SDF-based RGB-D Camera Tracking in Neural Scene Representations

Leonard Bruns[1] and Fereidoon Zangeneh[1,2] and Patric Jensfelt[1]

***Abstract*— We consider the problem of tracking the 6D pose of a moving RGB-D camera in a neural scene representation. Different such representations have recently emerged, and we investigate the suitability of them for the task of camera tracking. In particular, we propose to track an RGB-D camera using a signed distance field-based representation and show that compared to density-based representations, tracking can be sped up, which enables more robust and accurate pose estimates when computation time is limited.**

I. Introduction

Recently, neural scene representations have been shown to possess promising characteristics for creating dense reconstructions of environments [1], [2], [3]. The continuous map stored in the weights of these coordinate-based networks can densely represent environments through quantities such as radiance fields [1], occupancy probability [4], [5], and signed-distance fields (SDFs) [6], [2], [7], [8].

To map an environment with a neural scene representation using a camera with unknown trajectory, camera tracking is necessary [9], [10]. Camera tracking can be done by direct comparison of observations and renders of the mapped scene. Rendering of views is typically done via volume rendering, which involves densely querying the viewing frustum. This requires many samples and is, therefore, time consuming.

In this paper, we investigate whether recently proposed SDF-based neural scene representations [2], [7] can be used for more efficient tracking when paired with RGB-D cameras compared to volume rendering-based tracking, such as in iMAP [10]. We propose a novel tracking scheme that estimates the camera pose by directly querying the observed surface points and minimizing the returned distances. This obviates the need for volume rendering, increasing the time budget that instead can be used for incorporating more of the observations.

A. Problem Definition

Given an initial camera pose ${}^{\mathrm{w}}_{0}\mathbf{T}_{\mathrm{c}}$ and a stream of RGB-D images $(\mathbf{I}_i \in \mathbb{R}^{H\times W\times 3}, \mathbf{D}_i \in \mathbb{R}^{H\times W}), i = 1, ..., M$ we want to find estimates ${}^{\mathrm{w}}_{i}\widetilde{\mathbf{T}}_{\mathrm{c}}$ of the true camera poses ${}^{\mathrm{w}}_{i}\mathbf{T}_{\mathrm{c}}$. We assume a known static environment that has previously been encoded in a neural scene representation.

[1]The authors are with the Division of Robotics, Perception and Learning (RPL), KTH Royal Institute of Technology, Stockholm, Sweden {leonardb,fzk,patric}@kth.se.

[2] Fereidoon Zangeneh is also with Univrses AB.

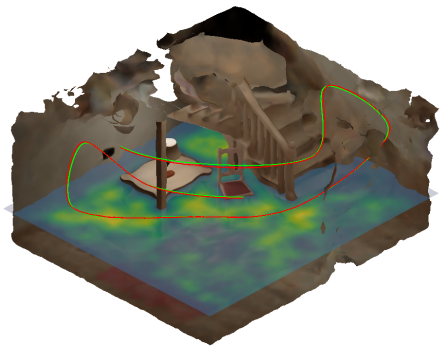

Fig. 1. True (—) and tracked (—) camera trajectory and signed distance field used for tracking. The mesh was extracted from the zero-isosurface of the signed distance field.

II. Method

A. Scene Representations

We consider two neural scene representations: iMAP [10] and NeuS [2]. We denote their respective networks by

$$f_{\mathrm{iMAP}} : \mathbb{R}^3 \to \mathbb{R}^3 \times \mathbb{R} \quad \mathbf{x} \mapsto (\mathbf{c}, \sigma) \quad \text{and} \quad f_{\mathrm{NeuS}} : \mathbb{R}^3 \to \mathbb{R}^3 \times \mathbb{R} \quad \mathbf{x} \mapsto (\mathbf{c}, d), \tag{1}$$

where $\mathbf{c}$ denotes color, σ density, and d the signed distance to the closest surface.

B. Camera Tracking

1) Density-based Rendering: In density-based representations (such as iMAP or NeRF), the camera pose can be optimized by rendering the color and depth image at the previous estimate and formulating a loss by comparing the rendered and observed color and depth images. This has previously been demonstrated by iNeRF [11] for RGB data only and with RGB-D data by iMAP [10]. Due to the expensive nature of volume rendering, both of these methods only render a small subset of pixels to reduce the computation time. We use volume rendering as described in iMAP as our baseline.

