose the scene space, using sub-fields to handle the static background and new vehicles individually. By sharing network parameters, training local area and implementing composite rendering, we address the challenge of quickly and faithfully integrating new objects into a pre-built static scene. Second, we extend editable field from operating in a single radiance field to multiple sub-fields. By manipulating ray warping to sample across sub-fields, we integrate vehicles into the complete scene and achieve dynamic motion, and accurately reflecting occlusion relationships in composite rendering of sub-fields. Third, we estimate the 6-DOF of vehicles using sequences of target coordinates and scene information to guide their movement, reducing the reliance on precise pose inputs for editing. The process involves two steps: initially, we roughly estimate

the rotation angles through displacement projection between frames. Then, we refine pose from road constraints, which is achieved by assessing the collision relationship between the vehicle and the road surface through sampling some volumetric density points. We perform our approach on a series of real-world experiments. In summary, our contributions are as follows:

- We propose a method for fast updating part of static scenes through scene decomposition, network parameters sharing and collaboration between different sub-fields. It addresses the challenge of quickly reconstructing new targets within pre-built scenes.
- We propose a method for real-time reconstruction of dynamic road scenes with multi-vehicles by cross-field editing. By warping sampling rays across sub-fields, we efficiently integrate vehicles and achieve dynamic motion while accurately reflecting occlusion relationships in composite rendering. It addresses the challenge of maintaining both real-time performance and quality.
- We propose a two-step method for estimating the 6-DOF of vehicles, including roughly estimating rotation angles through displacement projection between frames first and then refining based on volumetric density-based geometric contact constraints. It addresses the challenge of heavy reliance on precise pose inputs for dynamic by editing.

2 Related Work

2.1 Static 3D scene Reconstruction

Methods for 3D scene representation include meshes, point clouds, signed distance functions (SDF), voxels, NeRF and 3D Gaussian. Some works [8] use triangulation and Poisson surface reconstruction for efficient rendering, though high-quality mesh reconstruction is time-cosuming [14]. Structure from Motion (SFM) [31] represents scenes with point clouds, while Multi-View Stereo (MVS) [9] enhances point density and performance. Some works [27] use SDF to reconstruct textureless geometries. And some other work leverages neural volumes [22] or multi-plane depiction [33].

NeRF [23] leverages volume rendering to reconstruct 3D scenes with high fidelity from images. Lots of NeRF works aim to enhance scene quality, efficiency, performance in sparse views or large-scale scenes. Some works improve efficiency by accelerating sampling [11, 25] or using hybrid representations with sparse data structures such as octrees [17, 43]. InstantNGP [24] employ hash encoding and multi-resolution frameworks to reduce training times to tens of seconds and enable real-time rendering. Blocknerf [34,38] employ multiple sub-NeRFs to represent a complete scene, greatly boosting efficiency in large-scale environments by replacing a single large MLP with several smaller MLPs.

2.2 Dynamic Scene Reconstruction

Many classic works exist for dynamic scene reconstruction. Some works require extensive depth cameras [1], LiDAR [26], cameras and optical flow input [7]. Some works [2] interpolate dynamic scenes but have small viewpoint changes.

Dynamic NeRF reconstruct realistic dynamic scenes from video but is very time-consuming. Work like VidoNeRF [4, 15] treats each frame as a static scene. D-NeRF [28,30] train to map each frame from a canonical field to the deformation field, still struggle with real-time capability and larger 360° scenes. Hexplane [3,5] enhance efficiency by multi-plane decomposition or hash grids to reduce MLP size. NeRFPlayer [32] achieve real-time rendering for the result, but still needs hours training for a 10s video.

Editable NeRF [6, 12] is a specific type of deformation field, it offer the possibility to quickly realize dynamic by editing. Editable NeRF creates a proxy mesh in reconstructed scenes, using ray warping to apply mesh edits in volumetric space for scene content editing. Editable NeRFs have a low cost for avoiding retraining, some works apply them to dynamic scenes treating vehicles as rigid bodies [21, 39]. Mars [39] leverages editing for vehicles movement simulations, achieving high-quality result. Yet, it lacks real-time capability, demands tens of hours of training, and depends on precise 3D vehicle bounding boxes and additional data for each frame, challenging to achieve in practice. Current editable methods quickly modify existing content but fail to rapidly reconstruct emerging objects.

2.3 3D Detection and 6-DOF Estimation

For 6-DOF estimation, works [19] require precise 3D models for guidance. Vehicles commonly use GPS, IMU, and integrated navigation for pose. 3DRCNN [10] estimates poses in images, while 3DSSD [42] leverage LiDAR for enhanced outcomes. Vision-centric bird's eye view (BEV) detection methods [13, 16, 41] use queries and image features for view transformation. Roadside monitoring-based BEV methods like BEVHeight [41], predict pixel heights instead of depths, suit for 3D road monitoring.

Pose information in reality includes noise. Liu et al. [20] correct sensor poses with explicit maps. But NeRF don't have clear geometric data. Nerf2Mesh [36] can generate meshes from volumetric density, shifting to a traditional approach. However, these require extensive training hours.

