2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

ν-DBA: Neural Implicit Dense Bundle Adjustment Enables Image-Only Driving Scene Reconstruction

Yunxuan Mao[1], Bingqi Shen[1], Yifei Yang[1], Kai Wang[2], Rong Xiong[1], Yiyi Liao[1], and Yue Wang[1*]

***Abstract*— The joint optimization of the sensor trajectory and 3D map is a crucial characteristic of bundle adjustment (BA), essential for autonomous driving. This paper presents ν-DBA, a novel framework implementing geometric dense bundle adjustment (DBA) using 3D neural implicit surfaces for map parametrization, which optimizes both the map surface and trajectory poses using geometric error guided by dense optical flow prediction. Additionally, we fine-tune the optical flow model with per-scene self-supervision to further improve the quality of the dense mapping. Our experimental results on multiple driving scene datasets demonstrate that our method achieves superior trajectory optimization and dense reconstruction accuracy. We also investigate the influences of photometric error and different neural geometric priors on the performance of surface reconstruction and novel view synthesis. Our method stands as a significant step towards leveraging neural implicit representations in dense bundle adjustment for more accurate trajectories and detailed environmental mapping.**

I. Introduction

Building a dense map of the driving scene is desirable for autonomous vehicles, aiding localization, planning, and simulation. Bundle adjustment (BA) is indispensable for this reconstruction task when using images with noisy trajectories, which jointly optimizes camera poses and map points. However, the sparse images and small parallax angles along the driving route make dense surface mapping challenging using only images and noisy trajectories in driving scenes.

To investigate this problem, we delve into BA from the perspectives of *map parametrization* and *measurement metric*, respectively. The most popular BA methods take sparse 3D scene points cloud as the map parametrization, as shown in Fig. 1 (a). This representation is consistent across different views, yet falls short in describing the dense geometry. Nevertheless, these sparse BA methods provide insights into the measurement metric. There are two types of measurement metrics in sparse BA: geometric error and photometric error. COLMAP [1] and photometric BA [2] are two representative works, respectively. Geometric BA utilizes the scene point correspondence across images for optimization, while photometric BA utilizes the local patch consistency of the scene point projections. In [3], geometric BA is shown to be more accurate than photometric BA, which might be induced by sensitivity in illumination and calibration.

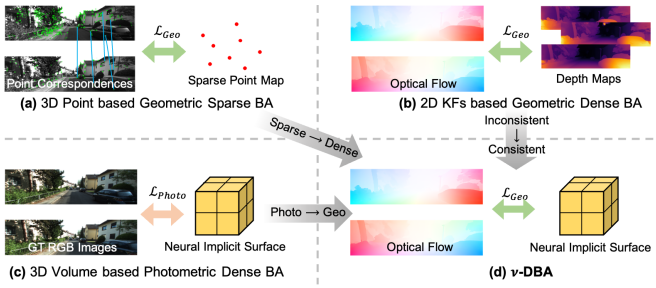


Fig. 1: **Comparison of BA methods.** (a) Traditional geometric sparse BA with 3D point map. (b) Geometric dense BA with 2D keyframes. (c) Photometric dense BA with the neural implicit surface. (d) **ν-DBA**: geometric dense BA with the neural implicit surface.

In recent years, dense BA has received increasing attention. A representative work, DROID-SLAM [4], parametrizes the dense map based on a set of 2D keyframes and their optimizable depth maps, as shown in Fig. 1 (b). A dense map can be obtained by projecting the depth maps to 3D. However, keyframe depth might be redundant in 3D, and there are no 3D consistency constraints to enforce the overlap part to be the same, which may limit the accuracy of the resultant map. Despite the limitation in the map parameterization, DROID-SLAM inherits the advantage of geometric BA, as it can be essentially considered as geometric BA by employing dense optical flow using RAFT [5] as the measurement. This makes DROID-SLAM a geometric dense BA (DBA), as the scene points correspondence in geometric sparse BA is akin to sparse optical flow.

Recent advances in neural implicit representation have achieved impressive high-fidelity novel view synthesis [6] and scene reconstruction [7], [8]. With the differentiable rendering, the image regression loss can be back-propagated to learn the 3D representation [7], [8]. Considering such neural mapping from the perspective of dense BA, we find that the neural implicit representation acts as a 3D dense map parametrization, avoiding the consistency issue of keyframe depth maps. The differentiable rendering can be viewed as a photometric error-based measurement model, yielding a photometric dense BA, as shown in Fig. 1 (c). Utilizing such a photometric dense BA as the back-end, several neural implicit representation-based SLAM systems

*This work was supported by the National Nature Science Foundation of China under Grant 62373322, Zhejiang Provincial Natural Science Foundation of China under Grant No. LD24F030001, and the Innovation and Development Special Fund of the Hangzhou Chengxi Sci-tech Innovation Corridor.

[1]Yunxuan Mao, Bingqi Shen, Yifei Yang, Yiyi Liao, Rong Xiong, and Yue Wang are with Zhejiang University, Hangzhou, China.

[2]Kai Wang is with the Application Innovate Lab, Huawei Incorporated Company, Beijing, China.

