Proceedings of the 2023 International Conference on Machine Learning and Automation
DOI: 10.54254/2755-2721/37/20230484

Current study on NeRF-based techniques in SLAM frameworks

Kunpeng Ying

School of Precision Instrument and Optoelectronics Engineering, Tianjin University, Tianjin, China

yingkunpeng@tju.edu.cn

Abstract. This paper comprehensively examines the application and current research status of Neural Radiance Fields (NeRF) technology within Simultaneous Localization and Mapping (SLAM) systems. NeRF, which has gained significant attention since 2020, has evolved into a powerful method for reconstructing 3D scenes from images, offering advantages such as continuous scene representation and photorealistic novel view synthesis. However, it also comes with drawbacks, including substantial training data requirements, limited model generalization, and challenges in map scalability. In contrast, SLAM is a complex, real-time, efficient, and robust system capable of tracking camera motion and constructing environmental maps in real-time, with no limitations on map size. The integration of NeRF technology into SLAM enhances the capabilities of various modules, including Mapping, Tracking, Optimization, Loop Closure, and Localization, providing potential advantages. Beginning with an exploration of NeRF's fundamental principles and its inherent strengths and weaknesses, this paper delves into the significant implications of integrating NeRF into the SLAM pipeline. It addresses the challenges encountered during implementation and outlines potential future directions. The aim is to provide a clear elucidation of the evolving landscape of the combined NeRF and SLAM approach, serving as a reference for researchers interested in pursuing this research direction.

Keywords: NeRF, SLAM, Scene Reconstruction, Map Representation, Computer Vision.

1. Introduction

NeRF, short for Neural Radiance Fields, represents scenes in the form of a neural radiance field, aiming to achieve view synthesis for novel perspectives. Essentially, it consists of two simple fully connected Multilayer Perceptrons (MLP) that model spatial positions and viewing directions to map them to point density and RGB values.

The specific process involves the first neural network, denoted as fσ, which takes spatial position x as input and outputs the point density σ for that location, along with an associated feature vector e. The second network, fc, is employed to estimate the RGB values. These RGB values are viewpoint-dependent due to factors such as material properties, lighting, and reflections. The input of this network is crucial for achieving photo-realistic rendering, as the colors of the same position can vary based on the viewing angle. The viewpoint d and the feature vector e obtained from the density network fσ are combined as inputs to the network fc, resulting in the final RGB values (see Figure 1).

DOI: 10.54254/2755-2721/37/20230484

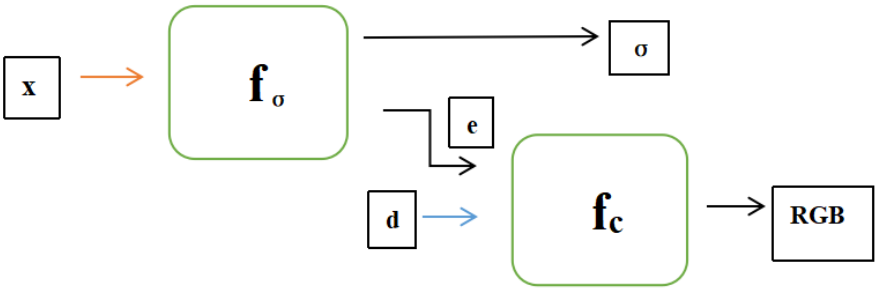

Figure 1. Fundamentals of NeRF.

For volume rendering, when generating an image, a sampling ray is generated from the camera center. Points are sampled along the ray, and the previously mentioned MLP is used to calculate the volume density σ and RGB values. A differentiable rendering formula is employed to sum up all sampled values and obtain a pixel value. Traditional representations like point clouds or TSDFs have occlusion issues – people might know the RGB value of a spatial point, but they lack information about what's behind that point along the viewing direction. This often leads to gaps or holes in models or maps, preventing novel view synthesis. NeRF overcomes this by predicting volume density along the entire ray, enabling optimization to infer accurate density and RGB values even for occluded regions, thus completing novel view synthesis. This is the underlying principle of NeRF [1].