3 Method

Our work focuses on the rapid reconstruction of newly emerged vehicles and achieving dynamic process within a complete scene. Our pipeline consists of 3 key parts as shown in Fig. 1: **1)** We efficiently update and reconstruct newly appearing vehicles in the complete reconstructed scene by leveraging scene decomposition, parameter sharing, and combined rendering strategies. (Sec. 3.1).

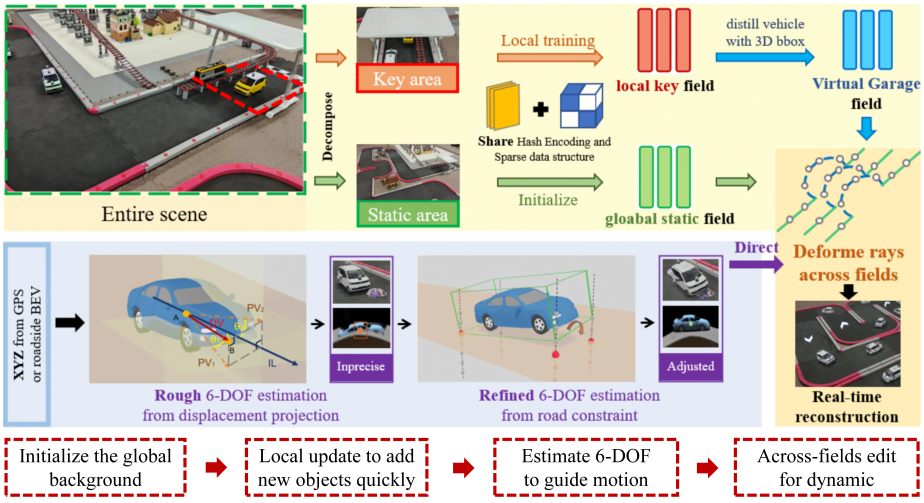

Fig. 1: An overview of our pipeline, our approach mainly consists of 3 key parts.

2) We achieve vehicle motion in the full scene by extending editing deformation methods across multiple neural fields (Sec. 3.2). **3)** We estimate complete 6-DOF from vehicle XYZ sequences and geometric constraints based on volumetric scene density when precise 6-DOF input is unavailable (Sec. 3.3).

The general process is as follows: First, we reconstruct static scenes from videos and align coordinate systems of different fields as initialization. The camera parameters of images are obtained from COLMAP [31]. The neural field of static scene will stop training once the reconstructed background is sufficiently clear. Then it will only participate in composite rendering to conserve computational resources. When new vehicles enter the key area, fixed cameras synchronously capture images to detect the 3D-Box and input images to train the key field. Shortly after, vehicles are distilled from the key area to the virtual garage based on the 3D-Box, which serves as the vehicle's proxy mesh. We achieve multi vehicles' motion in the entire scene by editing across the field of the static scene and the garage. We estimate the 6-DOF to guide vehicle motion from the projection of displacements and road constraints, interpolating for time steps where the XYZ data is missing.

3.1 Scene Decomposition and Fast Reconstruction of New Vehicles

We divide the scene into static background and key areas, represented by global static field and local key field respectively. We use another field as a virtual garage. We first explain these areas separately, as shown in Fig. 1:

– **Static background**, such as roads and brick buildings, has a constant location and appearance. They don't need frequent update after initialization.

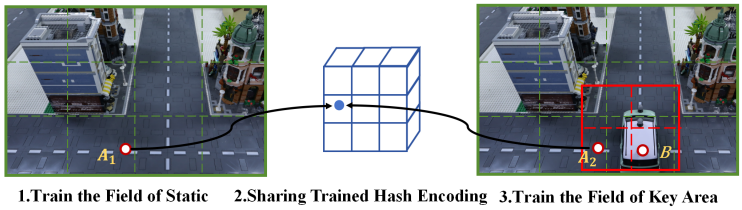

Fig. 2: Sharing hash encoding. Since point $A2$ in the key field can directly query its attribute values from the pretrained hash table, backpropagation gradients focus more on reconstructing the new vehicle.

- **Key areas** serve as entry points or necessary paths for new vehicles into the full scene, such as parking lot checkpoints or campus driveway entrances. Fixed cameras here don't need frequent re-calibration. As the full scene's scale grows, it's not necessary to increase the number of cameras here.
- **The virtual garage** is used to maintain all new vehicles learned by the key field. Otherwise, updating images in key areas and training the key field can overwrite historical vehicle information.

We modify Instant-NGP for static 3D scenes reconstruction. Quickly adding new vehicles into the static background is challenging due to the extensive images required for high-quality, large-scale reconstruction. Retraining the entire scene wastes computation on the static background. Adjusting sampling on new vehicles will degrade the background quality.