*Corresponding author.

are proposed [9], [10], [11], [12], [13]. Despite some of them are applicable to RGB images for indoor scenes [10], [13], existing methods in this direction struggle in street scenes with RGB input, indicating that street scenes pose challenges due to complex illumination and textureless regions. This raises a question: *Can we bridge the 3D neural implicit representation with the geometric error to improve the dense BA accuracy?*

Following the idea, in this paper, we propose ν-DBA (pronounced as "neuDBA"), a geometric dense BA framework with 3D neural implicit surface as map parametrization, which optimizes both the map surface and trajectory poses using geometric error derived from dense flow, as illustrated in Fig. 1 (d). Apart from the framework, we investigate the effectiveness of photometric error and other neural geometric prior on the quality of the surface reconstruction and the novel view synthesis. Moreover, to relieve the generalization gap of the flow prior, we further propose to fine-tune the flow model through per-scene self-supervision. In the experiments, we evaluate our method on several driving scene datasets and demonstrate superior trajectory and mapping accuracy. In summary, we make the following contributions:

- We propose a geometric error-based dense BA framework with 3D consistent representation i.e. neural implicit surface as the map parametrization, for driving scene reconstruction.
- We propose to self-supervise the optical flow model to narrow the generalization gap, which further improves the quality of the reconstructed surface.
- We evaluate the framework on several driving scene datasets, demonstrating superior performance in reconstruction and trajectory optimization, and investigating the effect of photometric error and other neural geometric priors.

II. Related Works

A. Bundle Adjustment

A comprehensive and precise road map is key for the effective operation of autonomous vehicles, facilitating their capabilities in self-localization, trajectory planning, and simulation exercises. Bundle Adjustment (BA) plays a critical role in this mapping process, especially when dealing with imprecise trajectories and imagery, by simultaneously adjusting the estimations of camera orientations and landmark locations. We conduct an in-depth exploration of Bundle Adjustment (BA) with respect to the dimensions of map parameterization and measurement metrics. The measurement metrics of existing mature BA include two main types: geometric error and photometric error. Methods that use geometric error perform optimization in two steps. First, the raw sensor measurements are pre-processed to generate an intermediate representation, solving part of the overall problem, such as computing the image coordinates of corresponding points. Second, the computed intermediate values are used to estimate the geometry and sensor poses. Photometric error-based methods skip the pre-computation step, directly using the actual sensor data for optimization.

Photometric error-based methods, such as DSO [14], LSD-SLAM [15], and DTAM [16], rely on minimizing the photometric error between consecutive frames to estimate the camera pose and reconstruct the scene structure. These methods suffer from challenges like illumination changes and geometric distortions, such as rolling shutter artifacts, which can adversely affect their performance. In contrast, geometric error-based methods, such as MonoSLAM [17], PTAM [18], and ORB-SLAM3 [3], solve the problems of direct methods by generating features and estimating a geometry prior before bundle adjustment. However, in early works, both methods utilize sparse 3D feature point cloud as the map parametrization due to the sparse feature correspondence across frames. The sparse bundle adjustment is limited when lacking features.

With the development of hardware, dense mapping has received increasing attention. DROID-SLAM [4] proposed a geometric dense BA with dense optical flow. DROID-SLAM builds a dense 3D map by projecting the estimated depth of 2D keyframes to 3D. might be redundant in 3D, and there are no 3D consistency constraints. In this paper, we aim to bridge the 3D neural implicit representation with the geometric error to improve the dense BA.

B. Neural Implicit Scene Representation

Neural implicit representations have gained popularity in 3D reconstruction [19], [20], [21], [22], [23], [24], [25]. As dense, consistent scene representations, these methods can extract high-fidelity surfaces useful for autonomous driving tasks. At the same time, neural radiance field (NeRF) [6] has achieved impressive novel view synthesis results with volume rendering techniques. The differentiable volume rendering techniques make it possible to optimize neural implicit surface representation with 2D information [7], [26]. From the dense BA perspective, we find that the neural implicit representation acts as a 3D map parametrization, and the differentiable rendering acts as the measurement model.

Recently, building upon such photometric dense BA, several neural implicit representation-based SLAM systems have been proposed [9], [27], [28], [10], [13], [12]. These works enable dense BA with neural implicit representation. However, most of them focus primarily on indoor scenes with dense viewpoints and utilize the ground truth depth image for optimization tasks. Yet, ground truth geometric priors, such as depth images or LiDAR, are not available for all driving scenes. NICER-SLAM [10] proposes a ground truth geometric prior-free dense SLAM system for indoor scenes but does not extend to outdoor unbounded scenes with driving views. The viewpoints are forward-facing with long and narrow trajectories in the outdoor unbounded driving scenes, which differ significantly from indoor scenes. StreetSurf [29] proposes a multi-shell cuboid neural scene representation for street driving views. However, the monocular geometry cues used in StreetSurf cannot be self-supervised and are therefore constrained by the training dataset. By contrast, we propose a geometric dense BA framework with a 3D neural implicit surface as the map parametrization for RGB input in outdoor

driving scenes and fine-tune the optical flow model for each scene through self-supervision to enhance performance.

III. ν-Dense Bundle Adjustment

Given a sequence of RGB images $\mathbf{I} = \{I_0, I_1, \ldots, I_k\}$ with noisy camera poses $\mathbf{T} = \{T_1, T_2, \ldots, T_k\}$, the proposed ν-DBA is capable of simultaneously extracting dense and complete surfaces and optimizing the camera poses. As shown in Fig. 2, there are two core components: the close-range SDF field as map parameterization, and the flow loss as the geometric metric. In addition, a self-supervision method is employed to narrow down the generalization gap of off-the-shelf optical flow predictor.