2) SDF-based Optimization: Under ideal conditions, all the observed depth points originate from surfaces in the environment. Therefore, these points should fall onto zero crossings of the SDF. We propose to exploit this synergy between RGB-D cameras and SDFs by directly querying the scene representation at the points from the RGB-D point cloud after transforming it into the world frame using the previous pose estimate. Instead of defining a loss based on the render differences, we formulate the loss based on the query points' colors and signed distances.

TABLE I

ROOT-MEAN-SQUARE AVERAGE TRAJECTORY ERROR (IN m) ON THE NEURAL RGB-D DATASET BY [3] (> 0.2 m, **BETTER**).

		Breakfast	Compl. Kitchen	Green Room	Grey-white	Kitchen	Morning Apart.	Staircase	Thin Geo.	White-room
$r = 10$ Hz	iMAP with VR	1.575	2.171	0.702	**0.156**	2.196	**0.107**	3.582	0.312	2.744
	NeuS with SDF	**0.096**	**0.118**	**0.215**	0.255	**0.110**	0.114	**0.052**	**0.031**	**0.067**
$r = 5$ Hz	iMAP with VR	0.082	0.105	**0.048**	**0.032**	1.313	**0.027**	3.868	0.035	0.080
	NeuS with SDF	**0.043**	**0.049**	0.100	0.072	**0.044**	0.073	**0.045**	**0.017**	**0.022**
$r = 2$ Hz	iMAP with VR	**0.021**	**0.020**	**0.015**	0.077	**0.077**	**0.007**	3.987	**0.012**	0.022
	NeuS with SDF	0.038	0.038	0.067	**0.043**	1.148	0.067	**0.044**	0.014	**0.015**

TABLE II

AVERAGE NUMBER OF OPTIMIZATION STEPS UNDER VARYING TIME CONSTRAINTS AND NUMBERS OF PIXEL SAMPLES n.

	n	50 ms	100 ms	500 ms
VR	128	3.2	6.6	36.2
	512	3.2	6.8	36.1
	1024	2.2	5.0	27.0
SDF	2048	7	15	77
	4096	7.1	15.3	79.3
	16384	5.6	12.7	65.8

Specifically, we sample n points from the current RGB-D image $(\mathbf{I}_i, \mathbf{D}_i)$ and compute the colored point set ${}^{\mathrm{c}}\mathcal{P}_i = \{({}^{\mathrm{c}}\mathbf{p}_k \in \mathbb{R}^3, \mathbf{c}_k \in \mathbb{R}^3) | k = 1, ..., n\}$ in the camera frame. We then optimize the loss

$$l = \lambda_{\mathrm{SDF}} \frac{1}{n} \sum_{k=1}^{n} |\tilde{d}_k| + \lambda_{\mathrm{color}} \frac{1}{3n} \sum_{k=1}^{n} \|\tilde{\mathbf{c}}_k - \mathbf{c}_k\|_1, \quad (2)$$

where $(\tilde{\mathbf{c}}_k, \tilde{d}_k) = f_{\mathrm{NeuS}}({}^{\mathrm{w}}_{i}\widetilde{\mathbf{T}}_{\mathrm{c}} \, {}^{\mathrm{c}}\mathbf{p}_k)$, and λ_{SDF} and λ_{color} are fixed hyperparameters. Similarly to iMAP, we sample a new set of n points for every optimization iteration.

Note that the same optimization is not applicable to density-based representations, which typically converge to very sharp boundaries without smooth spatial gradients that could guide the optimization (see Fig. 1 for an example of a learned SDF). Furthermore, the isosurface on which the depth points lie is undefined. Similarly, occupancy fields [4], despite having a well-defined isosurface at 0.5, converge to very sharp transitions under ideal training conditions.

III. EXPERIMENTS

1) Implementation Details: We use the same network architecture as iMAP for both methods. NeuS contains a single additional trainable parameter for the standard deviation of the s-density [2]. Both methods are implemented in PyTorch. We parametrize the camera pose as a position ${}^{\mathrm{w}}_{i}\tilde{\mathbf{t}}_{\mathrm{c}} \in \mathbb{R}^3$ and unit quaternion ${}^{\mathrm{w}}_{i}\tilde{\mathbf{q}}_{\mathrm{c}} \in \mathbb{H}_1$ (we renormalize after every optimization step). We use Adam optimizer [12] with learning rates 5×10^{-4} and 1×10^{-3} for position and orientation, respectively.