We address this with a multi-field approach: First, we make the MLP of key radiance field more focused by constraining the key field's sampling points to only key areas; Second, we use the pre-trained hash encoders and bitfields from the global radiance field to initialize the local radiance field. Because the hash tables and density bitfields in INGP almost explicitly store the density information of points in space as shown in Eq. 1. Note T as hash table size, d as the dimension of the input vector, and x_i are the components of the input, π_i are unique prime numbers, the spatial hash function is given in the form:

$$h(\mathbf{x}) = \left(\bigoplus_{i=1}^{d} x_i \pi_i \right) \mod T \tag{1}$$

$\bigoplus$ denotes the bit-wise XOR operation and in INGP $\pi_1 = 1$, $\pi_2 = 2654435761$,$\pi_3 = 805459861$. We share parameters to avoid retraining the entire encoding, and the bitfields help quickly skip over a large number of meaningless points. As shown in Fig. 2, the hash encoding obtained from training the static scene radiance field is shared with key areas field for initialization. For points A_1 and $A2$ with the same coordinates, the same hash encoding directs them to the same position in the hash table, allowing to obtain the attribute values of these sampling points immediately. For sampling points like point B within the 3D bounding box of newly appearing vehicles, training is conducted after setting the right bitfield

attributes to valid to avoid overlooking. This strategy helps focus the gradients on the training of new vehicles during backpropagation. After that, we distill new vehicles learned into the "virtual garage" by re-sampling in the box.

3.2 Reconstruct Vehicles' Dynamic Process by Cross-field Editing

Most editable NeRF methods only support warping rays within a single radiance field. Our method extends this capability to efficiently warp rays between multiple fields. This allows for low-cost integration of reconstructed vehicles and static scenes, thereby improving real-time performance. The task of aligning the coordinate systems of different radiance fields has already been completed during initialization. During rendering, if a sampling ray passes through a vehicle's proxy mesh, we use the garage's radiance field to calculate the color and opacity of the sampling points within the mesh. For sampling points outside the proxy mesh, we use the radiance field of the static scene. Note $Field(x, y, z)$ as which radiance field we use to calculate the properties of point (x, y, z):

$$Field(x, y, z) = \begin{cases} Field_{garage}(x, y, z), if(x, y, z) \in BBox_{Vehicles} \\ Field_{global}(x, y, z), else \end{cases} \tag{2}$$

In this way, we can appropriately replace the sampling points along the rays and accurately obtain the occlusion between vehicles and the scene, as well as between vehicles, in the volumetric rendering results.

3.3 Two-step 6-DOF Estimation Method

We developed a two-step method assuming vehicles' wheels attempt to align with the road surface, enabling full pose estimation from XYZ sequence and sampling scene information, as shown in the part of Fig. 1. For the input pose, rotation's accuracy is often lower than position's. We start with the vehicle's initial pose as global initialization, using each frame to initialize the next.

First, due to the consistent rotation and direction caused by small frame intervals, we calculate the frame-to-frame displacement vector from the XYZ sequence and project it onto the vehicle's coordinate plane. By calculating the angle between the vector and its projection, we obtain preliminary estimates of the heading and pitch angles. Note D as the inter-frame displacement, note $proj_{xz}$ as the projection of D on vehicle's XZ plane, RT as the vehicle's transform matrix in NeRF, p as the coordinates of vehicle, and θ as the rotation angle for the vehicle:

$$\begin{aligned} &D = R^T(p_{new} - p_{old}), proj_{xz} = \sqrt{|D|^2 - (D.y)^2}, \\ &\theta_{yaw} = sig(RT.x[2] \times RT.z[2]) \cdot \arccos\left(\frac{|D.x|}{proj_{xz}}\right) \\ &\theta_{pitch} = sig(RT.x[2] \times RT.y[2]) \cdot \arcsin\left(\frac{|D.y|}{D.norm}\right) \end{aligned} \tag{3}$$

Second, assuming vehicles are rigid bodies not flying, we refine initial poses with rigid contact constraints between the vehicle and the road surface by sampling. We achieve it by querying volumetric density around the vehicle's bottom, instead of extracting full surfaces from NeRF. We map the vehicle to a rough pose within the global scene. As shown in Fig. 1, by sampling 5 points along each line extending vertically from the 4 bottom corners of the BBox, we query the density from the sub-NeRF of global scene to approximate the road surface's position. Then we assess the vehicle-road relationship, including floating, clipping through the road, or tilting to a side. Then we incrementally adjust the pose. The first sampling point t greater than the gradient threshold is considered as the road surface:

$$\begin{aligned} &if : |\nabla D(x_t, y_t, z_t)| > \tau, \quad t \in surface \\ &otherwise : not \end{aligned} \tag{4}$$

For occasional missing time step inputs, we use Lagrange interpolation method to estimate their XYZ. Considering road complexity and randomness, we use position data from $5s$ before and after the target time to interpolate, then correcting with mentioned volumetric density-based geometric contact constraints. Note $l_i(t)$ as the Lagrange basis function, $L(x, y, z, t)$ as the interpolation result:

$$\begin{aligned} &L(x, y, z, t) = (L(x, t), L(y, t), L(z, t)) \\ &= \sum x_i * l_i(t) + \sum y_i * l_i(t) + \sum z_i * l_i(t) \\ &l_i(t) = \prod_{j=0, j\neq i}^{n} \frac{t - t_j}{t_i - t_j}, t_j = t_i + 5 \end{aligned} \tag{5}$$

4 Experiment

We demonstrate the performance of our approach to reconstruct dynamic multi-vehicle scenes in real-time through a series of indoor and outdoor experiments in real-world settings. Additionally, we conduct two ablation studies to verify the effectiveness of the shared parameter strategy in the new vehicle reconstruction module and the effectiveness of road constraints in the 6-DOF estimation.