A. Close-Range SDF Field

The close-range model represents geometry as a signed distance function (SDF). Given a 3D point, it returns the point's distance to the closest surface. In this work, the SDF function is parameterized by a 3D hash table h_θ as the feature grid, and a single MLP f_θ as the geometry decoder

$$\hat{s} = f_\theta(h_\theta(\mathbf{x})) \tag{1}$$

Here, $\mathbf{x}$ is the 3D point, $\hat{s}$ denotes the corresponding SDF value, and θ are learnable parameters.

B. Neural Geometric Prior with Self-Supervision

To build the geometric error, we choose optical flow estimated by a pre-trained model as the neural geometric prior i.e. observations. When a stereo pair is available, disparity is also employed.

1) Optical Flow: Optical flow finds the 2D-pixel displacement field between two images, which can be easily obtained using an off-the-shelf flow estimator. We use the pretrained unimatch model [30] to estimate the optical flow $\mathbf{F}_i$ for each input frame pair $\{I_i, I_{i+1}\}$. Compared with monocular depth, the optical flow infers the correspondences, which is not affected by the metric scale.

2) Stereo Matching: Stereo matching can be regarded as correspondence between the left and right cameras, of which the relative pose is known, thus reducing the correspondence space to epipolar lines. The unimatch model also estimates disparity D_{disp} for a given stereo pair I_i^l, I_i^r. The accuracy of stereo matching is often higher than that of optical flow, but it is applicable only in scenes where a stereo pair is available.

3) Self-supervised Fine-tune: One benefit of flow is that the model can be self-supervised using only RGB image inputs, which is able to narrow down the generalization gap of the optical flow inference model. Therefore, we follow [26] to fine-tune the inference result of the optical flow model in a per-scene manner. Similar to [31], we apply census loss, smoothness loss, and self-supervision loss to the optical flow model.

4) Other Neural Geometric Priors: Following [8], [29], the proposed model can also be supervised by the monocular geometry cues i.e. monocular depth and normal. Note that compared with flow, the monocular depth and normal can only be self-supervised when camera poses are given, which may limit the generalization performance of monocular cues. In addition, we consider that noise in flow prediction is more uniform in near and far regions than that in monocular depth and normal prediction, which makes flow more aligned with the uniform weighting in loss terms.

C. Optimization

The ν-DBA optimizes the SDF field and the camera poses together with the supervision of 2D neural geometric prior estimated by the model in Section III-B.

1) Volume Rendering: Following recent work [6], we employ volume rendering as a measurement model, to learn the pose and map supervised by estimated optical flow. Given the camera's intrinsic parameters and current camera pose, we cast a ray $\mathbf{r}$ from the camera center $\mathbf{o}$ through the pixel along its normalized view direction $\mathbf{v}$. We sample N points along a ray, denoted as $\mathbf{x}_i = \mathbf{o} + d_i\mathbf{v}$, where d_i is the depth of point $\mathbf{x}_i$, and $i \in 1, \cdots, N$.

Given an image pair $\{I_k, I_j\}$ and a ray $\mathbf{r}_j$ from the pixel $\mathbf{p}_j$ in image I_j, we can calculate the pixel location of every sample $\mathbf{x}_i^{\mathbf{r}_j}$ in the ray $\mathbf{r}_j$ on another image via differentiable tomography as in [32]

$$H(d) = K_k R_k \left(I - \frac{(R_k^{-1}\mathbf{t}_k - R_j^{-1}\mathbf{t}_j)\mathbf{n}_j^\top R_j}{d} \right) R_j^{-1} K_j^{-1} \tag{2}$$

that $\{K_j, R_j, \mathbf{t}_j\}$ are the camera intrinsics, rotations, and translations of frames $\{I_j\}$, $\mathbf{n}$ is the principle axis of the reference camera, and d is the depth of the point in reference view I_j. Then, we can get the pixel location of sample $\mathbf{x}_i^{\mathbf{r}_j}$ on image I_k: $\mathbf{p}_k^i \sim H(d_i) \cdot \mathbf{p}_j$ where '$\sim$' denotes the projective equality.

For volume rendering, following NeuS [7], we first query the SDF value $\hat{s}_i$ of sample $\mathbf{x}_i$ and convert the SDF values into density values σ_i. The optical flow $\hat{\mathbf{F}}$ of pixel $\mathbf{p}_j$ from frame I_j to frame I_k can be calculated as

$$\hat{\mathbf{F}}(\mathbf{r}_j) = \sum_{i=1}^{N} T_i \alpha_i \mathbf{p}_k^i - \mathbf{p}_j \tag{3}$$

where

$$T_i = \prod_{j=1}^{i} (1 - \alpha_j) \text{ and } \alpha_i = 1 - \exp(-\sigma_i \delta_i) \tag{4}$$

Similarly to optical flow, we can render the disparity when given a stereo pair. We can calculate the disparity of a sample ray $\mathbf{r}$

$$\hat{D}_{disp}(\mathbf{r}) = \sum_{i=1}^{N} T_i \alpha_i \frac{f \cdot b}{d_i} \tag{5}$$

where f is the focal length of the camera and b is the baseline of the stereo pair.