From the principle, it is easy to deduce NeRF's advantages: continuous scene representation and photo-realistic novel view synthesis. As the input positions (xyz) to the network are not bound by resolution constraints, they can take arbitrary values, resulting in continuous scene representation.

However, NeRF's drawbacks are evident. Training a model for an object often requires hundreds of images and several days. Rendering an image can take tens of seconds. Changing the model necessitates retraining since the model obtained essentially lies within the MLP's parameters, implying that a new scene would require a new MLP. The limitation of NeRF being an MLP also hampers map scalability. MLP's fitting capability is ultimately limited; when trained with a series of scene images, it tends to favor memorizing subsequent data, a phenomenon known as the "forgetting problem."

SLAM (Simultaneous Localization and Mapping) involves the synchronization of localization and mapping tasks within a system. A SLAM system comprises multiple modules [2]: Firstly, the front-end tracking module estimates camera poses from input images or depth data, computes camera motion, and extracts keyframes. Secondly, the back-end mapping module constructs a map from the keyframes, often creating a point cloud map based on feature points. Additionally, this module performs local or global camera pose and map optimization. Lastly, the loop closing module, a vital component, corrects drift that accumulates during prolonged operation. When this module detects that the camera revisits a previously explored location, it rectifies the accumulated drift error. Additionally, the system includes a location module that ensures reacquisition of camera pose information when the front-end tracking module loses track, allowing the SLAM process to continue.

2. The Significance of Integrating NeRF with SLAM

A SLAM system is capable of real-time tracking of camera motion and simultaneous construction of an environmental map, with an unlimited map scale. This stands in stark contrast to the fundamental characteristics of NeRF. So, why would researchers pursue NeRF-based SLAM, and what breakthroughs can NeRF bring to a well-established SLAM system?

The SLAM system serves as a crucible for various 3D reconstruction techniques, and NeRF's map representation naturally integrates into the back-end of SLAM systems to test its performance. The most evident enhancement NeRF brings to SLAM lies in the mapping aspect. What advantages does a NeRF map offer? Firstly, it can address map occlusion by utilizing ray sampling and volume density, allowing the map to learn about occluded areas and thereby facilitating hole filling and novel view

DOI: 10.54254/2755-2721/37/20230484

synthesis. Novel view synthesis, in turn, can be utilized for predicting camera motion. Imagine a camera capturing an image while in motion. To estimate the camera pose based on this image, one can initially employ motion models, such as constant velocity models, to estimate a coarse pose. Using this pose, a NeRF rendering of an image is generated, containing previously unseen content relevant to the new frame. By computing a loss between the rendered and actual images, the camera pose and local map can be optimized concurrently. In fact, this principle underpins the camera tracking approach in iMAP [3] and Nice-SLAM [4].

Furthermore, NeRF offers scene editing capabilities, which are particularly favorable for VR and AR applications, constituting one of NeRF's practical applications. NeRF-based SLAM eliminates the need for intermediate steps like point cloud creation, mesh generation, and texture mapping. It achieves an end-to-end process, generating AR or VR-ready maps directly.

3. Research Status of NeRF Application in Different Stages of SLAM

3.1. Mapping

The most intuitive role of NeRF in SLAM is to serve as the backend map representation. SLAM demands maps that can be continuously expanded and easily queried. Correspondingly, within NeRF, this translates to properties such as scalability, convergence, and fast rendering speed.

iMAP implemented a NeRF-based SLAM system using an MLP to represent the map [3]. However, the map representation capability of a single MLP in iMAP is limited, and for real-time applications, rendering performance is compromised, restricting it to modeling small scenes. For larger scenes, termed unbounded scenes within NeRF, this poses challenges.

Nice-SLAM [4] follows the idea of iMAP's camera tracking by using three voxel grids of different resolutions, nested to represent the scene, and then decoded with a pre-trained decoder that fuses the decoded content of each layer of the grids for voxel rendering.

