NeRF-Based SLAM

Subjects: Computer Science, Artificial Intelligence

Contributor: Kaiyun Yang , Yunqi Cheng , Zonghai Chen , Jikai Wang

Simultaneous Localization and Mapping (SLAM) systems have shown significant performance, accuracy, and efficiency gains, especially when Neural Radiance Fields (NeRFs) are implemented. NeRF-based SLAM in mapping aims to implicitly understand irregular environmental information using large-scale parameters of deep learning networks in a data-driven manner so that specific environmental information can be predicted from a given perspective. NeRF-based SLAM in tracking jointly optimizes camera pose and implicit scene network parameters through inverse rendering or combines VO and NeRF mapping to achieve real-time positioning and mapping.

SLAM NeRF robotics 3D reconstruction

1. NeRF-Based SLAM in Mapping

1.1. Map Representations

According to different structures and models of learning, implicit modeling methods are divided into fully implicit modeling methods based on multilayer perceptron, modeling methods based on high-dimensional feature networks, and modeling methods based on high-dimensional feature points. Modeling methods based on multi-layer perceptron are pure end-to-end learning methods that require regression learning of large-scale, highly non-linear geometric and texture information of the environment. These methods tend to cause challenges, including difficulty with effective convergence of the model, difficulty with timely and accurate updating, and insufficient generalizability across multiple environments. To address these issues, scholars have proposed modeling methods based on high-dimensional feature networks and modeling methods based on high-dimensional feature points. Modeling methods based on high-dimensional feature grids adopt a structured way to make the model better understand the spatial structure by gridding the environmental information. Modeling methods based on high-dimensional feature points are explicit and implicit hybrid strategies that improve modeling by focusing on important feature points. These methods are proposed to overcome the limitations of pure end-to-end modeling methods, make the model more robust and generalized, and better adapt to highly nonlinear geometric information and texture information in complex environments.

A. Implicit Representations

In order to achieve real-time mapping, iMAP [1] uses a single MLP network with a smaller network architecture. Meanwhile, in order to capture more geometric information, the 3D coordinates are upgraded to n-dimensional space by a Fourier feature network, which is used as the MLP network input. The color and volume density

obtained as decoded by the MLP network are used to jointly render the depth and color of the map. Also, to reduce computational consumption, only 200 points are sampled for each image in each iteration. In addition, rendering loss is used to actively sample areas that require higher detail and areas where mapping is not yet accurate. Since iMAP uses an MLP network, catastrophic forgetting is unavoidable. To solve this problem [1], Suar and Liu et al. used an incremental approach to select representative keyframes with information gain and to form a memory bank collection of selected keyframes, which was used to continuously optimize the map in the back-end. At the same time, the selection of keyframes is controlled by the normalized depth error to adapt to the change in the camera's distance from the object. However, the expression ability of a single MLP network is limited. In order to achieve real-time performance, rendering performance is sacrificed. And its failure to consider reflections causes some photometric errors.

In order to solve the problem that only 3D coordinates as inputs leads to poor generalization. DeepSDF [2] encodes the object's shape as a feature vector and combines the feature vector with the 3D coordinates as the network inputs. DeepSDF first randomly defines the feature vector of the object and then uses it as the network input to decode the SDF value. Finally, DeepSDF optimizes the feature vector by back-propagating the SDF value error. This auto-decoder method is more robust than the auto-encoder method. However, generalization by the auto-encoder is obviously better than that of the auto-decoder. In order to solve this problem, DeepSDF only uses part of the sampling points when performing feature vector inference and uses all of the sampling points when reconstructing the target object, and it updates the network weight by the back-propagation error.