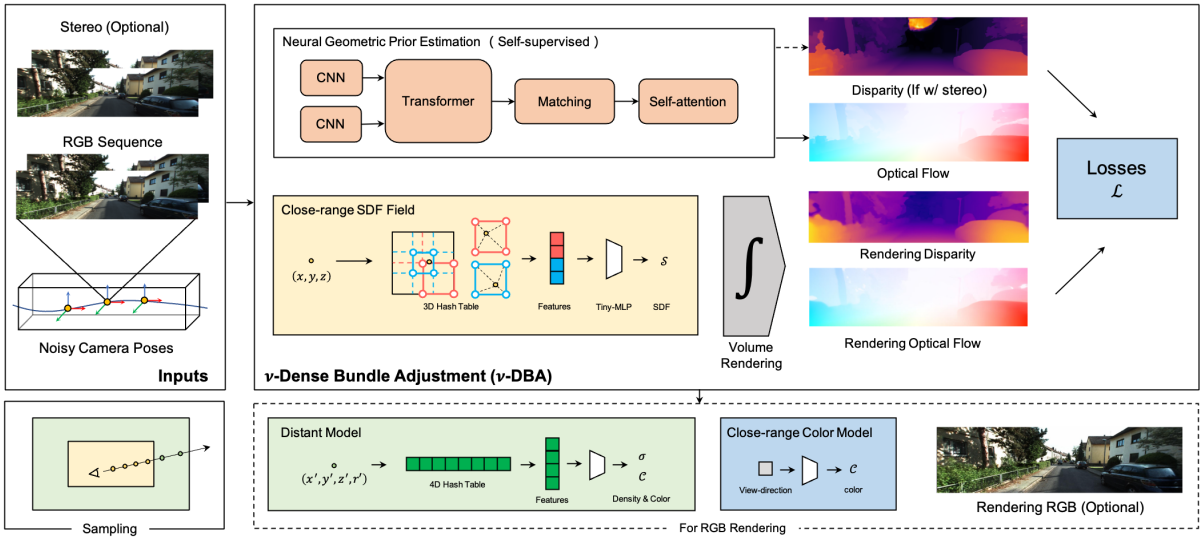


Fig. 2: **Overview.** Our method takes a sequence of RGB images with noisy camera poses as input. In ν-DBA, the self-supervised model inference optical flow and disparity as neural geometric priors, which is utilized to supervise the close-range SDF field and the camera poses with geometric losses. Moreover, a distant model and a close-range color model are used for RGB rendering.

2) Loss Function: To perform ν-DBA, we define the geometric error loss function over optical flow and disparity

$$\mathcal{L}_f = \sum_{\mathbf{r}\in R} \|\hat{\mathbf{F}}(\mathbf{r}) - \mathbf{F}(\mathbf{r})\|_2, \ \mathcal{L}_d = \sum_{\mathbf{r}\in R} \|\hat{D}_{disp}(\mathbf{r}) - D_{disp}(\mathbf{r})\|_2 \tag{6}$$

Following common practice, we also add an Eikonal term on the sampled points to regularize SDF values in 3D space and sparsity loss to penalize the uncontrollable free surfaces:

$$\mathcal{L}_{eik} = \sum_{\mathbf{x}\in\mathcal{X}} (\|\nabla \mathcal{S}_\theta(\mathbf{x})\|_2 - 1)^2 \tag{7}$$

$$\mathcal{L}_{spa} = \sum_{\mathbf{x}\in\mathcal{X}} \exp(-\tau \cdot \|\mathcal{S}_\theta(\mathbf{x})\|) \tag{8}$$

where τ is a hyperparameter to rescale the SDF value.

To eliminate ambiguous occupancy of the close-range SDF field, an entropy regularization term is added

$$\mathcal{L}_{ent} = f_{ent}(\hat{O}_c(\mathbf{r})), \ f_{ent}(x) = -(x \ln x + (1-x)\ln(1-x)) \tag{9}$$

where $\hat{O}(\mathbf{r})$ is the opacity of close-range SDF field along ray $\mathbf{r}$ that can be calculated as $\hat{O}(\mathbf{r}) = \sum_{i=1}^{N} T_i \alpha_i$.

The overall loss used for ν-DBA is

$$\mathcal{L}_{\nu DBA} = \mathcal{L}_f + \lambda_1 \mathcal{L}_d + \lambda_2 \mathcal{L}_{eik} + \lambda_3 \mathcal{L}_{spa} + \lambda_4 \mathcal{L}_{ent} \tag{10}$$

D. Road Surface Initialization

As discussed in [29], the disentanglement of close-range and distant view is an unsupervised and ill-posed problem with almost no constraints. The road surface initialization, wherein the close-range SDF field is pre-trained to possess a zero-level set approximately aligned with the road surface, is crucial for addressing this issue in driving scenes. To get the road surface, we leverage the corresponding information provided by the estimated optical flow. The corresponding points are triangulated with the preprocessed camera poses to obtain a 3D point cloud. Subsequently, we identify the plane with the maximal support within the point cloud by employing the Random Sample Consensus (RANSAC) algorithm. This best-fitting plane serves as the initial reference for the close-range SDF model.

E. Voxel-based Sampling

The sampling strategy is important for neural implicit field training. Recent works [33], [12], [26], [29] have proved that the octree structure can significantly improve sampling efficiency. As a result, we employ the octree to sample points along the rays in the close-range model. The octree is first initialized by the triangulated point cloud in subsection III-D and is updated iteratively throughout the training of the close-range model. We uniformly sample N points each voxel intersected by the ray. To further improve the sampling efficiency, we discard those sampling points whose SDF value is higher than the threshold after 60% of the total number of training iterations. These sampling strategies ensure that the selected points are in proximity to the surface, thereby enhancing the model's accuracy.

F. Color Rendering

In this work, we consider color rendering optional as our geometric error provides sufficient guidance for the DBA task. As discussed in [9], [10], when applying color-related loss and geometric-related loss together, the loss terms may conflict, degenerating the performance. Therefore, we decouple the color model from the geometry model and

only learn the color model if novel view synthesis is required in addition.