4.1 Data

Our method learns new vehicles entering the entire scene from fixed surveillance cameras in the local key area. Therefore, we require images for static reconstruction and images from fixed cams in areas like toll booths. Given the lack of similar public datasets, we built a series of physical experimental scenes and collected data for testing to ensure the validity of comparison in Fig. 3.

Indoor Scene The main indoor setup includes an RC car track and a train track, with their starting positions as the local key area in Fig. 3a. A truss platform with 16 fixed cams captures images simultaneously when new vehicles enter. Static reconstruction images of resolution 1920x1080 are recorded with

(a) Main exp. site and partial data. (b) Part of data.

Fig. 3: We designed physical experiments simulating traffic monitoring scenes. Fixed cameras on the marked truss continuously update images of the key area.

a cam. We first reconstruct an empty local key area as a training checkpoint. Then, fixed cams capture vehicles as they enter it. Vehicle positioning signals are provided by a motion capture system mimicking GPS.

Outdoor Scene We used a handheld camera to capture video in a parking lot as shown in Fig. 3b, extracting 300 images for static reconstruction. Although some artifacts exist due to camera shake, they don't affect our method's workflow. The onboard navigation system provided the vehicle's position in the map coordinate system, which was aligned with the NeRF space.

4.2 Baseline

In the static scene update experiments, we compared Nerfacto [35], INGP [24], and INGP with the combined field strategy, with INGP demonstrating SOTA performance in reconstruction speed. For the dynamic reconstruction comparison experiments, we chose NeRF-T, Hexplane [3], and INGP-T as baselines.

4.3 Implementation Details

The Optitrack motion capture system we used provides vehicle coordinates at a frequency of 10Hz. The cameras on the truss platform are connected to a Mivii industrial PC that is part of the local network. The intrinsic and extrinsic parameters of the cameras are estimated using COLMAP [31]. By considering camera calibration and the results provided by COLMAP together, the extrinsic parameters of the fixed cameras are unified into the world coordinate system.

We conduct experiments on an RTX Laptop 4080. Our code is primarily written in CUDA and C++, including both the reconstruction part and the ROS-based coordinate transmission part. We use the GStreamer framework for real-time image transmission.

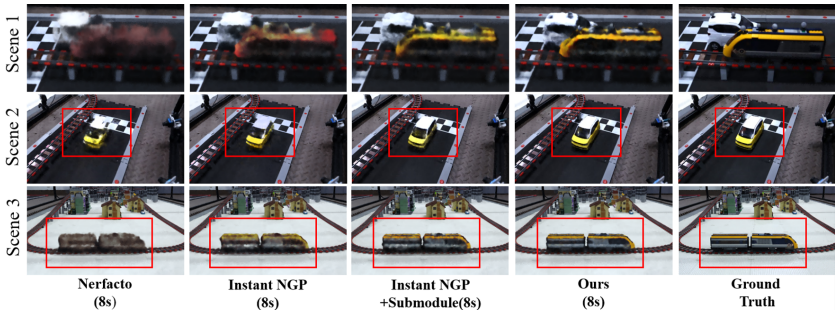

Fig. 4: Newly entering objects update speed experiment. PSNR, SSIM, LPIPS in regions highlighted in red are calculated.

Table 1: Quantitative comparison results of indoor exps. on new objects update.

	Scene 1			Scene 2			Scene 3		
Method	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LIPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Nerfacto	18.908	0.716	0.386	21.685	0.834	0.274	19.275	0.549	0.578
INGP	18.974	0.713	0.375	23.961	0.869	0.232	20.428	0.586	0.532
INGP+submodules	21.398	0.752	0.337	25.411	0.895	0.206	21.948	0.625	0.499
Ours	**23.387**	**0.815**	**0.325**	**29.082**	**0.929**	**0.165**	**24.055**	**0.699**	**0.453**

4.4 Result

Updating Local Areas We test the speed of updating local areas in indoor experiments as shown in Fig. 4. All methods share the same images, starting with a 20-minute training of an empty scene as a checkpoint. Upon new vehicle enter, cameras captured a simultaneous shot, followed by 8 seconds of training. Our method outperformed others, including CUDA version INGP with sub-modules strategy. It also serves as an ablation study, demonstrating the effectiveness of sharing parameters between fields in our method. We calculate PSNR, SSIM, LPIPS in regions highlighted within red, as shown in Tab. 1. Additional exp. on the 3DGS is in the Supple.

Indoor Dynamic Process Reconstruction We initially trained the empty static scenes without vehicles using different methods and saved the training checkpoints. The dynamic process contains 15 frames once vehicles enter. In Fig. 5, the static scene reconstruction time is shown on the left side of the brackets below each method name, and the dynamic reconstruction time on the right. Our render effect at 15s is close to INGP-T at 6min, but minor deviations due to vehicle pose errors decrease pixel-wise metric PSNR, as shown in Tab. 2. Notably, 15s for 15 frames is too short for INGP-T, making the average training of all frames meaningless as the results are nearly identical to an empty scene. So we set the first frame's training time to 15s, yielding geometrically clear but detail-deficient results, akin to Fig. 4. Then we compute the average