B. Implicit Joint Representations using Voxels

The iMAP model is a fully implicit representation, but the training speed of this representation is slower than that of the traditional SLAM mapping methods, and the map is not scalable. The advantage of NeRF lies in its advanced rendering equations rather than MLP networks, so photo-level rendering of NeRF can be realized if the map supports rendering. Therefore, various researchers have proposed combining a traditional explicit network and an implicit network to get a new way of environmental representation. In this part, modeling methods based on high-dimensional feature networks are unfolded and analyzed. This involves dividing the map into single or multiple voxel grids with different resolutions and storing the feature vectors by using the displayed voxel grids. Then, the model decodes the feature vectors by using a perceptron network during rendering to obtain the SDF values and RGB values or occupancy rates and RGB values.

Inspired by Mip-NeRF 360 [3], which uses different MLPs to store the foreground and the background, NICE-SLAM represents the scene with a nested grid of voxels of three different resolutions: mid-level, fine-level, and coarse-level. Feature vectors are stored in the voxel grids, and the network ID is pretrained by trilinear interpolation. Four different MLP networks are applied to complete the map: mid-level is used to optimize the grid features; fine-level is used to capture smaller, high-frequency geometric details; and coarse-level is used to capture large-scale features, such as objects with geometric structures such as floors, and it is used to predict unobserved geometric features, thus giving the system predictive power for unseen perspectives; color-level stores color information to generate more detailed color representations in the scene, thereby improving the accuracy of the tracking thread.

Finally, the depth and color of the reconstructed map are obtained through joint rendering of the volume density and color. In order to solve the forgetting problem, keyframe selection follows the iMAP approach and is selected in an incremental way. Meanwhile, NICE-SLAM deletes pixels with high depths or dark colors during the mapping process: effectively ignoring dynamic objects and improving system robustness.

NICER-SLAM [4], as a successor to NICE-SLAM, does not require the input of RGB-D information: it only needs to be provided with RGB information. The voxels still follow the coarse–medium–fine three-layer voxel division of NICE-SLAM, but NICER-SLAM decodes out the SDF value instead of the volume density. Because the SDF value is better than the volume density for mapping, NICER-SLAM also introduces locally adaptive transformations that can dynamically adjust the smoothness of the SDF value in different regions. So it can better adapt to the geometric complexity of the map. The RGB observation alone suffers from serious ambiguity, so five kinds of losses including depth loss, normal vector loss, and optical flow loss are fused to improve the mapping quality. However, due to the complexity of the loss function used, although the mapping effect is better than that of the original NICE-SLAM, the real-time performance is greatly reduced. And it does not solve the most serious localization problem of NICE-SLAM: there remains more lifting space.

Although the rendering speed of NICE-SLAM is greatly improved compared to that of iMAP, its dense voxel grid is pre-allocated: it still cannot realize expansion of the map and is not suitable for large outdoor scenes. Moreover, NICE-SLAM uses a pretrained geometry decoder, which greatly reduces its generalization ability. To address this problem, Vox-Fusion [5] dynamically allocates new voxels by using an explicit octree structure and encodes the voxel coordinates by Morton coding to improve the voxel retrieval speed. Thus, the system can incrementally expand the implicit scene to complete a mapping of large outdoor scenes. In contrast to iMAP and NICE-SLAM, which use MLP networks to decode voxel density, Vox-Fusion uses feature embedding as the MLP network input, directly decodes the SDF values, and renders a map with the SDF value. SDF values can provide richer local geometric information about surfaces as well as distance information, which can support light tracing to improve the rendering quality and geometric accuracy of the scene. Light tracing can be used for high-precision collision detection and to create various visual effects, and thus, it is widely used in VR, AR, and game development. Although it has been experimentally proven that SDF values are better for mapping, they also lose the rendering advantage brought by the volume density and lose the ability to fit some new perspectives.

Wang et al. [6] proposed a neural RGB-D SLAM system, Co-SLAM, based on a hybrid representation. Co-SLAM proposed loss functions with depths, colors, SDF values, and feature smoothness in order to realize the supervision of accurate and smooth mapping scenes. These loss functions help the model to better adapt to the geometric and color features of the scene in the training process. Additional sampling near the surface points speeds up the convergence of the network. High-fidelity reproduction of maps based on coordinate representations is possible due to the continuity and smoothness of an MLP network. But the inherent limitations of MLP often lead to slower convergence and catastrophic forgetting when used. Real-time mapping for SLAM cannot be achieved.

The current implicit network representation method has achieved better results, but it still cannot effectively deal with poor lighting conditions and large-scale scenes. To solve the occlusion problem and to supervise sampling the

points behind objects, Yan and Shi et al. [7] introduced the concept of generalized thickness for modeling, which regards the generalized thickness as a random variable. The probability of each point on the light line to be occupied is derived by applying a prior on the generalized thickness. This method can supervise directly in 3D space without 2D rendering. Binary cross-entropy loss is applied to the occupancy function and uncertainty factors are considered in the binary cross-entropy loss so that the model can deal with complex scenes more robustly. However, required manual adjustment to the generalized thickness prior causes difficulty with generalization; this problem needs to be solved by introducing a learnable prior in the later stage.

In global sampling, the majority of points fall in free space. A large number of invalid points are generated at the beginning of sampling, which makes network convergence slow. In order to accelerate the training speed, Shi and Yang et al. [8] proposed a mapping method based on a three-layer sampling strategy. In addition to global sampling, local sampling is introduced. However, the sampling effects of local sampling and global sampling are basically the same as the number of iterations increases, which leads to a lack of surface information. Therefore, near-ground sampling is added to emphasize the penalty of noise near the surface. In addition to this, to adapt to scene changes, Shi and Yang et al. also estimate a dynamic boundary. To trade-off between point cloud density and computational efficiency, keyframes are selected every three frames, while to solve the forgetting problem, 75% of the points in the previous keyframes are selected in each iteration, and 25% of the points from the latest keyframes are selected for the network update.

Isaacson and Kuang et al. [9] improve mapping accuracy by introducing a novel dynamic edge loss function that combines depth loss and sky loss. The dynamic edge loss function is based on Jensen–Shannon divergence, which assigns unique edges to each LiDAR ray to improve training convergence speed and mapping accuracy. This loss function uses dynamic edge sizes by measuring the differences between the learning degrees for different map regions, allowing the system to retain and refine the learned geometric information while learning new regions. The JS dynamic margin uses a larger margin for rays pointing toward regions of the map with unknown geometries while using a smaller margin for rays pointing toward well-learned regions. In addition, LONER uses depth loss to measure the error between the rendered depth and the LiDAR-measured depth. And it introduces sky loss to force the weight of the rays pointing to the sky to be zero.

NICE-SLAM uses three voxel grids with different resolutions to represent the scene. Zhong and Pan et al. [10] stitched and merged the eigenvalues of voxel grids with different resolutions stored in octree nodes after trilinear interpolation, which improved the modeling effect for different spatial resolutions. The fused feature values are input into an MLP network to decode the SDF values of the corresponding points, thus better capturing the geometry of the scene. To solve the problem of catastrophic forgetting brought by MLP networks. SHINE-Mapping limits the updates to the weights by adding regular terms to the loss function: that is, each iteration only updates the weight values that have less impact on the previously learned frames to ensure that the current update does not have a significant impact on the previously modeled region. This improves the model's ability to retain historical knowledge during incremental mapping and reduces the risk of forgetting previous data.

Liu and Chen et al. [11] introduced local maps and global maps. The size of a local map is set according to the sensor's range and the size of the task space. The model also uses an independent encoder and decoder. A freeze–activate mechanism is used to transfer submaps between the system memory and video memory for real-time training on large-scale scenes. The sigmoid function is used to map the SDF values to the range (0, 1) to cope with the effects of noise and sensor errors. Eikonal regularization is also introduced to obtain an accurate and continuous signed distance field, especially in regions far away from the object to avoid over-smoothing. To avoid catastrophic forgetting, local maps are used to accumulate historical input points, i.e., points retained only within the scope of the local map. And downsampling is performed when the number of historical points exceeds the threshold. After a certain number of frames of training, the decoder parameters are fixed to prevent inconsistency in the decoder parameters over time. However, parameter fixation can solve the catastrophic forgetting problem to a certain extent, but historical information loss and blurring may still occur with long-time mapping, and MLP network parameter fixation cause a decline in generalization ability.

Yu and Liu et al. [12] used the same map representation method as Vox-Fusion and called it the Neural Feature Volume. In order to effectively estimate surfaces in a scene in the early stages of training, NF-Atlas introduces a differentiable range approximation method. A SLAM method is established by combining all measurement models (such as range measurements, SDF measurements, and semantic measurements) into a maximum a posteriori problem. The map can be efficiently constructed and regularized by different priors.

Li and Zhao et al. [13] combine a discriminative model and a generative model. The discriminative model uses sparse convolution to extract the shape prior, while the generative model uses an MLP network to decode the SDF values for subsequent map rendering. This hybrid structure improves the flexibility and performance of the model. To improve the accuracy of the decoded SDF value, the Eikonal equation constraint, normal vector constraint, function value constraint, and off-plane point constraint are combined. And a training method based on a loss function is used to optimize the network parameters by minimizing the loss function. The LODE method also demonstrates adaptability to semantic extensions and can be extended to implicit semantic completion problems in two ways. This further extends the applicability of the method to different application scenarios. In contrast, Wiesamann and Guadagnino et al. [14] approximated the direction to the nearest surface by using gradient information, and they estimated the distance to the nearest surface through direction projection. A weight strategy is introduced to prioritize nearby surface points, and additional loss is added to ensure that the sampling points are located on the surface. NeRF-SLAM [15] uses dense depth maps as inputs to optimize the parameters of the neural volume and the camera pose. Combined with the uncertainty of a dense depth map, a depth loss function for weighted depth loss is proposed to reduce the bias during map construction due to noise.

NeRF-LOAM [16] adopts the octree form and recursively divides voxels into leaf nodes. Meanwhile, a new loss term is introduced to distinguish surface SDF values from non-surface SDF values, which is more suitable for the outdoor environment of SLAM. In terms of sampling point selection, the near-surface points on a ray that intersects the currently selected voxel are prioritized, which accelerates the convergence speed of the network. To avoid the catastrophic forgetting problem caused by MLP networks, Deng and Chen et al. added a keyframe buffer to selectively add keyframes. GO-SLAM [17] aims to provide real-time mapping: that is, fast rendering of the

reconstructed scene and ensuring that the mapping maintains global consistency after updating. In order to achieve this goal, a keyframe selection strategy is introduced to sort keyframes. According to the pose differences between them, the model prioritizes the keyframes for which the pose difference is the largest and keeps two of the most delicate keyframes and unoptimized keyframes, which can be efficiently updated and reconstructed to avoid excessive computational overhead.

The summary of NeRF-based SLAM methods are shown in **Table 1**.

Table 1. Summary of NeRF-based SLAM methods.

Method Name	Year	Utilized Sensors			Decoded Parameters		
		RGB-D	RGB	LiDAR	SDF	Density	Color
NICE-SLAM [18]	2022	✓				✓	✓
Vox-Fusion [5]	2022	✓			✓		✓
NICER-SLAM [4]	2023		✓		✓		✓
Co-SLAM [6]	2023	✓			✓		✓
LONER [9]	2023			✓		✓	✓
Shine-mapping [10]	2023			✓	✓		✓
NF-Atlas [12]	2023			✓	✓		✓
LODE [13]	2023			✓	✓		✓
NeRF-LOAM [16]	2023			✓	✓		✓
LocNDF [18]	2023			✓	✓		✓

C. Implicit Joint Representations Using Points

Although voxel-grid-based methods can recover high-quality maps and textures, they require a large number of sampling points, which inevitably leads to slow training convergence and affects the real-time performance of the system. Point-SLAM introduces the concept of neural point clouds and defines a set of neural point clouds, in which the location information, geometrical features, and color features are stored. A point addition strategy for dynamic point density is adopted. The search radius changes according to the color gradient, and the compression level and memory usage are controlled to achieve higher point density in areas that require detailed modeling and lower point density in areas with less detailed information. This strategy flexibly explores the scene by gradually increasing the neural point cloud without specifying the scene boundary in advance, which improves the perception and robustness of modeling. Compared with the traditional voxel-based method, it does not have to consider the blank regions between the camera and object surfaces, has fewer sampling points, and converges faster, which

makes it suitable for online scene mapping. The depth information is synthesized through a combination of uniform sampling in the image plane and gradient-driven sampling, and the neural point cloud is updated based on deep camera noise features.

1.2. Map Encoding

A. Parametric Encoding

Parametric encoding aims at arranging additional trainable parameters in the auxiliary data structure and finding and interpolating these parameters according to the input vectors $x \in R^d$. Its encoding trades a larger memory footprint in exchange for smaller computational cost. Both NeRF-SLAM and SHINE-Mapping use the same encoding method as instant-NGP. Feature vectors are stored in a compact spatial hash table that does not depend on a priori knowledge of the scene geometry. The feature values interpolated by voxels of different resolutions are fused and are then used as the MLP network input to decode the SDF value. This approach yields a greater degree of adaptability compared to traditional parameter encoding.

In order to solve the problem of low accuracy in small instance mapping, Shi and Yang et al. encode shapes by introducing potential vectors: expressing the process of instance mapping by probabilistic inference. And they use the obtained shape coding and 3D coordinate series as input. SDF values are obtained by decoding, which makes the surface of the reconstructed instance smoother.

B. Frequency Encoding

Taking iMAP as an example, an implicit network representation uses sinusoidal or other types frequency embedding to map the coordinates of the input points to high-dimensional space in order to capture high-frequency details that are essential for high-fidelity geometric mapping. The iMAP model improves the 3D coordinates in n-dimensional space $\sin(\beta p)$ by means of the Gaussian positional embedding method proposed in the Fourier feature network. In addition to connecting this representation as the network input, it is also connected to the activation layer of the network, and allows optimization of the embedding matrix B. It is implemented as a single fully connected layer with sinusoidal activation. NICE-SLAM employs the same strategy for encoding, using different frequencies to map the representation into voxel grids with different resolutions.

C. Mixture Encoding

To improve the training speed, many researchers have used acceleration methods such as instant-NGP to improve the performance of MLP itself. Instant-NPG is fast to train, but it is discontinuous at many places in space because it uses hash coding. While methods based on parameter coding improve computational efficiency, they lack the ability to fill in holes and have poor smoothness. To solve this problem, Co-SLAM proposed a coding method that combines coordinate coding and parametric coding and introduces one-blob coding into traditional parametric coding. One-blob coding and multi-resolution hash coding are input into the geometry decoder to obtain SDF

values and feature vectors, and the decoded feature vectors and one-blob coding are input into the color decoder to obtain RGB values.

2. NeRF-Based-SLAM in Tracking

NeRF-based SLAM can be divided into two main methods in the tracking stages: One uses inverse rendering of NeRF to jointly optimize the camera pose and network parameters through photometric loss. The other uses traditional visual odometry as the front-end, while NeRF mapping is the back-end, and the front and back are decoupled for joint optimization.

2.1. The Method of Inverting NeRF

Both iMAP and NICE-SLAM use a tracking approach similar to iNeRF. They use inverse rendering to self supervise. By implementing two processes in parallel, the pose of the latest frame is optimized at a higher frequency than joint optimization, which helps to optimize small displacements to the camera more robustly. A modified geometric loss function is used to improve the robustness of tracking based on the line-of-sight overlap between the current frame and the keyframe. A coarse feature grid can be divided across previously unseen regions, allowing effective tracking even when most of the region is unseen. In order to deal with huge redundancies in video images, representative keyframes with information gain are selected incrementally. At the same time, the selection of keyframes is controlled by a normalized depth error to accommodate for variations to the camera's distance from the object. However, inverse rendering processes are sensitive to the initial pose. When the pose deviation is large, the mapping effect is greatly reduced. Therefore, how to improve the accuracy of the initial pose when applying NeRF inverse rendering is still a major difficulty.

In large-scale scenarios, Yu and Liu et al. use pose maps to generate multiple neural feature volumes as nodes. By using the edge between nodes to represent the relative pose between adjacent volumes, an elastic neural feature field is established. An incremental mapping strategy is adopted to construct neural feature volumes progressively through a series of poses and measurements. The initial pose of each neural feature volume is fixed, and as the map is built, past neural feature volumes are frozen and new volumes are gradually initialized. NF-Altas assures that the local region of the map is captured efficiently and limits the computational complexity. Compared with existing methods, the method proposed by NF-Altas does not need to be reconstructed after loop detection. It only needs to refine the initial pose of the neural feature volume to match the trajectory of the updated robot. Meanwhile, loop detection updating is based on volume measurements and uses NeRF inverse rendering to guide pose estimation, which improves the robustness of the system for tracking in large-scale scenarios. In addition, NF-Altas supports on-demand global mapping to extract maps from multiple neural feature volumes, enables flexible and efficient access to the global map, and avoids global map parameter updates.

2.2. The Method of VO

Orbeez-SLAM [19] follows the tracking strategy of ORB-SLAM2 and uses feature matching to obtain the camera pose and to optimize pose estimation by minimizing the reprojection error. After a triangulation step of the visual odometer, new map points are added to the local map, and BA is used to further minimize the reprojection error. The consistency of the map is improved by optimizing the pose of the keyframe and the settable map points. To further speed up the rendering process, the concept of density is introduced, while robustness to surfaces is improved by storing the number of samples per voxel. To reduce noise, only the points within voxels that are frequently scanned by light are measured by triangulation so as to ensure the reliability of the map. Chun and Tseng et al. continued ORB-SLAM2, took new keyframes, stored sparse point cloud maps from the mapping thread, utilized point cloud sparsity to improve NeRF sampling efficiency, sampled near sparse map points, and used a voxel skipping strategy to improve network convergence speed. This method is equivalent to using traditional visual odometry as the front-end and the NeRF map as the back-end, decoupling the front- and back-ends, and combining the methods on both sides. Specifically, it generates a sparse point cloud by ORB-SLAM2 and samples the sparse point cloud. Then it uses a voxel skipping strategy to decode the voxel, using an implicit network to get the color information for rendering. Although excellent tracking effects and better rendering quality can be achieved at the same time, NeRF plays a limited role in tracking, and the front- and back-ends are not well integrated.

Although iMAP and NICE-SLAM use inverse rendering for joint optimization of network parameters and camera poses, they are not accurate enough due to the lack of loop detection and BA. Although Orbeez-SLAM applies traditional loop detection to improve tracking accuracy, it cannot update the scene representation after loop detection. To solve the above problems, the three parallel processes of NEGL-SLAM (tracking, dynamic local map, and loop closure) ensure high-speed training and fast response to loops, enabling the system to meet the low latency requirements of practical applications. NEGL-SLAM [20] follows the ORB-SLAM3 tracking strategy and represents the whole scene with multiple local maps, avoiding the need to retrain the whole scene representation in the single volumetric implicit neural method, for which the time consumption of retraining the whole scene representation is required. NEGL-SLAM also performs global BA during loop detection. In order to avoid the trajectory jump problem caused by global BA in traditional methods, after global BA, the model undergoes two-stage optimization. The first stage corrects the errors between local maps in real time, and the second stage eliminates small errors in sub-real-time optimization and improves the accuracy of scene representation.

GO-SLAM uses the RAFT [21] algorithm for optical flow computation; RAFT can be used to process monocular, binocular, or RGB-D camera inputs. Based on the average values of optical flow calculations, new keyframe initializations for front-end tracking are implemented. And a local keyframe map is established for loop detection by selecting high common-view keyframe connections. An efficient connection between local keyframes is realized by using common-view matrix and optical flow computation. In addition to this, Zhang and Tosi et al. run global BA in a separate thread and use global geometric features to reduce the real-time requirements of global BA, making it more efficient for processing tens of thousands of input frames. By establishing a global keyframe map, online global BA is realized, and the global consistency of the trajectory is improved.

NeRF-SLAM uses Droid-SLAM [22] for tracking. It uses an architecture similar to that of RAFT to solve the optical flow between frames: generating a new optical flow and weight for each optical flow measurement. The BA problem is then solved by densifying the optical flow and weight and representing the 3D geometry of each keyframe with an inverse depth map: transforming the problem into a linear least squares problem. Using block partitions based on Hessian matrixes, the edge covariances of dense depth maps and poses are calculated to provide estimations of depth and pose uncertainty.

LocNDF uses a learned NDF to achieve accurate registration of point clouds to maps through nonlinear least squares optimization without searching for corresponding points for ICP optimization. With the obtained movement direction and distance, the robot moves directly in the direction without searching for corresponding points, which simplifies the traditional ICP method. Global positioning in NDF is achieved using MCL positioning. A particle filter is used to estimate the robot's pose through a motion model and an observation model, where the observation model is based on the distance between the measured point cloud and the NDF.

3. Loss Function

Eikonial loss: The eikonal loss is a constraint on the gradient that requires the second derivative of the gradient to be equal to one, which can ensure the rationality of the deformation space.

$$\mathcal{L}_e = -\frac{1}{N}\sum_i (\| \frac{\partial f_\theta(p_i)}{\partial p_i} \| - 1)^2.$$

Photometric loss: The photometric loss is the L1-norm between the rendered and measured color values.

$$\mathcal{L}_p = \frac{1}{M}\sum_{i=1}^{W}\sum_{(u,v)\in s_i} \left| I_i[u,v] - \hat{I}_i[u,v] \right|,$$

where I_i is the predicted color, $\hat{I}_i$ is the true color, and u,v is the corresponding pixel on the image plane.

Geometric loss: The geometric loss measures the depth difference.

$$\mathcal{L}_d = \frac{1}{|R_d|}\sum_{r\in R_d}\left(\hat{d}_r - D[u,v]\right)^2,$$

where D_i is the predicted depth value, D_i is the true depth value, and u,v is the corresponding pixel on the image plane.

References

1. Sucar, E.; Liu, S.; Ortiz, J.; Davison, A.J. iMAP: Implicit mapping and positioning in real-time. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 6229–6238.
2. Park, J.J.; Florence, P.; Straub, J.; Newcombe, R.; Lovegrove, S. Deepsdf: Learning continuous signed distance functions for shape representation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 165–174.
3. Barron, J.T.; Mildenhall, B.; Verbin, D.; Srinivasan, P.P.; Hedman, P. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 5470–5479.
4. Zhu, Z.; Peng, S.; Larsson, V.; Cui, Z.; Oswald, M.R.; Geiger, A.; Pollefeys, M. Nicer-slam: Neural implicit scene encoding for rgb slam. arXiv 2023, arXiv:2302.03594.
5. Yang, X.; Li, H.; Zhai, H.; Ming, Y.; Liu, Y.; Zhang, G. Vox-Fusion: Dense tracking and mapping with voxel-based neural implicit representation. In Proceedings of the 2022 IEEE International Symposium on Mixed and Augmented Reality (ISMAR), Singapore, 17–21 October 2022; pp. 499–507.
6. Wang, H.; Wang, J.; Agapito, L. Co-SLAM: Joint Coordinate and Sparse Parametric Encodings for Neural Real-Time SLAM. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, AL, Canada, 18–22 June 2023; pp. 13293–13302.
7. Yan, D.; Lyu, X.; Shi, J.; Lin, Y. Efficient Implicit Neural Reconstruction Using LiDAR. arXiv 2023, arXiv:2302.14363.
8. Shi, Y.; Yang, R.; Wu, Z.; Li, P.; Liu, C.; Zhao, H.; Zhou, G. City-scale continual neural semantic mapping with three-layer sampling and panoptic representation. Knowl.-Based Syst. 2024, 284, 111145.
9. Isaacson, S.; Kung, P.C.; Ramanagopal, M.; Vasudevan, R.; Skinner, K.A. LONER: LiDAR Only Neural Representations for Real-Time SLAM. IEEE Robot. Autom. Lett. 2023, 8, 8042–8049.
10. Zhong, X.; Pan, Y.; Behley, J.; Stachniss, C. Shine-mapping: Large-scale 3d mapping using sparse hierarchical implicit neural representations. In Proceedings of the 2023 IEEE International Conference on Robotics and Automation (ICRA), London, UK, 29 May–2 June 2023; pp. 8371–8377.
11. Liu, J.; Chen, H. Towards Real-time Scalable Dense Mapping using Robot-centric Implicit Representation. arXiv 2023, arXiv:2306.10472.

12. Yu, X.; Liu, Y.; Mao, S.; Zhou, S.; Xiong, R.; Liao, Y.; Wang, Y. NF-Atlas: Multi-Volume Neural Feature Fields for Large Scale LiDAR Mapping. arXiv 2023, arXiv:2304.04624.
13. Li, P.; Zhao, R.; Shi, Y.; Zhao, H.; Yuan, J.; Zhou, G.; Zhang, Y.Q. Lode: Locally conditioned eikonal implicit scene completion from sparse lidar. arXiv 2023, arXiv:2302.14052.
14. Wiesmann, L.; Guadagnino, T.; Vizzo, I.; Zimmerman, N.; Pan, Y.; Kuang, H.; Behley, J.; Stachniss, C. Locndf: Neural distance field mapping for robot localization. IEEE Robot. Autom. Lett. 2023, 8, 4999–5006.
15. Rosinol, A.; Leonard, J.J.; Carlone, L. Nerf-slam: Real-time dense monocular slam with neural radiance fields. In Proceedings of the 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Detroit, MI, USA, 1–5 October 2023; pp. 3437–3444.
16. Deng, J.; Wu, Q.; Chen, X.; Xia, S.; Sun, Z.; Liu, G.; Yu, W.; Pei, L. Nerf-loam: Neural implicit representation for large-scale incremental lidar odometry and mapping. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Paris, France, 2–3 October 2023; pp. 8218–8227.
17. Zhang, Y.; Tosi, F.; Mattoccia, S.; Poggi, M. Go-slam: Global optimization for consistent 3D instant reconstruction. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Paris, France, 2–3 October 2023; pp. 3727–3737.
18. Zhu, Z.; Peng, S.; Larsson, V.; Xu, W.; Bao, H.; Cui, Z.; Oswald, M.R.; Pollefeys, M. Nice-slam: Neural implicit scalable encoding for slam. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 12786–12796.
19. Chung, C.M.; Tseng, Y.C.; Hsu, Y.C.; Shi, X.Q.; Hua, Y.H.; Yeh, J.F.; Chen, W.C.; Chen, Y.T.; Hsu, W.H. Orbeez-slam: A real-time monocular visual slam with orb features and nerf-realized mapping. In Proceedings of the 2023 IEEE International Conference on Robotics and Automation (ICRA), London, UK, 29 May–2 June 2023; pp. 9400–9406.
20. Mao, Y.; Yu, X.; Wang, K.; Wang, Y.; Xiong, R.; Liao, Y. NGEL-SLAM: Neural Implicit Representation-based Global Consistent Low-Latency SLAM System. arXiv 2023, arXiv:2311.09525.
21. Moad, G.; Rizzardo, E.; Thang, S.H. Living radical polymerization by the RAFT process. Aust. J. Chem. 2005, 58, 379–410.
22. Teed, Z.; Deng, J. Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras. Adv. Neural Inf. Process. Syst. 2021, 34, 16558–16569.