In the color model, we employ a shader for close-range view and a distant model that represents the scene hundreds or even thousands of meters away from the camera as well as the sky.

1) Close-Range Color Field: Given point $\mathbf{x}$ and a view direction $\mathbf{v}$, the function of RGB color $\mathcal{C}$ can be defined as

$$\hat{\mathbf{c}} = \mathcal{C}_\theta(\mathbf{v}, \hat{\mathbf{z}}) \tag{11}$$

The feature vector $\hat{\mathbf{z}}$ is the output of the geometry decoder of the SDF model.

2) Distant Model: To handle unbounded scene rendering, we leverage a scene parameterization similar to [29]. The distant model is a hyper 4D hash table. Given the 4D input $\mathbf{x}'$, the network directly outputs the density $\hat{\sigma}$ and RGB color $\hat{\mathbf{c}}$ of the sample point.

$$[\hat{\sigma}, \hat{\mathbf{c}}] = h_\theta^d(\mathbf{x}') \tag{12}$$

To form a 4D input, we sample on cuboid shells and apply inverse cuboid warping. Samples of the distant-view model lie on cuboid shells. The cuboid shells are scaled proportionally from the cuboid shell of the close-range space with inverse-proportionally increasing scales. The warped point $\mathbf{x}'$ of the sample $\mathbf{x}$ on the ith cuboid shell can be expressed as

$$\mathbf{x}' = [r_i \cdot \mathbf{x}, r_i], \; r_i = \frac{1}{(1 - i/n) + (i/n)(1/r_{max})} \tag{13}$$

where n is the number of cuboid shells and r_{max} is the scale of the largest cuboid shell relative to the close-range shell.

3) Optimization: Given a sample x_i, the color model outputs the RGB color value $\mathbf{c}$. Similar to the section. III-C.1, the color $\hat{\mathbf{C}}$ for the given ray $\mathbf{r}$ is computed via numerical integration

$$\hat{\mathbf{C}}(\mathbf{r}) = \sum_{i=1}^{N} T_i \alpha_i \mathbf{c}_i \tag{14}$$

Then, we can optimize the color model with a simple photometric loss

$$\mathcal{L}_{photo} = \sum_{\mathbf{r} \in R} \|\hat{\mathbf{C}}(\mathbf{r}) - \mathbf{C}(\mathbf{r})\|_2 \tag{15}$$

IV. Experiments

In this section, we validate our method on various real-world outdoor datasets. We perform ablation study wrt. different geometry cues and provide qualitative and quantitative comparisons against state-of-the-art baselines and perform ablation study wrt. different geometry cues.

A. Experiment Setup

1) Datasets: In this work, we mainly focus on outdoor unbounded scenes with monocular input or stereo pair input. Thus we consider 3 real-world datasets for evaluation: KITTI-360 [34], Waymo [35], and a self-collected dataset from an autonomous driving mine truck with more unstructured views.

		Acc. ↓	Comp. ↓	Comp. Ratio ↑
KITTI-360	w/o ss	40.92	25.38	73.42
	w/ ss	**38.40**	**23.82**	**74.78**
Waymo	w/o ss	48.06	81.95	63.87
	w/ ss	**45.98**	**72.89**	**73.96**

TABLE I: **Ablation Study on Self-Supervised (ss) Fine-tuning.**

2) Baselines: We compare our method to 1) state-of-the-art neural implicit surfaces methods: NeuS-facto [36] and StreetSurf [29]. 2) neural radiance field methods: NeRFacto [36] and F2-NeRF [37]. 3) state-of-the-art SLAM system: ORB-SLAM3 [3] and DROID-SLAM [4].

3) Metrics: To evaluate the reconstruction accuracy, we report Accuracy [cm], Completion [cm], and Completion Ratio with a threshold of 20cm of the extracted meshes compared with ground truth LiDAR data. For a fair comparison, we remove unseen regions that are not inside any camera's viewing frustum and crop the meshes with the oriented bounding box of LiDAR data. For appearance, we evaluate the PSNR, SSIM, and LPIPS for rendered color images averaging on the training set and test set the same as [29]. To evaluate the tracking accuracy, we use ATE [m].

4) Implementation Details: We implement our method in PyTorch [38]. We use the cuboid hash-grids in nr3d [29] as the hash table in scene representation. We sample 8192 rays per iteration and train our model for 20k iterations for about 30 minutes on a single NVIDIA RTX 4090 GPU.

B. Ablation Study

We verify whether the self-supervised fine-tuning of the optical flow model can improve the performance of the method and ablate the impact of different geometric cues on reconstruction quality. To avoid interference caused by incorrect pose, the comparison is conducted using ground truth trajectories. The data is averaged over four different scenes in the KITTI-360 dataset [34].

1) Self-Supervised Fine-tuning: We compare the reconstruction quality before and after self-supervised fine-tuning and report metrics averaged over the KITTI-360 and Waymo datasets in TABLE I. Thanks to the narrower generalization gap, we observe that self-supervised fine-tuning improves the reconstruction accuracy. It is worth noting that the pre-trained models without (w/o) fine-tuning can also achieve relatively good results, but as shown in Fig. 3, self-supervised fine-tuning can improve the details of the model and achieve better reconstruction.