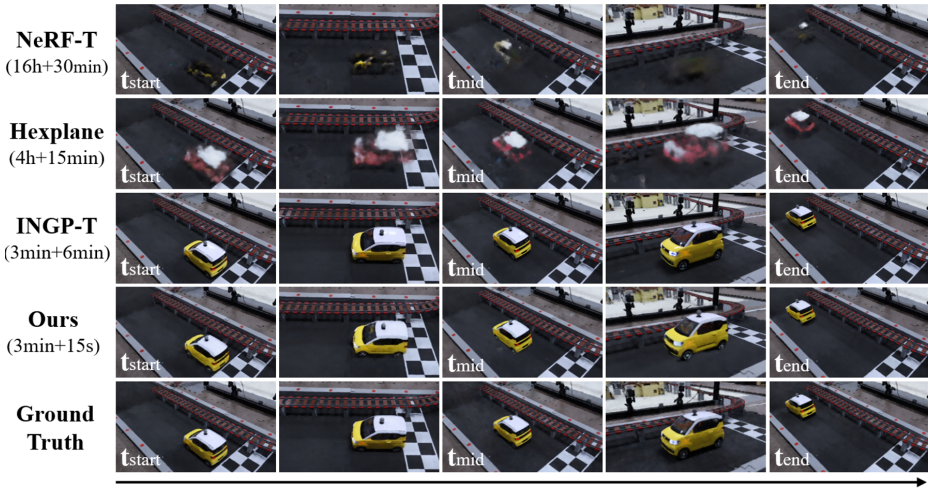

Fig. 5: Results of dynamic process reconstruction quantitative experiments.

Table 2: Quantitative comparison results on dynamic process experiments.

Method	PSNR↑	SSIM↑	LPIPS↓	Input Images↓	Training Time↓
NeRF-T	19.171	0.816	0.379	240	30min
Hexplane	22.979	0.819	0.381	240	15min
INGP-T	**26.077**	**0.842**	**0.339**	240	6min
INGP-T	19.435	0.818	0.376	240	15s
Ours	22.995	0.823	0.348	**61**	**15s**

metrics across 15 frames. Our editing-based method outperforms in real-time performance. Our method used 16 images of the first frame to reconstruct the vehicle's appearance, while the remaining to detect and provide vehicle positions at the corresponding time steps. The mentioned 15s time in the bracket is solely dedicated to reconstructing the vehicle in the first frame; subsequent frames were modified through editing without training. As the number of frames increases and the dynamic process extends, the advantages of our method become more evident, because other methods require reconstructing each frame from scratch. This is also a key factor enabling us to achieve real-time performance.

As shown in Fig. 6, our approach continues dynamic reconstruction when vehicles drive out of a well-covered local area, by utilizing the previously learned yellow car model. The 6-DOF of the vehicles are estimated by our 2-step method, with necessary coordinates provided by both Optitrack and roadside BEV methods. Our approach renders at a resolution of 807x454 at 20fps, 1.6s delay. As new vehicles enter the scene, our approach rapidly integrates cars into the scene, maintaining real-time interactive rendering. New vehicles rendered in reconstruc-

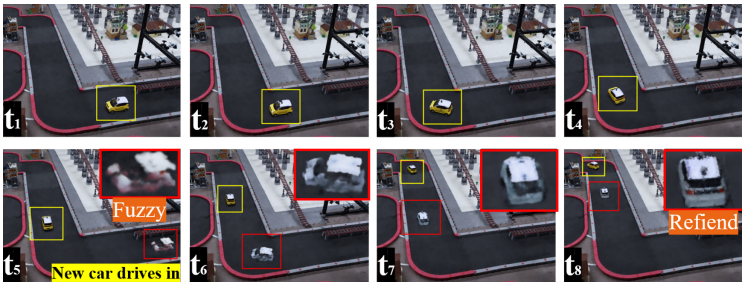

Fig. 6: Results of dynamic reconstruction without cameras well-covered.

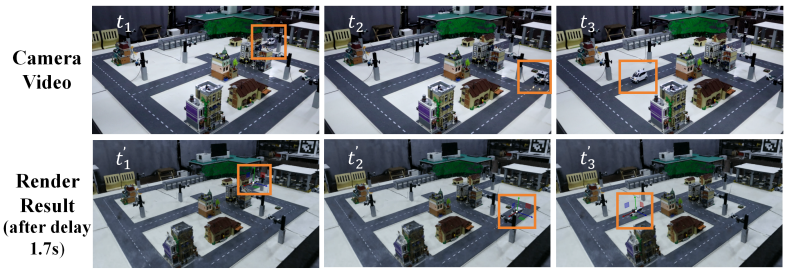

Fig. 7: Real-time dynamic reconstruction results. The 2nd row displays the rendered result after a 1.7s delay, with vehicle positions in orange boxes.

tion space change from an initial fog-like state to a refined appearance for 16s, and the frame rate drops to 6fps during this process.

We also conducted experiments in a LEGO town model. Our tests showed that at a rendering resolution of 807x454, our real-time reconstruction method had a delay of approximately 1.7 seconds. The first row shows the camera footage, while the second row displays the delayed rendering results, with the vehicle positions marked by yellow boxes in Fig. 7. It is important to note that the camera resolution differs from the rendering resolution.