2) Evaluation Protocol: We report the root-mean-square of the absolute trajectory error (ATE)

$$ATE = \sqrt{\frac{1}{M} \sum_{i=1}^{M} \| {}^{\mathrm{w}}_{i}\tilde{\mathbf{t}}_{\mathrm{c}} - {}^{\mathrm{w}}_{i}\mathbf{t}_{\mathrm{c}} \|_2^2}, \quad (3)$$

where ${}^{\mathrm{w}}_{i}\tilde{\mathbf{t}}_{\mathrm{c}}$ and ${}^{\mathrm{w}}_{i}\mathbf{t}_{\mathrm{c}}$ denote the translation part of ${}^{\mathrm{w}}_{i}\widetilde{\mathbf{T}}_{\mathrm{c}}$ and ${}^{\mathrm{w}}_{i}\mathbf{T}_{\mathrm{c}}$, respectively. We initialize the tracking using the ground-truth starting pose ${}^{\mathrm{w}}_{0}\mathbf{T}_{\mathrm{c}}$ and process every frame in the sequence with a fixed tracking rate r (i.e., we do not force a certain playback frame rate or drop frames due to too slow tracking). For each frame, we estimate the average time per optimization iteration and continue to the next frame if the remaining time budget is not sufficient for another iteration.

3) Results: We report results on the sequences of the dataset by [3] in Table I. *NeuS with SDF* refers to our proposed SDF-based tracking and *iMAP with VR* refers to iMAP with volume rendering as described in [10].

We can see that our proposed SDF-based loss fails less frequently when tracking at faster frame rates. The results for $r = 2$ Hz indicate that volume rendering achieves lower ATE when sufficient optimization time is available. We hypothesize that tracking using volume rendering might be more accurate here, since the underlying representation was trained via volume rendering as well. By contrast, our SDF-based tracking loss differs significantly from the training time loss.

In Table II we further show the number of iterations for different time budgets and number of samples n. Note that because our SDF-based tracking does not rely on expensive volume rendering, we can incorporate more of the available sensor data into each optimization step.

IV. CONCLUSION

Our experiments confirm that an SDF-based representation can be used to more efficiently track an RGB-D camera inside a neural scene representation. This comes at the cost of a more involved mapping task, which in our case involved the eikonal term [13], which requires the computation of second-order gradients and therefore slows down training roughly by a factor of two. In the future we want to investigate whether the SDF loss can similarly be used to speed up mapping by giving direct supervision to the isosurface in combination with volume rendering.

ACKNOWLEDGMENT

This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.

REFERENCES

[1] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "NeRF: Representing scenes as neural radiance fields for view synthesis," in *Proceedings of the European Conference on Computer Vision*. Springer, 2020, pp. 405–421.

[2] 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," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[3] D. Azinović, R. Martin-Brualla, D. B. Goldman, M. Nießner, and J. Thies, "Neural RGB-D surface reconstruction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[4] 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.

[5] 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.

[6] 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.

[7] L. Yariv, J. Gu, Y. Kasten, and Y. Lipman, "Volume rendering of neural implicit surfaces," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[8] J. Ortiz, A. Clegg, J. Dong, E. Sucar, D. Novotny, M. Zollhoefer, and M. Mukadam, "iSDF: Real-time neural signed distance fields for robot perception," *arXiv preprint arXiv:2204.02296*, 2022.

[9] Z. Wang, S. Wu, W. Xie, M. Chen, and V. A. Prisacariu, "NeRF–: Neural radiance fields without known camera parameters," *arXiv preprint arXiv:2102.07064*, 2021.

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

[11] L. Yen-Chen, P. Florence, J. T. Barron, A. Rodriguez, P. Isola, and T.-Y. Lin, "iNeRF: Inverting neural radiance fields for pose estimation," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2021, pp. 1323–1330.

[12] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proceedings of the International Conference on Learning Representations*, 2015.

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