Supplementary Material for Neural Surface Reconstruction and Rendering for Lidar-Visual Systems

I. Supplementary Details of Methodology

A. Structure-aware sampling

The overall algorithm is shown in Alg. 1 and Fig. 1. Given any desired rendering direction $\boldsymbol{d}$, the algorithm starts from the camera origin (Alg. 1-1), and iteratively steps forward (Alg. 1-4) along the ray direction according to its signed distance value. The gradient of the signed distance field along the ray direction M is estimated using linear interpolation (Alg.1-8). A filtered gradient m is updated with a relaxation coefficient $\gamma = 0.7$ (Alg.1-9), which adaptively determines the next step size δ_i (Alg. 1-3) for efficiency. At every step, we ensure that the space between two adjacent steps is intersected (Alg. 1-6), wherein we take predicted scale β_i into account to avoid triggering false revert steps in unconverged NDF. To avoid poor local minima, we keep marching even behind surfaces (Alg.1-4), and until a ray's transmittance (Alg. 1-7) falls below a threshold $\epsilon_T = 0.001$. The filtered slope m is verified by sphere tracing, and smaller step sizes near surfaces yield a higher sampling frequency for more accurate slope estimations. Therefore, we apply the estimated filtered slope M in the SDF-to-density transition to avoid expensive analytical or numerical gradient calculations. We further utilize the visible-aware occupancy map to skip the free space for efficiency, as shown in Fig. 1's skip step.

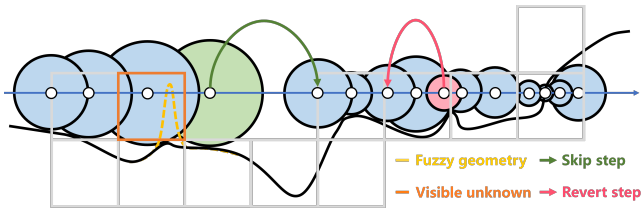


Fig. 1. Illustrations of visible-aware occupancy map and structure-aware sampling based on adaptive sphere tracing. Each circle has a radius equal to the SDF value at the sample point.

Algorithm 1: Structure-aware Sampling

Input: Neural distance field $f_{\mathcal{S}}$, point on rendering ray $\boldsymbol{x}(t) = \boldsymbol{o} + t\boldsymbol{d}$, termination conditions ϵ_T, relaxation coefficient $\gamma \in (0, 1)$.
Output: ray samples $\{t_i, m_i\}$.
Notation: Slope M, filtered slope m, transmittance T.

1 $t_i := 0,\ (s_i, \beta_i) := f_{\mathcal{S}}(\boldsymbol{x}(t_i)),\ m := -1,\ T := 1$
2 **while** $s_i > 0 \cup T > \epsilon_T$ **do**
3 $\quad \delta_i := |s_i| * \frac{2}{1-m}$ ▷ Adaptive step
4 $\quad t_{i+1} := t_i + \delta_i$
5 $\quad (s_{i+1}, \beta_{i+1}) := f_{\mathcal{S}}\left(\boldsymbol{x}\left(t_{i+1}\right)\right)$
6 $\quad$ **if** $\delta_i \leq |s_i| + |s_{i+1}| + 3\beta_i$ **then**
7 $\quad\quad T := T * \exp\left(-\sigma\left(s_i, \beta_i, m\right)\delta_i\right)$
8 $\quad\quad M := (s_{i+1} - s_i)/\delta_i$
9 $\quad\quad m := \gamma m + (1-\gamma)\, M$
10 $\quad\quad (t_i, m_i) := (t_{i+1}, m)$ ▷ Sample step
11 $\quad$ **end**
12 $\quad$ **else**
13 $\quad\quad m := -1$ ▷ Revert step
14 $\quad$ **end**
15 **end**