Outdoor Dynamic Process Reconstruction We reconstruct two driving sequences along different routes in an outdoor parking lot, and present real-time rendering results at 4 different moments from 2 perspectives for each scene as shown in Fig. 9. We utilized the onboard integrated navigation system to obtain vehicle coordinates(XYZ) and estimate the complete pose. Affected by network signals delay, the total system delay from the real vehicle movement to rendering the corresponding images is approximately around 6s, still reaching real-time standard. The accompanying video contains more information.

4.5 Ablation Study

Effectiveness of sharing parameters To validate its effectiveness in enhancing the update speed of local areas, we compare our method with INGP with

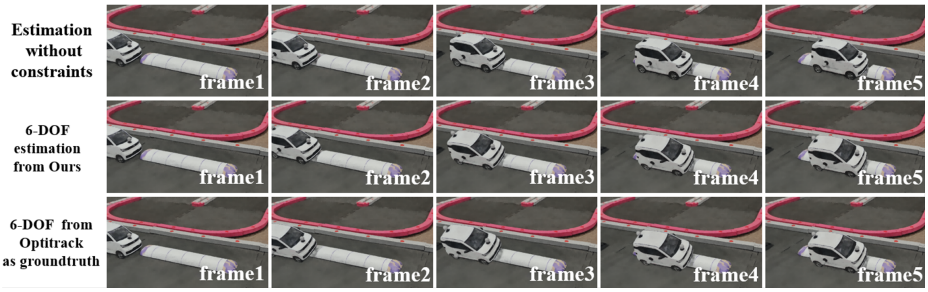

Fig. 8: Experiment on 6-DOF estimation with ablation studies. Results show that the introduced road constraints effectively enhance scene interaction and prevent clipping.

Table 3: Accuracy of Our 6-DOF Estimation in Our Experiment

Error Range	Accuracy
$\leq 10^{\circ}$	63.2%
$\leq 30^{\circ}$	85.7%

submodules strategy in Fig. 4. The results show that sharing hash encodings and density bitfields for initialization effectively improves the update speed of the key field, as evidenced in Tab. 1. This approach essentially shares the scene information learned by the implicit radiance field with the local field through an explicit storage structure.

Effectiveness of road constraints To validate the effectiveness of volumetric density-based geometric constraints for estimating 6-DOF from XYZ, we design an experiment that a long strip-shaped silicone semicylinder placed on the left side of the racetrack, causing the remote-controlled car to lift and change its rotation angle as it moves forward. As shown in Fig. 8, vehicles would clip through the model without road constraints. Our method enables organic interaction between vehicles and the scene by sampling. Upon detecting a significant increase in volumetric density around the left tire, it adjusts the roll angle to align with expectations. We selected 80 instances from dynamic sequences within the full track and rail system, including curves and slopes, using Optitrack as ground truth to assess the accuracy of our 6-DOF estimation. The quantitative results are shown in the Tab. 3.

5 Conclusion

Existing methods fail to reconstruct and monitor the appearance and motion of multiple vehicles in road scenes in real time. In this paper, we address this task based on the concept of key areas, dynamic-static separation, utilizing editable fields as the foundation. Our system supports real-time interaction and

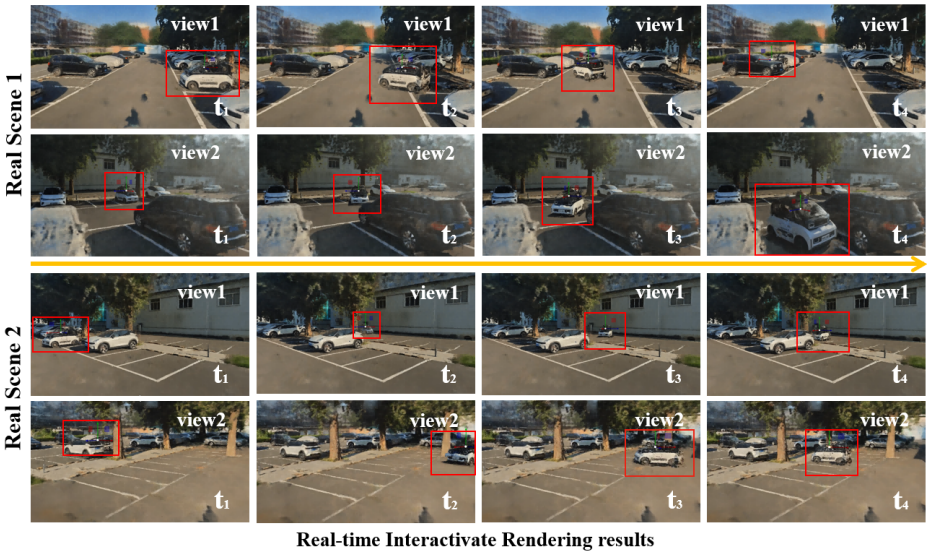

Fig. 9: Outdoor vehicle driving reconstructed in real-time. The first 2 rows show a longer distance, the last 2 rows a shorter one.

rendering at resolution of 807*454, 20fps with 2s delay. Based on the experiments presented, the following conclusions can be drawn: (1) Efficiently update and reconstruct newly appearing vehicles in the complete scene by leveraging scene decomposition, parameter sharing, and combined rendering strategies. (2) Handle vehicle motion across the full scene by extending editing deformation methods across multiple neural fields. (3) Estimate the 6-DOF to guide vehicle motion using the projection of displacements and road constraints, thereby reducing the reliance on precise pose inputs.