2) Different Geometry Cues: We now investigate the effectiveness of different geometric cues. To avoid interference from RGB cues, we do not use photometric loss in this experiment. We conducted monocular and stereo-pair experiments on the KITTI-360 dataset. Note that we also ablate monocular depth and normal cues with our model, and the loss function is the same as [8]. As shown in TABLE II, the results indicate that for monocular input, self-supervised flow outperforms monocular cues as expected. For stereo

(a) Ours w/o ss (b) Ours (c) GT

Fig. 3: **Ablation study on Self-supervised Fine-tuning.** Self-supervised (ss) fine-tuning improves the details of the model and achieves a better reconstruction.

Geometry cues	Acc. ↓	Comp. ↓	Comp. Ratio ↑
m-depth	82.71	32.80	61.73
m-depth+m-normal	63.45	27.92	64.80
flow	36.79	25.65	71.14
ss-flow	31.21	23.93	74.44
stereo	27.17	20.27	78.71
stereo+ss-flow	**23.95**	**19.96**	**80.34**

TABLE II: **Ablation Study on Different Geometry Cues.** `ss-flow` denotes self-supervised optical flow, `m-` denotes monocular geometry cues.

		Acc. ↓	Comp. ↓	Comp. Ratio ↑
ss-flow	w/ $\mathcal{L}_p$	38.40	**23.82**	**74.78**
	w/o $\mathcal{L}_p$	**31.21**	23.93	74.44
stereo+ss-flow	w/ $\mathcal{L}_p$	26.42	**19.67**	**80.53**
	w/o $\mathcal{L}_p$	**23.95**	19.96	80.34

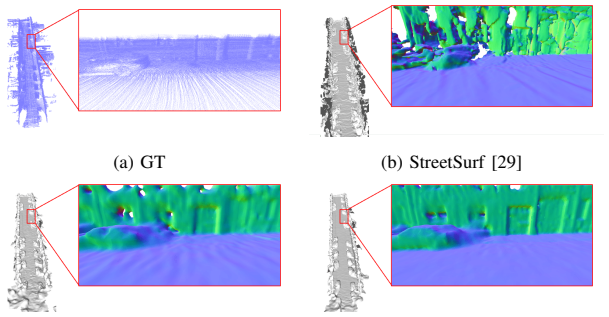
TABLE III: **Ablation Study on Photometric Loss.**

pair input, optical flow further helps stereo improve the reconstruction performance. Comparing monocular and stereo pair inputs, stereo pair provides better geometry, which is obvious with more information.

3) Photometric Loss: In this experiment, we investigated the impact of photometric loss on reconstruction. As shown in TABLE III, photometric loss increases completeness at the cost of a significant decrease in accuracy. This result derives the conflict and ambiguity mentioned in [9], [10]. We consider that the photometric loss hurts the geometry in the way of more parameters to learn, and illumination disturbance to color loss as found in sparse bundle adjustment literature [3].

C. Dense Bundle Adjustment

1) Trajectory and Mapping: In this experiment, we demonstrate the effectiveness of our method as a dense geometric bundle adjustment with noisy camera poses. The results are averaged over four different scenes in the KITTI-360 dataset. We compare our method with the back-end of DROID-SLAM and StreetSurf. The input camera poses for all methods are obtained from the front-end of DROID-SLAM. ORB-SLAM3 is also utilized as a reference for tracking comparison. As shown in Table IV, we compare the accuracy of the trajectory as well as that of the resultant surface. All other methods outperform ORB-SLAM3, illustrating that dense geometric bundle adjustment is more effective than traditional sparse geometric bundle adjustment. DROID-SLAM is not as effective as the other two methods based on neural implicit surfaces, indicating the effectiveness



(a) GT (b) StreetSurf [29]

(c) **Ours** (d) **Ours (w/ stereo)**

Fig. 4: **Reconstruction results on the KITTI-360 dataset.** Since no ground truth mesh is given, the ground truth LiDAR point cloud is displayed.

Method	ATE ↓	Acc. ↓	Comp. ↓	Comp. Ratio ↑
ORB-SLAM3 [3]	0.186	-	-	-
DROID-SLAM [4]	0.084	87.01	81.31	40.07
StreetSurf [29]	0.078	75.52	38.63	56.51
Ours	0.073	41.63	30.62	62.48
Ours (w/ stereo)	**0.071**	**29.77**	**25.35**	**73.40**

TABLE IV: **Quantitative comparison of bundle adjustment results.**

of the 3D consistency provided by neural implicit representation. Compared with StreetSurf, our method employs a purely geometric BA without a photometric loss. Our method achieves the best performance in both tasks by eliminating color task disturbance and utilizing a 3D-consistent map parametrization. For qualitative comparison, as shown in Fig. 4, our method reconstructs more accurate and detailed surfaces.

2) Trajectory and Colored Mapping: In the previous subsection, we evaluate the results of ν-DBA: camera tracking accuracy and reconstruction quality. In addition to ν-DBA, our method, as a neural implicit scene representation, can render high-quality color images.