B. Losses

The Eikonal loss [1] and curvature loss [2] are further employed for both NeRF and NSDF samples to prevent the zero-everywhere and overfitting solutions for the SDF:

$$\begin{aligned} \mathcal{L}_{eik} &= \frac{1}{N_i}\sum_i \left(\|\nabla f_{\mathcal{S}}(\boldsymbol{x}_i)\|_2 - 1\right)^2, \\ \mathcal{L}_{curv} &= \frac{1}{N_i}\sum_i |\nabla^2 f_{\mathcal{S}}(\boldsymbol{x}_i)|, \end{aligned} \tag{1}$$

where $\nabla f_{\mathcal{S}}(\boldsymbol{x}i)$ and $\nabla^2 f_{\mathcal{S}}(\boldsymbol{x}_i)$ are the gradient and Hessian of the SDF at $\boldsymbol{x}_i$ calculated by numerical differentiation with a progressively smaller step size [3].

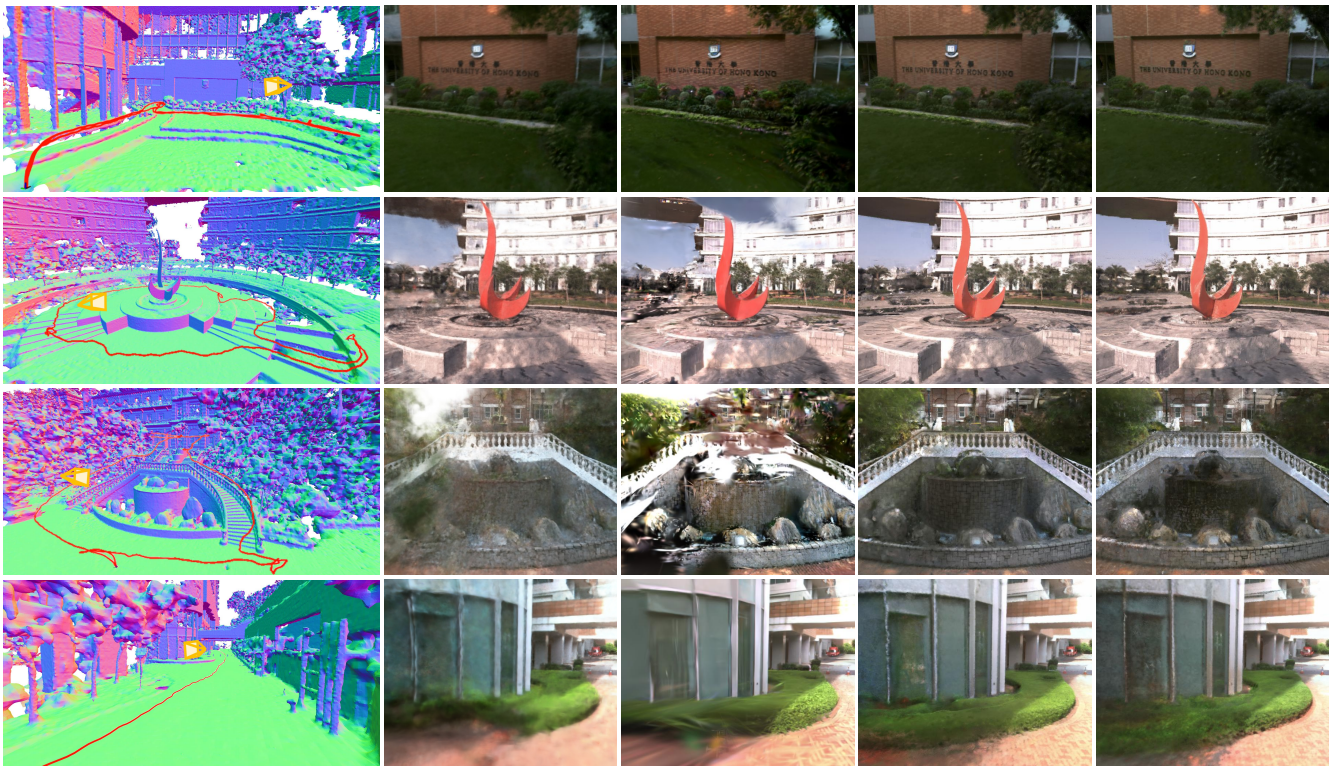
The overall training loss is defined as follows:

$$\mathcal{L} = \mathcal{L}_{sdf} + \lambda_{rgb}\mathcal{L}_{rgb} + \lambda_{eik}\mathcal{L}_{eik} + \lambda_{curv}\mathcal{L}_{curv}, \tag{2}$$

where $\lambda_{eik} = 0.1$ and $\lambda_{curv} = 5\times10^{-4}$ are the weights for the Eikonal, and curvature losses, respectively. We schedule the λ_{rgb} linearly increases from 10^{-4} to 10 during the training to ensure that the NeRF learns on a well-structured NSDF to avoid local minima.

C. Training

1) Outlier removal: The zero-level set of the NDF defines the fitting surface, and the inferred signed distance values of the input LiDAR points naturally indicate the reconstruction error. In dynamic scenes, points on dynamic objects are supervised with an SDF value of zero. However, LiDAR points on the static background traversing through these dynamic points produce a supervised SDF value greater than zero. Given that dynamic points are typically sparse, this region is often dominated by the background points and has a SDF value greater than zero. Based on this observation, we propose an outlier removal strategy that periodically infers



(a) Mesh (Ours) (b) InstantNGP (c) 3DGS† (d) Ours (e) wo dir. scheduler

Fig. 2. The qualitative results of FAST-LIVO2 datasets (scenes from top to bottom are Campus, Sculpture, Culture, and Drive). We show our surface reconstruction results on the left, and the red line indicates the training path and the orange cameras indicates the extrapolation views for the right side's rendering results.

the signed distance values of training LiDAR points and eliminates points whose predicted signed distance values are more than ϵ away from 0. This enables the NDF remains a static structure field and also helps to erase dynamic objects in rendering, as shown in Fig. 6.

2) Directional embedding scheduler: To synthesize photorealistic novel-view images, the neural radiance field considers the view direction $\boldsymbol{d}$ to output the view-dependent color $\boldsymbol{c}$ at each position $\boldsymbol{x}$: $\boldsymbol{c} = f_C(\boldsymbol{x}, \boldsymbol{d})$, where the view direction is encoded using a 4-degree spherical harmonics encoding [4]. The tightly coupled position and view direction in training make the rendering quality in extrapolation views show a degree of color degeneration, especially at places that have only an image observations observing from similar directions, as shown in Fig. 2 (d). We consider that the surface's color is composed of view-independent diffuse color and view-dependent specular color. We schedule the degree of directional embedding to learn the surface diffuse color at the degree of 0 (i.e., view-independent feature) in the beginning and gradually increase from 0 to 4 during training to learn high-frequency view-dependent specular color.

3) Scene contraction for background rendering: To address the infinitely far background space, we define a boundary space that extends outward from the occupied map for background color rendering. For each ray, n_b points are uniformly sampled in the boundary space and contracted [5] as follows:

$$\text{contract}(\boldsymbol{x}) = \begin{cases} \boldsymbol{x} & , \ |\boldsymbol{x}| \leq 1 \\ \left(1 + B\left(1 - \frac{1}{|\boldsymbol{x}|}\right)\right)\left(\frac{\boldsymbol{x}}{|\boldsymbol{x}|}\right) & , \ |\boldsymbol{x}| > 1 \end{cases}, \quad (3)$$

where B is the size of the extending boundary space and is defined as the same voxel size as the occupancy map. The contracted samples are used to color the infinite space via volume rendering with false surfaces.

II. Experiments

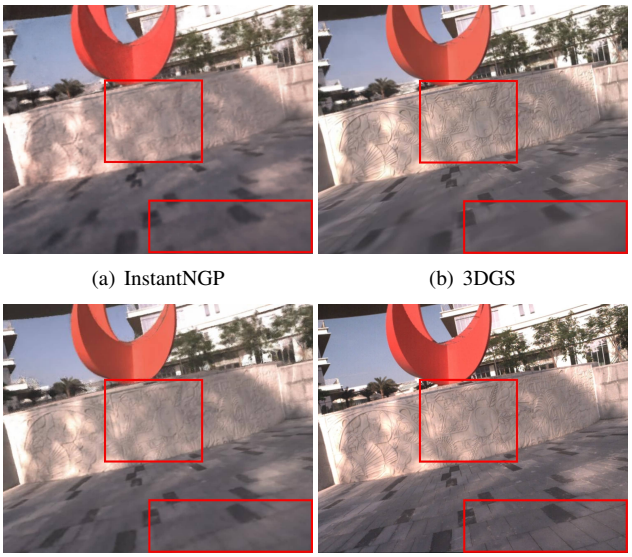
A. Implementation Details

We represent our neural fields following a similar architecture to InstantNGP [6], utilizing a combination of multi-resolution hash encoding and a tiny MLP decoder. The hash encoding resolution spans from 2^5 to 2^{21} with 16 levels, and each level contains 2^{19} two-dimensional feature vectors. Given any position x, the hash encoding concatenates each level's interpolation features to form a feature vector of size 32. One encoding feature is fed to a geometry MLP with 64-width and 3 hidden layers to obtain the SDF value and scale. Another encoding feature is concatenated with the spherical harmonics encoding of view directions and fed to an appearance MLP with 64-width and 3 hidden layers to obtain the view-dependent color. We sample 8192 rays for NDF training and follow InstantNGP [6] to fix the batch size of point samples as 256000 while the batch size of rays is

dynamic per iteration. We take 20000 iterations for training, with outlier removal performed every 2000 iterations. We implement our method using LibTorch and CUDA. All experiments are conducted on a platform equipped with an Intel i7-13700K CPU and an NVIDIA RTX 4090 GPU.

B. FAST-LIVO2 Dataset

We present more surface reconstruction results and extrapolation rendering results on the FAST-LIVO2 datasets, as shown in Fig 2. We further dive into the performance differences between volume-rendering-based methods (Instant-NGP and Ours) and rasterization-based methods (3DGS), as shown in Fig. 3. InstantNGP and Our methods show more reasonable rendering results and fewer artifacts overall, such as the ground in the down-left image. While 3DGS shows powerful capability to capture high-frequency textures, such as the carved wall in the middle image.



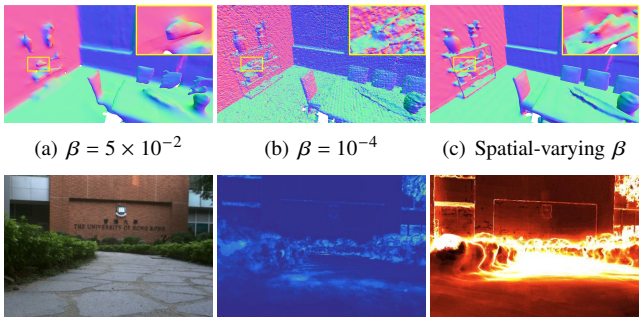
(a) InstantNGP (b) 3DGS

(c) Ours (d) Ground Truth

Fig. 3. We show the performance differences between volume-rendering-based methods (InstantNGP and Ours) and rasterization-based method (3DGS).

C. Ablation Study

1) Spatial-varying scale: To demonstrate the necessity of spatial-varying scale, we conducted experiments on the Replica dataset's room-2 scene, where we applied 2cm normal noise to the ground truth depth. As shown in Fig. 4 (a-c), a large scale β results in smoother surface reconstructions with a loss of details, while a small scale leads to overfitting and noisy surface reconstructions. A spatial-varying scale is introduced to the neural distance field to adapt to the scene's various granularity and preserve levels of detail for objects of different scales. In Fig. 4 (d-e), the rendering scale that accumulates the scale along the rays, shows that rays traverse fuzzy geometries return larger scale β indicating more uncertainty along these rays. Then the larger scale resulting in lower density (Eq. 3) makes the structure-aware



(a) $\beta = 5 \times 10^{-2}$ (b) $\beta = 10^{-4}$ (c) Spatial-varying β

(d) Render Color (e) Render Scale β (f) Sample numer

Fig. 4. In the first row, we show different scale settings in the Replica Room-2 scene. The average spatial-varying β in the scene is 1.6×10^{-4}. In the second row, we show the render scale (the lighter color corresponds to the bigger scale) and sample numer per ray (the lighter color corresponds to the bigger number) in Campus scenes.

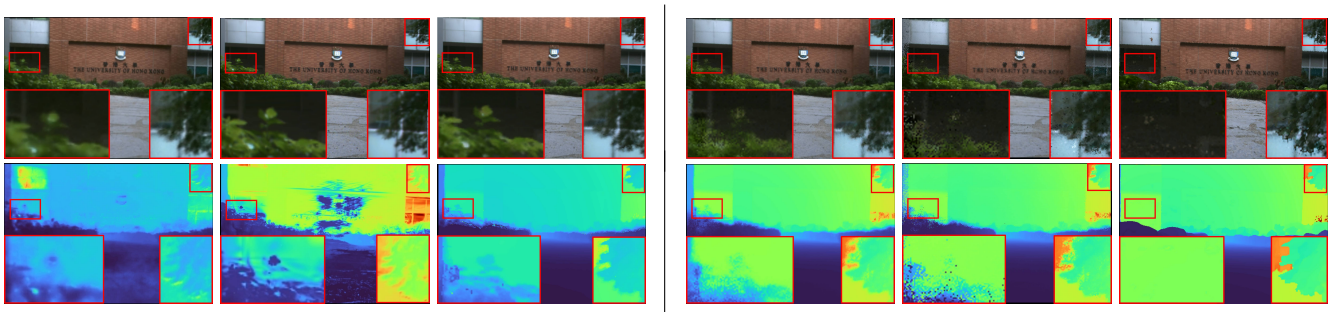
sampling return more samples on these rays to meet the transmittance requirements.

2) Structure-aware sampling: To validate the proposed structure-aware sampling strategy, we compare the qualitative reconstruction results with occupancy sampling [6], probabilistic density function (PDF) sampling [7] and surface rendering, as shown in Fig. 5 (Right). The occupancy sampling takes uniform samples in every occupied grid. The PDF sampling first takes uniform samples along the ray to obtains the cumulative distribution function, and then generates samples using inverse transform sampling. Surface rendering renders the scene with the first intersecting surfaces' radiance. The proposed structure-aware sampling strategy better adapts to the SDF prior and focuses more on the surface than previous sampling methods, avoiding the skipping of fuzzy geometries as seen in surface rendering for more accurate structure rendering.

3) Outlier removal: To validate the necessity of outlier removal (Sec. I-C.1) for real-world applications, we conducted experiments on the FAST-LIVO2 dataset's Drive scene, where dynamic objects (cart pusher and car) move in the scene. As shown in Fig. 6, the neural distance field with outlier removal can remove false surfaces caused by dynamic objects and regularize low density in the space traversed by the dynamic objects, and the more consistent background appearance filters out temporary dynamic objects in rendering.

4) Directional embedding scheduler: We propose to use a directional embedding scheduler (Sec. I-C.2) to enhance the generalization of the neural radiance field in extrapolating views, which can be verified in Fig. 2 (d). The rendering image with the directional embedding scheduler can better generalize the extrapolating views for more consistent color, especially in the free-view trajectory scenes, like Drive. Meanwhile, the directional embedding scheduler has little influence on the interpolation rendering results, as shown in Tab. I (wo dir. sched.).

5) Visual-aided structure: To validate the influence of image measurements on surface reconstructions, we first



(a) InstantNGP (b) 3DGS† (c) Ours (d) Occ. Sampling (e) PDF Sampling (f) Surface Rendering

Fig. 5. (Left) Rendering results between different baselines on the FAST-LIVO dataset's Campus scene. (Right) Rendering results with different sampling and rendering methods.

(a) Source Image (b) InstantNGP (c) 3DGS† (d) Ours (e) wo outlier removal

Fig. 6. Ablation study of outlier removal on the FAST-LIVO2 Drive dataset where dynamic objects (cart pusher and car) move in the scene.

TABLE I

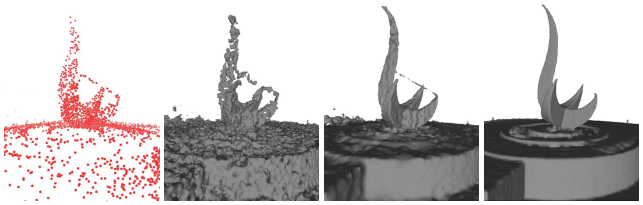
QUANTITATIVE RESULTS ON THE FAST-LIVO2 DATASET.

Metrics	Methods	Campus	Sculpture	Culture	Drive	Avg.
SSIM↑	InstantNGP	0.789	0.698	0.670	0.697	0.714
	3DGS†	**0.849**	**0.769**	0.726	**0.778**	**0.780**
	Ours	0.834	0.729	**0.727**	0.764	0.764
	wo dir. sched.	0.833	0.726	0.729	0.767	0.764
PSNR↑	InstantNGP	28.880	22.356	21.563	24.145	24.236
	3DGS†	**31.310**	**24.128**	21.764	25.837	25.760
	Ours	30.681	23.453	**24.695**	**25.941**	**26.193**
	wo dir. sched.	30.572	23.534	24.797	26.072	26.244
LPIPS↓	InstantNGP	0.255	0.376	0.428	0.416	0.369
	3DGS†	**0.182**	**0.266**	0.361	0.296	**0.276**
	Ours	0.210	0.321	**0.350**	**0.293**	0.293
	wo dir. sched.	0.213	0.330	0.351	0.293	0.297

downsampled the point cloud in the Sculpture scene and compared the reconstruction results with and without visual supervision $\mathcal{L}_{rgb}$. As shown in Fig. 7, the neural radiance fields can complete the structure from sparse point clouds and avoid overfitting in scenes.

D. Training and Rendering Efficiency

So far the bottleneck of the efficiency is greatly impeded by the unbalanced sphere tracing's steps, once there is a ray that does not converge to the surface, the other converged rays need to wait until it is finished or reach the maximum steps. The training time for a Replica scene takes about 20 minutes and the rendering time for a 1200x680 image takes about 0.07 seconds (13Hz). In the future, it could be solved by conducting a ray-oriented rendering for ray rendering instead of batch rendering.



(a) Point cloud (b) wo visual (c) Ours (d) Ours (Dense)

Fig. 7. We show the Sculpture scene's reconstruction results from downsampled point clouds (a) without (b) and with (c) visual supervision. And (d) shows the reconstruction result from dense point clouds.

REFERENCES

[1] A. Gropp, L. Yariv, N. Haim, M. Atzmon, and Y. Lipman, "Implicit geometric regularization for learning shapes," in *Proceedings of the 37th International Conference on Machine Learning*, 2020, pp. 3789–3799.

[2] H. Yang, Y. Sun, G. Sundaramoorthi, and A. Yezzi, "Steik: Stabilizing the optimization of neural signed distance functions and finer shape representation," in *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[3] Z. Li, T. Müller, A. Evans, R. H. Taylor, M. Unberath, M.-Y. Liu, and C.-H. Lin, "Neuralangelo: High-fidelity neural surface reconstruction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 8456–8465.

[4] D. Verbin, P. Hedman, B. Mildenhall, T. Zickler, J. T. Barron, and P. P. Srinivasan, "Ref-nerf: Structured view-dependent appearance for neural radiance fields," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2022, pp. 5481–5490.

[5] 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.

[6] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *arXiv preprint arXiv:2201.05989*, 2022.

[7] 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.