Limitations and future work. We restricted continuous updating of the global scene because this would result in performance overhead and real-time capability decline. This area merits future exploration. Furthermore, update vehicle appearances while maintaining real-time performance would broaden applications, and implement this task using 3D Gaussians is worth attempting.

Acknowledgement. This work was partly supported by National Natural Science Foundation of China (Grant No. 62233002, U1913203, 61973034 and CJSP-Q2018229) and the BIT Research and Innovation Promoting Project (Grant No.2023YCXY033). We would like to thank Yu Gao, Tao Wang, Xiaodong Guo, Tianji Jiang, Kai Yu, Dianyi Yang, Jiadong Tang, and Bohan Ren for their help and guidance in writing this paper and constructing the experimental site.

References

1. Bârsan, I.A., Liu, P., Pollefeys, M., Geiger, A.: Robust dense mapping for large-scale dynamic environments. In: 2018 IEEE International Conference on Robotics

and Automation (ICRA). pp. 7510–7517. IEEE (2018)
2. Bemana, M., Myszkowski, K., Seidel, H.P., Ritschel, T.: X-fields: Implicit neural view-, light-and time-image interpolation. ACM Transactions on Graphics (TOG) **39**(6), 1–15 (2020)
3. Cao, A., Johnson, J.: Hexplane: A fast representation for dynamic scenes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 130–141 (2023)
4. Du, Y., Zhang, Y., Yu, H.X., Tenenbaum, J.B., Wu, J.: Neural radiance flow for 4d view synthesis and video processing. In: 2021 IEEE/CVF International Conference on Computer Vision (ICCV). pp. 14304–14314. IEEE Computer Society (2021)
5. Fridovich-Keil, S., Meanti, G., Warburg, F.R., Recht, B., Kanazawa, A.: K-planes: Explicit radiance fields in space, time, and appearance. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12479–12488 (2023)
6. Jambon, C., Kerbl, B., Kopanas, G., Diolatzis, S., Drettakis, G., Leimkühler, T.: Nerfshop: Interactive editing of neural radiance fields. Proceedings of the ACM on Computer Graphics and Interactive Techniques **6**(1) (2023)
7. Joo, H., Soo Park, H., Sheikh, Y.: Map visibility estimation for large-scale dynamic 3d reconstruction. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1122–1129 (2014)
8. Kazhdan, M., Bolitho, M., Hoppe, H.: Poisson surface reconstruction. In: Proceedings of the fourth Eurographics symposium on Geometry processing. vol. 7 (2006)
9. Kopanas, G., Philip, J., Leimkühler, T., Drettakis, G.: Point-based neural rendering with per-view optimization. In: Computer Graphics Forum. vol. 40, pp. 29–43. Wiley Online Library (2021)
10. Kundu, A., Li, Y., Rehg, J.M.: 3d-rcnn: Instance-level 3d object reconstruction via render-and-compare. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3559–3568 (2018)
11. Kurz, A., Neff, T., Lv, Z., Zollhöfer, M., Steinberger, M.: Adanerf: Adaptive sampling for real-time rendering of neural radiance fields. In: European Conference on Computer Vision. pp. 254–270. Springer (2022)
12. Li, S., Pan, Y.: Interactive geometry editing of neural radiance fields. arXiv preprint arXiv:2303.11537 (2023)
13. Li, Y., Ge, Z., Yu, G., Yang, J., Wang, Z., Shi, Y., Sun, J., Li, Z.: Bevdepth: Acquisition of reliable depth for multi-view 3d object detection. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 37, pp. 1477–1485 (2023)
14. Li, Z., Müller, T., Evans, A., Taylor, R.H., Unberath, M., Liu, M.Y., Lin, C.H.: Neuralangelo: High-fidelity neural surface reconstruction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8456–8465 (2023)
15. Li, Z., Niklaus, S., Snavely, N., Wang, O.: Neural scene flow fields for space-time view synthesis of dynamic scenes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6498–6508 (2021)
16. Li, Z., Wang, W., Li, H., Xie, E., Sima, C., Lu, T., Qiao, Y., Dai, J.: Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. In: European conference on computer vision. pp. 1–18. Springer (2022)
17. Liu, L., Gu, J., Zaw Lin, K., Chua, T.S., Theobalt, C.: Neural sparse voxel fields. Advances in Neural Information Processing Systems **33**, 15651–15663 (2020)

18. Liu, R., Xiang, J., Zhao, B., Zhang, R., Yu, J., Zheng, C.: Neural impostor: Editing neural radiance fields with explicit shape manipulation (2023), https://arxiv.org/abs/2310.05391
19. Liu, Y., Wang, T., Zhang, X., Sun, J.: Petr: Position embedding transformation for multi-view 3d object detection. In: European Conference on Computer Vision. pp. 531–548. Springer (2022)
20. Liu, Y., Wen, Y., Peng, S., Lin, C., Long, X., Komura, T., Wang, W.: Gen6d: Generalizable model-free 6-dof object pose estimation from rgb images. In: European Conference on Computer Vision. pp. 298–315. Springer (2022)
21. Liu, Y., Tu, X., Chen, D., Han, K., Altintas, O., Wang, H., Xie, J.: Visualization of mobility digital twin: Framework design, case study, and future challenges. In: 2023 IEEE 20th International Conference on Mobile Ad Hoc and Smart Systems (MASS). pp. 170–177. IEEE (2023)
22. Lombardi, S., Simon, T., Saragih, J., Schwartz, G., Lehrmann, A., Sheikh, Y.: Neural volumes: Learning dynamic renderable volumes from images. arXiv preprint arXiv:1906.07751 (2019)
23. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. Communications of the ACM **65**(1), 99–106 (2021)
24. Müller, T., Evans, A., Schied, C., Keller, A.: Instant neural graphics primitives with a multiresolution hash encoding. ACM transactions on graphics (TOG) **41**(4), 1–15 (2022)
25. Neff, T., Stadlbauer, P., Parger, M., Kurz, A., Mueller, J.H., Chaitanya, C.R.A., Kaplanyan, A., Steinberger, M.: Donerf: Towards real-time rendering of compact neural radiance fields using depth oracle networks. In: Computer Graphics Forum. vol. 40, pp. 45–59. Wiley Online Library (2021)
26. Newcombe, R.A., Fox, D., Seitz, S.M.: Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 343–352 (2015)
27. Park, J.J., Florence, P., Straub, J., Newcombe, R., Lovegrove, S.: Deepsdf: Learning continuous signed distance functions for shape representation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 165–174 (2019)
28. Park, K., Sinha, U., Barron, J.T., Bouaziz, S., Goldman, D.B., Seitz, S.M., Martin-Brualla, R.: Nerfies: Deformable neural radiance fields. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5865–5874 (2021)
29. Peng, Y., Yan, Y., Liu, S., Cheng, Y., Guan, S., Pan, B., Zhai, G., Yang, X.: Cagenerf: Cage-based neural radiance field for generalized 3d deformation and animation. In: Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., Oh, A. (eds.) Advances in Neural Information Processing Systems. vol. 35, pp. 31402–31415. Curran Associates, Inc. (2022)
30. Pumarola, A., Corona, E., Pons-Moll, G., Moreno-Noguer, F.: D-nerf: Neural radiance fields for dynamic scenes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10318–10327 (2021)
31. Schonberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4104–4113 (2016)
32. Song, L., Chen, A., Li, Z., Chen, Z., Chen, L., Yuan, J., Xu, Y., Geiger, A.: Nerfplayer: A streamable dynamic scene representation with decomposed neural radiance fields. IEEE Transactions on Visualization and Computer Graphics **29**(5), 2732–2742 (2023)

33. Srinivasan, P.P., Tucker, R., Barron, J.T., Ramamoorthi, R., Ng, R., Snavely, N.: Pushing the boundaries of view extrapolation with multiplane images. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 175–184 (2019)
34. Tancik, M., Casser, V., Yan, X., Pradhan, S., Mildenhall, B., Srinivasan, P.P., Barron, J.T., Kretzschmar, H.: Block-nerf: Scalable large scene neural view synthesis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8248–8258 (2022)
35. Tancik, M., Weber, E., Ng, E., Li, R., Yi, B., Kerr, J., Wang, T., Kristoffersen, A., Austin, J., Salahi, K., Ahuja, A., McAllister, D., Kanazawa, A.: Nerfstudio: A modular framework for neural radiance field development. In: ACM SIGGRAPH 2023 Conference Proceedings. SIGGRAPH '23 (2023)
36. Tang, J., Zhou, H., Chen, X., Hu, T., Ding, E., Wang, J., Zeng, G.: Delicate textured mesh recovery from nerf via adaptive surface refinement. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 17739–17749 (2023)
37. Wang, X., Zhu, J., Ye, Q., Huo, Y., Ran, Y., Zhong, Z., Chen, J.: Seal-3d: Interactive pixel-level editing for neural radiance fields (2023), https://arxiv.org/abs/2307.15131
38. Wu, X., Xu, J., Zhang, X., Bao, H., Huang, Q., Shen, Y., Tompkin, J., Xu, W.: Scanerf: Scalable bundle-adjusting neural radiance fields for large-scale scene rendering. ACM Transactions on Graphics (TOG) **42**(6), 1–18 (2023)
39. Wu, Z., Liu, T., Luo, L., Zhong, Z., Chen, J., Xiao, H., Hou, C., Lou, H., Chen, Y., Yang, R., et al.: Mars: An instance-aware, modular and realistic simulator for autonomous driving. In: CAAI International Conference on Artificial Intelligence. pp. 3–15. Springer (2023)
40. Xu, T., Harada, T.: Deforming radiance fields with cages (2022), https://arxiv.org/abs/2207.12298
41. Yang, L., Yu, K., Tang, T., Li, J., Yuan, K., Wang, L., Zhang, X., Chen, P.: Bevheight: A robust framework for vision-based roadside 3d object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 21611–21620 (2023)
42. Yang, Z., Sun, Y., Liu, S., Jia, J.: 3dssd: Point-based 3d single stage object detector. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11040–11048 (2020)
43. Yu, A., Li, R., Tancik, M., Li, H., Ng, R., Kanazawa, A.: Plenoctrees for real-time rendering of neural radiance fields. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5752–5761 (2021)