(a) F2-NeRF [37] (b) StreetSurf [29]

(c) **Ours** (d) GT

Fig. 5: **Rendering results on the KITTI-360 dataset.**

	Appearance			Geometry		
Method	PSNR ↑	SSIM ↑	LPIPS ↓	Acc. ↓	Comp. ↓	Comp. Ratio ↑
NeRFacto [36]	17.02	0.593	0.412	-	-	-
F2-NeRF [37]	26.70	0.725	0.359	-	-	-
NeuS-Facto [36]	14.27	0.533	0.437	446.93	201.86	9.92
StreetSurf [29]	30.25	0.887	0.342	62.56	74.84	70.19
Ours (ss-flow)	**30.54**	**0.891**	**0.298**	**45.98**	**72.89**	**72.96**

TABLE V: **Quantitative comparison of appearance and geometry on Waymo dataset.**

Method	PSNR ↑	SSIM ↑	LPIPS ↓
NeRFacto [36]	16.89	0.580	0.421
F2-NeRF [37]	22.24	0.781	0.365
NeuS-Facto [36]	16.23	0.550	0.432
StreetSurf [29]	24.37	0.816	0.329
Ours	25.14	0.830	0.324
Ours w/ stereo	**25.93**	**0.840**	**0.316**

TABLE VI: **Quantitative comparison of appearance on KITTI-360 dataset.**

Method	PSNR ↑	SSIM ↑	LPIPS ↓
StreetSurf [29]	26.92	0.766	0.389
Ours	**27.33**	**0.823**	**0.385**

TABLE VII: **Quantitative comparison of appearance on Mine dataset.**

Continuing with the above experiment, we compare the two neural implicit-based methods, StreetSurf and our method, with other neural implicit scene representation methods: NeRFacto [36], F2-NeRF [37], and NeuS-Facto [36]. For a fair comparison, the camera pose inputs for these three methods are also obtained from the front-end of DROID-SLAM, which is the same as Section IV-C. As shown in TABLE VI, both our methods with only self-supervised optical flow prior and with additional stereo pair outperform other methods. The qualitative experimental results are shown in Fig. 5, where our method achieves a sharper reconstruction than other methods.

D. Colored Mapping-only Bundle Adjustment

The above experiment utilizes the noisy camera pose input, which means the accuracy of camera poses after BA will influence the experimental results. To eliminate interference caused by incorrect pose, we conduct a reconstruction experiment using ground truth camera pose input on the Waymo dataset. As shown in TABLE V, our method still performs better than other methods when the pose is accurate.

KITTI-360 and Waymo are both driving scene datasets in street view. To validate the generalization of our method, we utilize a monocular outdoor dataset on a more unstructured view self-collected from an autonomous driving mine truck. Since the dataset lacks ground truth geometry data and camera poses, we only quantitatively analyzed the rendering results in TABLE VII. The rendering results and reconstruction results are shown in Fig. 6. On this challenging dataset, our method rendered higher-fidelity RGB images and extracted more precise meshes with challenging obstacles.

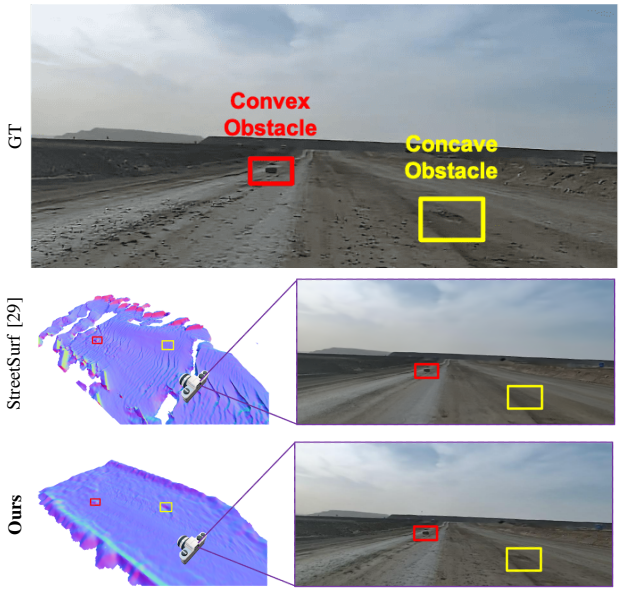


Fig. 6: **Rendering and reconstruction results on the Mine dataset.** The convex obstacle is about 30cm high and the concave obstacle is about 20cm deep.

All the experiment results above demonstrate that our method achieves better performance on camera tracking and reconstruction thanks to our ν-DBA and eliminating color task disturbance.

V. Conclusion

In this paper, we propose ν-DBA, a novel geometric dense BA framework that utilizes a 3D neural implicit surface representation as the map parametrization. This framework simultaneously optimizes the neural implicit map surface and the camera trajectory poses by minimizing geometric error derived from dense optical flow across consecutive frames, thereby bridging the 3D neural implicit representation with geometric error minimization to enhance the accuracy of dense bundle adjustment.

In addition, we investigate the effects of photometric error and other neural geometric prior on the accuracy of surface reconstruction and novel view synthesis. Moreover, we refine the flow model through per-scene self-supervision for better performance. Compared with other bundle adjustment methods and neural implicit reconstruction methods, our method achieves better performance in pose optimization and reconstruction.

REFERENCES

[1] J. L. Schönberger and J.-M. Frahm, "Structure-from-motion revisited," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[2] H. Alismail, B. Browning, and S. Lucey, "Photometric bundle adjustment for vision-based slam," in *Computer Vision–ACCV 2016: 13th Asian Conference on Computer Vision, Taipei, Taiwan, November 20-24, 2016, Revised Selected Papers, Part IV 13*. Springer, 2017, pp. 324–341.

[3] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. Montiel, and J. D. Tardós, "Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1874–1890, 2021.

[4] Z. Teed and J. Deng, "Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras," *Advances in neural information processing systems*, vol. 34, pp. 16 558–16 569, 2021.

[5] ——, "Raft: Recurrent all-pairs field transforms for optical flow," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*. Springer, 2020, pp. 402–419.