Mip-NeRF 360 [5] uses Extended Kalman Filtering based on its predecessor Mip-NeRF [6] to improve the rendering quality and accelerate the rendering by mapping the unbounded scene inside a bounded coordinate space, but it still fails to satisfy the condition of arbitrary motion of the camera position in the SLAM system.

Block-NeRF addresses large scenes by employing multiple MLPs [7]. It reconstructs an entire city block of San Francisco through scalable neural view synthesis. The scene is divided into distinct NeRF blocks, each individually trained, updated, and then fused through appearance alignment strategies. Nevertheless, distant blocks not within the visible radius are not involved in view synthesis, potentially omitting far-away objects visible from certain angles. While Block-NeRF addresses many of SLAM's map requirements, it doesn't fully resolve background modeling issues.

Vox-fusion is also a NeRF-based SLAM method, prioritizing geometry reconstruction for VR and AR applications. Its tracking performance surpasses that of nice-slam, reducing memory consumption from 200M to 0.15M while achieving map expansion [8].

Lisus et al. made improvements to the NICE-SLAM algorithm: considering depth uncertainty by weighting depth loss according to depth measurement noise, incorporating motion information through IMU data for enhanced camera tracking and motion handling, and splitting NeRF into finite foreground grids and background spheres to handle environments of any size and retain visual information beyond predefined grids [9]. This boosted tracking accuracy from 85% to 97%, yielding better results in trajectory estimation and scene reconstruction.

In summary, NeRF map representations can be categorized into three types: fully implicit, fully explicit, and hybrid. Full implicit utilizes MLPs to store maps, extending the original NeRF concept, but struggles with scalability and real-time application. Full explicit denotes map representation without MLPs, favoring fast convergence and easy expansion, albeit with suboptimal voxel rendering quality. Hybrid combines implicit and explicit methods, preserving the advantages of traditional map representations while achieving rendering quality comparable to NeRF, making it most suitable for

DOI: 10.54254/2755-2721/37/20230484

SLAM. It's important to note that as long as the map maintains continuity, NeRF can perform voxel rendering without necessarily relying on MLPs.

3.2. Tracking

This section can be considered one of the most critical aspects of SLAM: estimating the camera's precise motion trajectory frame by frame. This is essential for determining where to place the locally reconstructed 3D points within the global map. Thus, the front-end tracking component of SLAM demands accurate, fast, and robust camera pose estimation. However, what is the relevance of NeRF to the back-end?

For front-end tracking, one can employ a well-established 2D visual odometry algorithm while dedicating a separate thread to train the NeRF map. This configuration represents a type of NeRF-based SLAM method where traditional visual odometry (VO) serves as the front end, and the NeRF map serves as the backend [10].

Nice-SLAM uses a tracking strategy where an initial pose generates depth and RGB images from the map. A depth loss and photometric loss are computed based on the rendered and input images, optimizing camera pose and refining feature vectors within the grid, achieving simultaneous camera pose estimation and local map optimization. This is NeRF's inverse rendering and tracking. However, this method's trajectory accuracy does not match that of traditional SLAM techniques.

Nicer-SLAM switches from RGB-D to RGB input, integrating monocular depth estimation [11]. Furthermore, Nicer-SLAM includes additional losses such as depth estimation loss, normalization loss, and optical flow loss on top of depth and photometric losses from Nice-SLAM, resulting in a total of five losses. This enhancement elevates both localization accuracy and rendering quality to a new level.

In conclusion, the Nice-SLAM concept utilizes NeRF's inverse rendering to concurrently optimize camera poses and update local maps, offering a compelling framework for integrating NeRF into SLAM. This approach easily combines with traditional visual odometry, yielding superior results. Nonetheless, NeRF inverse rendering also has its drawbacks, particularly its sensitivity to initial pose estimation. Significant initial pose deviations may lead to inaccurate local map optimization, counterproductive to the mapping process.

3.3. Optimization

Within NeRF, camera pose optimization is also necessary as inaccurate camera poses can hinder reconstruction. Simultaneously conducting reconstruction and optimization is a mutually reinforcing challenge. Barf introduced image alignment theory into NeRF, co-optimizing camera poses and reconstruction models. Therefore, during camera pose adjustment, low-pass filtering is applied across different frequency bands, effectively optimizing poses through alignment with low-frequency image information [12].

Orbeez-SLAM achieves ultra-fast super-resolution reconstruction and accurate camera pose estimation by combining implicit neural representations and visual odometry techniques, enhancing SLAM's performance and efficiency [13].

ESLAM employs a hybrid representation approach, decomposing scenes into coarse and fine-grained feature planes, decoded through multi-layer perceptron (MLP) decoders to generate TSDF and original colors [14] . By introducing a novel rendering method, ESLAM optimizes TSDF, original colors, camera poses, and feature planes simultaneously, enabling incremental scene reconstruction while estimating the current camera position within the scene.

It's evident that NeRF-based maps, unlike traditional maps, require iterative pose optimization, and for certain features, transferring mature geometry-based optimization algorithms is challenging, highlighting an area for breakthrough.

3.4. Loop Closure

In SLAM systems, sensor data collection and long-term camera pose estimation can accumulate drift errors. Loop closure detection detects whether a camera has revisited a previously traversed location.

DOI: 10.54254/2755-2721/37/20230484

This helps rectify global map drift errors by comparing camera pose errors during two visits to the same location. ORB-SLAM2 employs a bag-of-words model to retrieve and match ORB features of the current frame, recognizing loop closures. Once detected, keyframe poses require correction.

For NeRF, the backend map encodes keyframe image content, particularly for implicit maps. This implies an irreversible mapping process, and these encodings can't undergo pose adjustments, rendering implicit maps unsuitable for transformations. However, Yuan et al.'s work addresses this issue by achieving an equivariant mapping from SO3 to the feature space [15]. They synchronously adjust camera poses and implicit map representations, creating a SLAM system using voxel grids as the backend through ORB-SLAM2's front-end localization module. This achieves loop closure correction for implicit maps.

3.5. Localization

Loop closure detection requires global map-based camera localization, which also facilitates relocalization after camera tracking loss or SLAM interruption. IR-MCL [16] achieves global localization within a NeRF map constructed from 2D radar scans using a Monte Carlo method. It samples multiple candidate camera poses in space, computes differences between these poses and actual observations, generates weights, eliminates candidates with low weights, iteratively updates pose, and ultimately converges to the vicinity of the actual pose. The remaining poses are weighted to derive the final pose. This concept might be applicable to 3D maps as well, although three-dimensional space convergence could be more challenging.

4. Conclusion

In conclusion, NeRF-based SLAM has laid the groundwork for its development. As elucidated earlier, when the backend map employs a sparse voxel grid to store feature vectors, it retains the favorable attributes of traditional map representations such as flexibility and scalability, while also achieving continuous representation through interpolation, inheriting NeRF's high-quality photo-realistic reconstruction. Building upon this foundation, inverse rendering utilizes rendering errors to concurrently optimize camera poses and local maps, introducing novel approaches to visual odometry in the SLAM front-end. This tight coupling between the front and back ends revitalizes SLAM into an end-to-end, concise structure.

However, NeRF remains a weak point within the SLAM system. For current NeRF-based SLAM systems to achieve more accurate and robust camera pose estimation, map representation needs to be more explicit, possibly through substituting volume densities with signed distance functions. Yet, this approach may compromise NeRF's capability for novel view synthesis. Balancing this trade-off is a challenge that demands resolution. Additionally, current scenarios indicate that introducing NeRF into SLAM systems does not surpass traditional SLAM methods in terms of localization accuracy, raising questions about NeRF's necessity within the SLAM context. Addressing how to elevate NeRF-based SLAM system's localization precision beyond or at least on par with conventional visual odometry (VO), or how to harness NeRF's unique potential in VO, is a conundrum that researchers need to address.

Furthermore, this research direction offers ample room for exploration. The questions posed earlier, each step taken towards resolution, can yield significant research accomplishments. For instance, NeRF's distinctive novel view synthesis should naturally serve as a valuable tool for predicting camera motion. Extending NeRF's application to outdoor environments, such as the Kitti dataset, is also within reach, given the compatibility of sparse voxel grids. In a broader sense, migrating more mature SLAM methods to NeRF, or adapting NeRF maps to existing SLAM front-end techniques, holds potential for performance enhancement and theoretical integration. In summary, the realm of NeRF-based SLAM presents a promising avenue for research, with the potential for breakthroughs that can redefine the landscape of simultaneous localization and mapping.

DOI: 10.54254/2755-2721/37/20230484

References

[1] Mildenhall B, Srinivasan P P, Tancik M, Barron J T, Ramamoorthi R and Ng R 2020 NeRF. Communications of the ACM, 65, 99 - 106.

[2] Mur-Artal R and Tardós J D 2017 ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras. IEEE Transactions on Robotics. 33(5):1255-62.

[3] Sucar E, Liu S, Ortiz J and Davison A J 2021 iMAP: Implicit Mapping and Positioning in Real-Time. IEEE/CVF International Conference on Computer Vision (ICCV). 2021:6209-18.

[4] Zhu Z, Peng S, Larsson V, Xu W, Bao H and Cui Z, Oswald MR, Pollefeys M 2022 NICE-SLAM: Neural implicit scalable encoding for SLAM. IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022:12786-96.

[5] Barron J T, Mildenhall B, Verbin D, Srinivasan P P and Hedman P 2022 Mip-NeRF 360: Unbounded Anti-Aliased Neural Radiance Fields. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2022:5470-79.

[6] Barron J T, Mildenhall B, Tancik M, Hedman P, Martin-Brualla R and Srinivasan P P 2021 Mip-NeRF: A Multiscale Representation for Anti-Aliasing Neural Radiance Fields. IEEE/CVF International Conference on Computer Vision (ICCV). 2021:5855-64.

[7] Tancik M, Casser V, Yan X, Pradhan S, Mildenhall B, Srinivasan P P, Barron J T and Kretzschmar H 2022 Block-NeRF: Scalable large scene neural view synthesis. IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022:8248-58.

[8] Yang X, Li H, Zhai H, Ming Y, Liu Y and Zhang G 2023 Vox-Fusion: Dense tracking and mapping with voxel-based neural implicit representation. ArXiv. 2023;abs/2307.12008.

[9] Lisus D and Holmes C T 2023 Towards Open World NeRF-Based SLAM. ArXiv. 2023;abs/2301.03102.

[10] Rosinol A, Leonard J J and Carlone L 2022 NeRF-SLAM: Real-Time Dense Monocular SLAM with Neural Radiance Fields. ArXiv. 2022;abs/2210.13641.

[11] Zhu Z, Peng S, Larsson V, Cui Z, Oswald M R, Geiger A and Pollefeys M 2023 NICER-SLAM: Neural Implicit Scene Encoding for RGB SLAM. ArXiv. 2023;abs/2302.03594.

[12] Lin C H, Ma W C, Torralba A and Lucey S 2021 BARF: Bundle-Adjusting Neural Radiance Fields. IEEE/CVF International Conference on Computer Vision (ICCV). 2021:5741-51.

[13] Chung C M, Lee J, Kim J and Park J 2023 Orbeez-SLAM: A real-time monocular visual SLAM with ORB features and NeRF-realized mapping. IEEE International Conference on Robotics and Automation (ICRA). 2023:9400-06.

[14] Johari M M, Carta C and Fleuret F 2023 ESLAM: Efficient Dense SLAM System Based on Hybrid Representation of Signed Distance Fields. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2023:17408-19.

[15] Yuan Y and Nuechter A 2022 An algorithm for the SE(3)-transformation on neural implicit maps for remapping functions. arXiv preprint arXiv:2206.08712. 2022.

[16] Kuang H, Chen X, Guadagnino T, Zimmerman N, Behley J and Stachniss C 2022 IR-MCL: Implicit Representation-Based Online Global Localization. IEEE Robotics and Automation Letters. 2022;8:1627-34.