[6] 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," *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021.

[7] 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," *arXiv preprint arXiv:2106.10689*, 2021.

[8] Z. Yu, S. Peng, M. Niemeyer, T. Sattler, and A. Geiger, "Monosdf: Exploring monocular geometric cues for neural implicit surface reconstruction," *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[9] 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*, 2022, pp. 12 786–12 796.

[10] Z. Zhu, S. Peng, V. Larsson, Z. Cui, M. R. Oswald, A. Geiger, and M. Pollefeys, "Nicer-slam: Neural implicit scene encoding for rgb slam," *arXiv preprint arXiv:2302.03594*, 2023.

[11] M. M. Johari, C. Carta, and F. Fleuret, "Eslam: Efficient dense slam system based on hybrid representation of signed distance fields," *arXiv preprint arXiv:2211.11704*, 2022.

[12] Y. Mao, X. Yu, K. Wang, Y. Wang, R. Xiong, and Y. Liao, "Ngel-slam: Neural implicit representation-based global consistent low-latency slam system," *arXiv preprint arXiv:2311.09525*, 2023.

[13] Y. Zhang, F. Tosi, S. Mattoccia, and M. Poggi, "Go-slam: Global optimization for consistent 3d instant reconstruction," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2023.

[14] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE Transactions on Pattern Analysis Machine Intelligence*, pp. 1–1, 2016.

[15] J. Engel, T. Schöps, and D. Cremers, "Lsd-slam: Large-scale direct monocular slam," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part II 13*. Springer, 2014, pp. 834–849.

[16] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, "Dtam: Dense tracking and mapping in real-time," in *2011 international conference on computer vision*. IEEE, 2011, pp. 2320–2327.

[17] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "Monoslam: Real-time single camera slam," *IEEE transactions on pattern analysis and machine intelligence*, vol. 29, no. 6, pp. 1052–1067, 2007.

[18] G. Klein and D. Murray, "Parallel tracking and mapping for small ar workspaces," in *2007 6th IEEE and ACM international symposium on mixed and augmented reality*. IEEE, 2007, pp. 225–234.

[19] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove, "Deepsdf: Learning continuous signed distance functions for shape representation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 165–174.

[20] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger, "Occupancy networks: Learning 3d reconstruction in function space," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 4460–4470.

[21] V. Sitzmann, J. Martel, A. Bergman, D. Lindell, and G. Wetzstein, "Implicit neural representations with periodic activation functions," *Advances in Neural Information Processing Systems*, vol. 33, pp. 7462–7473, 2020.

[22] M. Oechsle, S. Peng, and A. Geiger, "Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 5589–5599.

[23] A. Chen, Z. Xu, F. Zhao, X. Zhang, F. Xiang, J. Yu, and H. Su, "Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 14 124–14 133.

[24] J. Wang, T. Bleja, and L. Agapito, "Go-surf: Neural feature grid optimization for fast, high-fidelity rgb-d surface reconstruction," *arXiv preprint arXiv:2206.14735*, 2022.

[25] X. Yu, Y. Liu, S. Mao, S. Zhou, R. Xiong, Y. Liao, and Y. Wang, "Nf-atlas: Multi-volume neural feature fields for large scale lidar mapping," *arXiv preprint arXiv:2304.04624*, 2023.

[26] X. Long, C. Lin, P. Wang, T. Komura, and W. Wang, "Sparseneus: Fast generalizable neural surface reconstruction from sparse views," in *European Conference on Computer Vision*. Springer, 2022, pp. 210–227.

[27] E. Sucar, S. Liu, J. Ortiz, and A. J. Davison, "imap: Implicit mapping and positioning in real-time," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 6229–6238.

[28] H. Wang, J. Wang, and L. Agapito, "Co-slam: Joint coordinate and sparse parametric encodings for neural real-time slam," in *CVPR*, 2023.

[29] J. Guo, N. Deng, X. Li, Y. Bai, B. Shi, C. Wang, C. Ding, D. Wang, and Y. Li, "Streetsurf: Extending multi-view implicit surface reconstruction to street views," *arXiv preprint arXiv:2306.04988*, 2023.

[30] H. Xu, J. Zhang, J. Cai, H. Rezatofighi, F. Yu, D. Tao, and A. Geiger, "Unifying flow, stereo and depth estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[31] R. Jonschkowski, A. Stone, J. T. Barron, A. Gordon, K. Konolige, and A. Angelova, "What matters in unsupervised optical flow," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*. Springer, 2020, pp. 557–572.

[32] Y. Yao, Z. Luo, S. Li, T. Fang, and L. Quan, "Mvsnet: Depth inference for unstructured multi-view stereo," *European Conference on Computer Vision (ECCV)*, 2018.

[33] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *ACM Transactions on Graphics (ToG)*, vol. 41, no. 4, pp. 1–15, 2022.

[34] Y. Liao, J. Xie, and A. Geiger, "Kitti-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 3, pp. 3292–3310, 2022.

[35] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine, *et al.*, "Scalability in perception for autonomous driving: Waymo open dataset," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 2446–2454.

[36] 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," in *ACM SIGGRAPH 2023 Conference Proceedings*, ser. SIGGRAPH '23, 2023.

[37] P. Wang, Y. Liu, Z. Chen, L. Liu, Z. Liu, T. Komura, C. Theobalt, and W. Wang, "F2-nerf: Fast neural radiance field training with free camera trajectories," *CVPR*, 2023.

[38] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, pp. 8024–8035. [Online]. Available: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf