World Electric Vehicle Journal

MDPI

Review

SLAM Meets NeRF: A Survey of Implicit SLAM Methods

Kaiyun Yang, Yunqi Cheng, Zonghai Chen and Jikai Wang *

Department of Automation, University of Science and Technology of China, Hefei 230027, China; yangkaiyun@mail.ustc.edu.cn (K.Y.); chengyunqi@mail.ustc.edu.cn (Y.C.); chenzh@ustc.edu.cn (Z.C.)
* Correspondence: wangjk@ustc.edu.cn

Abstract: In recent years, 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. This paper firstly analyzes the current situation of NeRF and SLAM systems and then introduces the state-of-the-art in NeRF-based SLAM. In addition, datasets and system evaluation methods used by NeRF-based SLAM are introduced. In the end, current issues and future work are analyzed. Based on an investigation of 30 related research articles, this paper provides in-depth insight into the innovation of SLAM and NeRF methods and provides a useful reference for future research.

Keywords: SLAM; NeRF; robotics; 3D reconstruction

Citation: Yang, K.; Cheng, Y.; Chen, Z.; Wang, J. SLAM Meets NeRF: A Survey of Implicit SLAM Methods. *World Electr. Veh. J.* **2024**, *15*, 85. https://doi.org/10.3390/wevj15030085

Academic Editor: Joeri Van Mierlo

Received: 26 January 2024
Revised: 20 February 2024
Accepted: 23 February 2024
Published: 26 February 2024

1. Introduction

Simultaneous Localization and Mapping (SLAM) is a research field aimed at tracking and mapping [1]. It can be applied to domestic robots [2], unmanned aerial vehicles [3], AR [4], and unmanned ground vehicles [5]. SLAM enables mobile devices and robots to achieve autonomous navigation and map construction in unknown environments. It allows location determination in the absence of GPS signals, which promotes progress in the fields of autonomous driving, service robots, virtual reality, and more. By simultaneously constructing maps and updating its own poses in real time, SLAM provides autonomy and perception capabilities for various application scenarios and promotes the continuous development of artificial intelligence and robotics.

Traditional SLAM can be divided into LiDAR SLAM and visual SLAM. SLAM consists of data processing, front-end registration, back-end optimization, loop detection, and mapping [6]. Although current SLAM systems have gradually transitioned from single-sensor to multi-sensor fusion systems, their performance is still heavily influenced by sensor noise, which is hard to model and handle [7]. SLAM systems have strong dependence on the perceptibility of the environment, especially in the case of insufficient illumination and texture-less areas, which tend to lead to system performance degradation. Therefore, it is necessary to apply deep learning modules to SLAM systems. The combination of deep-learning-based odometry, environmental feature extraction and matching, and feature coding within the traditional SLAM framework greatly improves the processing ability of SLAM systems for sensor observations and their performance in complex scenes.

One of the key challenges of SLAM is how to retrieve and apply environmental information in real time [8]. The current SLAM methods have defects in new viewpoint synthesis, which brings great inconvenience to mapping and map reuse. To tackle this problem and with the development of the NeRF method in the field of 3D reconstruction,

combining NeRF with SLAM systems to construct more powerful and available SLAM systems has become a frontier development hotspot in the field of SLAM in recent years. The Neural Radiance Field (NeRF) [9] model is a method of new perspective synthesis that implicitly represents the scene through a multi-layer perceptron and applies a classical volume rendering function to reconstruct the scene. The mechanism of NeRF is the result of the integration of end-to-end learning and the core theoretical methods of computer graphics. Since they are different paradigms, constructing novel SLAM systems based on NeRF frameworks is challenging.

Figure 1 shows classification by a NeRF-based SLAM system. The SLAM system mainly includes two parallel threads: namely, tracking and mapping [10]. This paper classifies different methods and strategies used by each thread based on NeRF-based SLAM. In terms of mapping, NeRF-based SLAM introduces neural implicit environment representation and volume rendering functions. Therefore, we divide the methods into two categories: implicit and hybrid expression. The fully implicit expression method mainly relies on neural implicit environment representation mapping. For example, iMAP uses an MLP network to store the map and decodes the color and volume density of the spatial points in real time. However, the rendering and training speed of the fully implicit expression method is slow, and the map cannot be expanded and adapted to large environments. The key to continuous scene expression is that the map can be rendered and differentiated. Therefore, many researchers combine an explicit expression method with an implicit expression method, which is called hybrid expression. For example, NICE-SLAM [11] and Co-SLAM [12] use voxel grids to explicitly store feature vectors. Point-SLAM [13] defines a set of neural point clouds in which to store location information, geometric features, and color features. The feature vector is extracted from the voxel or point cloud during rendering, and the parameters required for rendering are obtained by MLP decoding. NeRF-based SLAM mainly uses inverse rendering or combines NeRF mapping with visual odometry (VO). Inverse rendering is the inverse process of NeRF, and the camera pose and MLP network parameters are jointly optimized by calculating the photometric loss. The tracking method of VO, such as Orbeez-SLAM, uses VO as the front-end and the NeRF map as the back-end. The front-end and back-end are decoupled, and the two ends are strongly combined to achieve good tracking effect. For methods using inverse NeRF, such as iMAP, the trained NeRF model back-propagation error is used to update the pose and minimize the photometric error. NeRF-based SLAM systems have been able to achieve high-fidelity mapping of some scenes. However, how to use mixed scene expressions to realize dynamic scene modeling and solve the catastrophic forgetting problem caused by MLPs in deeper levels is a key challenge to the development of NeRF-based SLAM systems.

This paper investigates 30 articles on NeRF-based SLAM and divides them into different categories according to the contribution of NeRF with regard to different aspects of SLAM. The authors hope their work will present a reference for researchers working to improve NeRF-based SLAM techniques. The rest of this paper is composed as follows: Section 2 introduces and discusses other state-of-the-art papers in the field of SLAM and NeRF. Section 3 describes the improvement of the performance of SLAM systems by adding NeRF. Section 4 introduces the main datasets used and the evaluation metrics of the system. Unsolved problems and potential trends in this field are discussed in Section 5. And in Section 6, the conclusion to this paper is drawn.

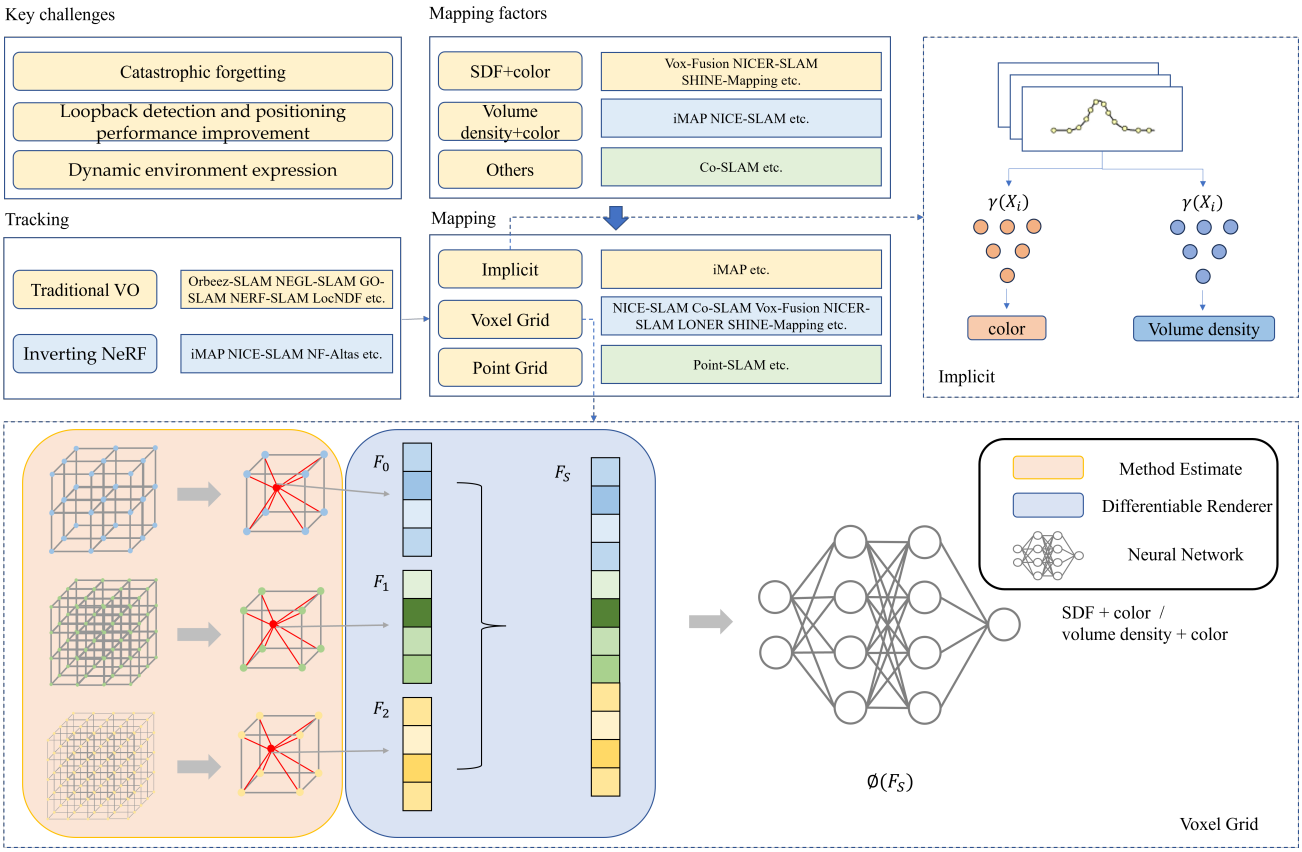

Figure 1. Classification of NeRF-based SLAM systems. Implicit example: using joint encoding ($\gamma(X_i)$ and $\gamma(X_i)$), input coordinates are mapped to color and volume density via two shallow MLPs. Voxel Grid example: the interpolation features F_0, F_1, and F_2 of the query point in each level with features are obtained by trilinear interpolation first, and they are spliced into F_s. The SDF value, color or volume density, and color are obtained by MLP ϕ regression.

2. Evolution of SLAM and NeRF

2.1. Evolution of SLAM Algorithms

The first system to implement real-time monocular VSLAM was proposed by Davsion et al. in 2007 [14]. Mono-SLAM introduces the SFM method into SLAM to create sparse but persistent maps under its framework. A map reconstructed by this method only includes landmarks but does not provide detailed information on the mapping area. In the same year, Klein et al. proposed a parallel tracking and mapping method: PTAM. It divides the VSLAM system into two threads: tracking and mapping. This multi-threaded baseline has brought inspiration and recognition to much subsequent work. The main idea of PTAM [15] is to reduce the computational cost and use parallel processing to improve the real-time performance of VSLAM systems. PTAM is also the first method to use bundle adjustment (BA) to jointly optimize camera pose and map features. In 2013, Salas-Moreno et al. proposed SLAM++ [16], which is the first method that fuses semantic information to a real-time SLAM framework. SLAM++ uses an RGB-D camera to collect information and generates a camera pose map by estimating and tracking the camera pose. The semantic information and relative pose estimation collected for the scene provide constraints for mapping, which is conducive to updating and optimizing the map. However, its framework requires structured priors and needs to be provided with a large number of geometric models, so its generalization is poor. In the same year, Endres et al. proposed a method based on an RGB-D camera [17]. The front-end processes the sensor data to obtain the geometric relationships and uses the RGB-D data to create a three-dimensional probability occupancy map that be effectively used for navigation tasks.

With the ripening of the VSLAM baseline, it became necessary to improve the real-time performance and mapping accuracy of systems. Forster et al. proposed a semi-direct VO known as semi-direct visual odometry (SVO) [18] in 2014. SVO combines the advantages of methods based on feature points and direct methods. It also proposes an attitude refinement module to minimize the reprojection error. But SVO cannot perform loop detection. In 2014, Engle et al. proposed an LSD-SLAM (Large-Scale Direct SLAM) [19] method that is compatible with monocular cameras and binocular cameras. LSD-SLAM is a direct SLAM method based on optical flow tracking, and it realizes semi-dense mapping and large-scale mapping and can run online in real time. The authors creatively proposed a relationship between the pixel gradient method and the direct method. The authors also used this relationship to achieve semi-dense mapping. With the development of deep learning, methods based on deep neural networks can provide higher recognition rates. In 2017, Tateno et al. proposed a method based on a convolutional neural network (CNN) [20]: titled CNN-SLAM. This method divides the map into smaller parts and performs depth prediction when estimating the pose between two frames. Compared with a monocular camera, CNN-SLAM based on a neural network can perform absolute-scale depth prediction.

Mur-Artal et al. proposed an indirect VSLAM method, ORB-SLAM [21], in 2015, and proposed ORB-SLAM2 [22] and ORB-SLAM3 [23] after optimizing its framework. The ORB-SLAM method performs high-precision positioning by directional FAST and rotating Brief features. Based on the PTAM framework, ORB-SLAM adds map initialization and loop detection functions and optimizes the keyframe selection strategy. ORB-SLAM2 runs three threads in parallel. For keyframe selection, it finds all of the keyframes through a common view for BA and establishes a local map. After detecting the loop, the pose is optimized with a similarity degree constraint to obtain a globally consistent map. ORB-SLAM2 also adds an additional global BA after the loop thread, which is only called when a loop occurs. ORB-SLAM3 is modified for situations during which the tracking thread maybe lost. ORB-SLAM3 uses ATLAS [24] to store multiple trivial maps that are matched to all previous small maps when the tracking thread is lost. If the matching is successful, the tracking thread continues.

Although the current visual SLAM has developed rapidly for positioning, it still has major deficiencies in environment modeling. It can only build simple feature maps, grid maps, or dense network maps but lacks effective methods and tools for more efficient retrieval, acquisition, and inference of environmental information.

2.2. Evolution of NeRF Algorithms

The NeRF system was first proposed in 2020; it selects a series of images with known poses and samples points on pixel rays. By combining the 3D coordinate values of the sampling point with the 2D viewing direction, a continuous 5D function is used to represent a static scene. By adding the observation direction, the model can better represent specular reflection. The 5D vector value function is used as the MLP network input to decode the corresponding volume density and direction of the emitted color, and NeRF implicitly represents the scene in the network without optimizing the camera position. NeRF uses a classical volume rendering function combined with a hierarchical sampling method to radially integrate the points on the rays, enabling the rendering of the color of any ray that passes through the scene and, thus, representation of the continuous scene.

The iNeRF [25] model is the first work that proposed a NeRF model for pose estimation. It proposes a framework based on inverting the neural radiance fields for meshless pose estimation based on a NeRF model that has been built in advance. The iNeRF model is essentially the inverse process of NeRF and uses the RGB image, the initial pose estimation, and the NeRF model as inputs. By calculating the photometric loss between the pixels of the NeRF-rendered image and the pixels of the real image, the pose is updated by the NeRF model back-propagation error to minimize the photometric error. Considering that the loss of all pixels in the calculated and back-propagated images reduce the real-time

performance of the network, a sampling strategy based on the region of interest is proposed to optimize the sampling ray and the sampling points above it to better optimize the pose. The new image of the estimated camera pose is added to the NeRF training set, allowing NeRF to train in a semi-supervised environment. Inspired by the two methods of the NeRF fixed-pose optimization network model and the iNeRF fixed-pose optimization network model, BARF [26] optimizes both the network model and the pose. BARF introduces image alignment theory into NeRF and demonstrates that coarse-to-fine registration is also suitable for NeRF. It also implements a global BA using a neural rendering grid. This method relies on a rough initial camera position generated by methods such as colmap and optimizes the network model and camera position by iterating over the network. In order to solve the problem of position encoding, which causes the gradients between different frequencies to affect each other, the authors proposed an operation similar to a dynamic low-frequency filter that makes the network prioritize the low-frequency part and then gradually learn the information of the high-frequency part. On the other hand, iMAP and NICE-SLAM run the two threads of tracking and mapping at the same time to realize joint optimization of the camera position and the implicit map network structure. The specific methods, pros, and cons are analyzed in Section 3.

In order to solve the problem that NeRF methods are time-consuming and are unable to realize real-time mapping, researchers mainly start from rendering acceleration and training acceleration to improve the real-time performance of their algorithms. The PlenOctrees [27] acceleration strategy decodes all the points in space as well as all the viewpoints using the network and stores the results. It is not necessary to use the network to decode online, but it is used to search directly in the table. However, since the network is a 5D input, it is difficult to exhaust it. Therefore, the authors modified the network and decoupled the observation directions from the network inputs. The network only inputs 3D coordinates and outputs volume density and spherical harmonic coefficients. The color is calculated by the angle of view and the spherical harmonic coefficient. In this way, the input degrees of freedom of the network are reduced from five dimensions to three dimensions. Further, only dense sampling is performed in the three dimensions of x, y, and z, and the results are stored in the octree to improve the rendering speed. Similar to PlenOctrees, SNeRG is an accelerated rendering method. SNeRG [28] divides the colors into inherent colors and specular colors, and inherent colors are independent of the observation viewpoints. The 3D coordinate positions are used as the network inputs. The volume density and intrinsic and specular color feature vectors are decoded. The specular color feature vectors are then decoded by a small network that combines the viewing angle to produce the specular color. The two colors are summed to obtain the final color. The backbone networks of both SNeRG and PlenOctrees are decoupled from the viewing angle—with PlenOctrees recovering the viewing angle color through a spherical harmonic function and SNeRG decoding the specular color by adding a small MLP network—which is superimposed on the intrinsic color to obtain the viewing angle color.

DVGO [29] proposed hybrid explicit and implicit scene representation based on voxel grids. A post-activation method is utilized in order to obtain the volume density of the 3D positions during rendering. That is, the volume density is stored within the voxel grid vertices, and trilinear interpolation is utilized to obtain the volume density at the desired location, followed by softplus activation and inputting of the alpha formula to calculate the opacity. This post-activation strategy uses trilinear interpolation to solve the problem of the voxel grid having low resolution and being unable to express continuous scenes. At the same time, the post-activation strategy enables the voxel grid to store sharp linear planes, which solves the problem of the voxel grid being unable to express high-frequency details. The voxels of DVGO also store the feature vectors of colors, and the RGB values are decoded by the MLP network. This network architecture uses MLP to decode only colors, which makes it lightweight and able to achieve substantial increases in training speed. Plenoxels [30], as a follow-up work of PlenOctrees, uses voxel networks to replace the MLP network for decoding feature vectors. One-dimensional voxel density and spherical

harmonic coefficients are stored within the voxel grid. The volume density and spherical harmonic coefficients of sampling points on any ray in space are obtained by trilinear interpolation. Since Plenoxels removes the MLP network and becomes a fully explicit expression, its training speed is substantially increased.

Overall, the current running rates of NeRF modeling methods still cannot meet real-time demands, and it is difficult to achieve real-time physical simulation in actual environments. NeRF is able to perform 3D dense modeling in an environment and has the ability to render new perspectives and predict unknown regions, which greatly improves the generation and processing of 3D models under preferred observation inputs. However, how to integrate with systems such as mobile robots to achieve real-time positioning, incremental environment reconstruction, dynamic update, and effective management of the 3D environmental space in 3D mission scenarios is still an issue that existing NeRF frameworks need to explore.

3. NeRF-Based SLAM State-of-the-Art

In the previous section, we analyzed the real-time requirements of NeRF. Considering this, combining the NeRF modeling paradigm with the SLAM paradigm is regarded as a crucial development direction in the field of 3D modeling and navigation of mobile robots. This combination aims to build a SLAM system that can implicitly model the environment and learn online. Currently, a large amount of research work has been carried out in this area in the academic community, and a variety of NeRF and SLAM fusion systems and methods have been proposed from different aspects and greatly improve the efficiency and accuracy of environmental 3D modeling.

3.1. NeRF-Based SLAM in Mapping

3.1.1. 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 [31] 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 [31], 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 [32] 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 [33], 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 [34], 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 [35] 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. [12] 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. [36] 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. [37] 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. [38] 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. [39] 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. [40] 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. [41] 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. [42] 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. [43] 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 [44] 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 [45] 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 [46] 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 [11]	2022	✓				✓	✓
Vox-Fusion [35]	2022	✓			✓		✓
NICER-SLAM [34]	2023		✓		✓		✓
Co-SLAM [12]	2023	✓			✓		✓
LONER [38]	2023			✓		✓	✓
Shine-mapping [39]	2023			✓	✓		✓
NF-Atlas [41]	2023			✓	✓		✓
LODE [42]	2023			✓	✓		✓
NeRF-LOAM [45]	2023			✓	✓		✓
LocNDF [11]	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.

3.1.2. 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.

3.2. 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.

3.2.1. 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.

3.2.2. The Method of VO

Orbeez-SLAM [47] 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 [48]

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 [49] 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 [50] 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.

3.3. Loss Function

Through the analyses in Section 3.1.1, we can see that researchers can achieve more accurate mapping by applying different loss functions. This section mainly introduces three widely used loss functions.

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.

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

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

$$\mathcal{L}_p = \frac{1}{M}\sum_{i=1}^{W} \sum_{(u,v)\in s_i} |I_i[u,v] - \hat{I}_i[u,v]|, \tag{2}$$

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.

$$\mathcal{L}_d = \frac{1}{|R_d|} \sum_{r \in R_d} (\hat{d}_r - D[u, v])^2, \tag{3}$$

where D_i is the predicted depth value, $\hat{D}_i$ is the true depth value, and u, v is the corresponding pixel on the image plane.

4. Dataset and System Evaluation

4.1. Datasets

While some of the NeRF-based SLAM methods, especially those capable of working in dynamic and challenging environments, have been tested on robots under realistic conditions, much research work has used publicly available datasets to demonstrate their applicability. The Replica dataset [51] by Straub et al. contains 18 different scenes, including five office scenes and three apartment scenes, and each replica scene contains dense geometry, high-resolution HDR textures, reflectors, and semantic class and instance annotations. It provides a high-fidelity 3D model of realistic indoor scenes, which is more conducive to evaluating the reconstruction performance of a method. The TUM RGB-D dataset [52] proposed by the Technical University of Munich consists of 39 sequences recorded by Microsoft Kinect sensors in different scenarios, contains ground-truth trajectories, and includes Testing and Debugging, HandheldSLAM, Robot SLAM, Structure vs. Texture, Dynamic Objects, 3D Object Reconstruction, Validation Files, Calibration Files, and several task-specific datasets, each of which contains multiple data and can be used for performance testing of a variety of tasks. The ScanNet [53] dataset features richly annotated RGB-D scans of real-world environments and contains more than 1000 captured RGB-D sequences, ground-truth poses, and challenging short and long trajectories estimated from SLAM systems.

Cube-Diorama [54] is a synthetic dataset generated using Blender and which provides ground-truth poses, depths, and images. Using this dataset, we are able to conduct accurate ablation experiments to evaluate the benefits of our proposed method. Maicity [55] is a smaller synthetic dataset with ground-truth grids and high-fidelity LiDAR measurements, which makes it suitable for evaluation testing. KITTI-360 [56] is larger in size and has noisy measurements, which makes it suitable for both local and global evaluation tests. In addition, KITTI-360 also provides semantic cues and filters out dynamic objects to evaluate semantic reconstruction.

4.2. System Evaluation

4.2.1. Scene Reconstruction Evaluation

Two point clouds P and Q are sampled from the real mesh and the reconstructed mesh, respectively, where $|P| = |Q| = n$, and the following three quantitative metrics are used as the evaluation metrics for network reconstruction accuracy [12].

Accuracy (cm): Accuracy is the average distance between the sampling points of the reconstructed mesh and the nearest true point.

$$\sum_{p \in P} (min(min_{q \in Q}) \parallel p - q \parallel) / \mid Q \mid . \tag{4}$$

Completion (cm): Completion is the average distance between the ground-truth mesh sampling point and the nearest reconstruction point.

$$\sum_{q \in Q} (min_{p \in P} \parallel p - q \parallel) / |Q|. \tag{5}$$

Completion Ratio (<5cm%): Completion ratio is the percentage of points in the reconstruction mesh for which the completion degree is less than 5 cm.

$$\sum_{q \in Q} (min_{p \in P} \parallel p - q \parallel < 0.05) / |Q|. \tag{6}$$

4.2.2. TUM Evaluation

RPE (relative pose error) is used to calculate the differences between pose variations within the same two timestamps and to calculate the pose variation between the real pose and the estimated pose for every interval after the timestamps are in alignment. The relative pose error is obtained by subtracting the variation. This standard is suitable for estimating the drift of the system. The RPE of frame i is defined as follows [52]:

$$E_i := (Q_i^{-1} Q_{i+\triangle})^{-1} (P_i^{-1} P_{i+\triangle}). \tag{7}$$

Knowing the total number of n and the interval $\triangle$, $m = n - \triangle$, the RPE can be obtained; then, the root mean square error (RMSE) is used to calculate the error, and a total value is obtained:

$$RMSE(E_{1:n,\triangle}) := (\frac{1}{m} \sum_{i=1}^{m} \parallel trans(E_i) \parallel^2)^{\frac{1}{2}}, \tag{8}$$

where $trans(E_i)$ represents the translational component of the relative pose error E_i. In addition, the rotation error can be evaluated in the same way. In order to ensure that the average of all the time intervals $\triangle$ is meaningful when evaluating the SLAM system, Equation (5) is changed to:

$$RMSE(E_{1:n}) := \frac{1}{n} \sum_{\triangle=1}^{1} RMSE(E_{1:n,\triangle}). \tag{9}$$

ATE (absolute trajectory error) is used to calculate the difference between the estimated pose and the real ground pose; it can be used to visualize the algorithm's accuracy and the global consistency of the trajectory. Since the estimated pose and the ground-truth are usually not in the same coordinate system, they need to be aligned first: for a binocular camera and an RGB-D camera with uniform scales, an estimated-to-true pose transformation matrix $S \in SE(3)$ needs to be computed by the least-squares algorithm; for a monocular camera with uncertain scales, it is necessary to calculate the similarity transformation matrix $S \in Sim(3)$ from the estimated pose to the real pose. The ATE of frame i is defined as follows [52]:

$$F_i := Q_i^{-1} S P_i^{-1}. \tag{10}$$

Knowing the total number of n and the interval $\triangle$, $m = n - \triangle$, the ATE can be obtained; then, the root mean square error (RMSE) is used to calculate the error, and a total value is obtained:

$$RMSE(F_{1:n,\triangle}) := (\frac{1}{m} \sum_{i=1}^{m} \parallel trans(F_i) \parallel^2)^{\frac{1}{2}}, \tag{11}$$

where $trans(F_i)$ represents the translation component of the relative pose error F_i.

5. Current Issues and Future Work

5.1. Dynamic Environment Expression

Traditional SLAM and NeRF methods mainly focus on static scenes, while real-life scenes usually involve object motion or topological changes, such as moving vehicles and pedestrians. In a real environment, the traditional SLAM method needs to implement the tracking of dynamic objects to ensure that the system can better react to real-time changes in the environment and achieve effective tracking of moving objects and timely updates to the environment map. This not only helps to improve the robustness of the SLAM method and the navigation performance in a dynamic environment, but it also provides wider

application potential in the unmanned field. Therefore, it is inevitable to realize dynamic modeling using NeRF-based SLAM methods. With the emergence of networks such as TiNeuVox [57], k-planes [58], and HexPlane [59], NeRF realizes dynamic scene modeling through mixed scene expression. NICE-SLAM takes into account the difficulty of processing dynamic environments for the next joint optimization of the camera pose and network parameters. By deleting pixels with large depths or dark colors during the mapping process, NICE-SLAM effectively ignores dynamic objects and improves system robustness. How to combine tiny transformable grids with increased time dimensions, motion coding, and NeRF inverse rendering and realize real-time dynamic scene expression while performing joint optimization is a major direction of future NeRF-based SLAM system research.

5.2. SDF and Volume Density Considerations

From the analysis in Section 3, it is not difficult to see that in order to make the NeRF-based SLAM method achieve higher accuracy of camera pose estimation, the map representation needs to be made more explicit. Numerous researchers have replaced the volume density proposed in NeRF with SDF values. Here, we first analyze the advantages of SDF compared to volume density.

- SDF values and volume densities decoded by MLP networks cannot be mutated without losing texture features. However, since zero iso-surface decision boundaries are abrupt, the use of SDF values can enable the extraction of surfaces containing texture features.
- A loss function based on SDF values constrains the on-surface points and off-surface points, so SDF values can provide more geometric information.
- The loss function of an SDF sets the SDF value away from the surface of the object as a cut-off value. Combined with the voxel skipping strategy, unnecessary calculations can be avoided, while the volume density is affected by floating objects in the air, which requires more sampling points and more calculations. Combined with the voxel skipping strategy, unnecessary calculations can be avoided. And because the volume density is affected by floating objects in the air, the calculation cost is large.
- An SDF value can visually describe the distance between a sampling point and an object's surface, which is conducive to the realization of AR and VR.

Although SDF outperforms volume density for reconstruction, replacing the volume density with SDF sacrifices the hole filling performance brought by the rendering function. How to find the trade-off between the SDF value and the volume density is a problem that needs to be solved.

5.3. Loop Detection and Positioning Performance Improvements

In the SLAM method, the acquisition of sensor data as well as camera pose estimation over long periods of time accumulates drift error; the loop detection module corrects the drift error to the global map by detecting whether the camera has passed through a position that has previously been passed through, and the drift error of the global map is corrected by the camera pose error when it reaches the same position twice. ORB-SLAM2 uses a bag-of-words model, which is specifically used to retrieve and match the ORB features of the current frame that are recognized as a loop. After identifying the loop, correction of the pose of the keyframe is also required. Under existing implicit environment representation frameworks, adjusting the global information of the environment makes the original implicit model parameters no longer applicable. However, there are already methods using implicit feature interpolation to modify local map parameters without modifying global map grid parameters and then merging them into the global map to restore the complete global map. It is not difficult to see that this process is not as good as traditional SLAM loop detection, which still has development prospects. How to make the position accuracy and loop detection performance of NeRF-based SLAM methods exceed or equal that of traditional visual odometry, or how to make NeRF play a unique role in VO, still deserves discussion.

6. Conclusions

With the development of NeRF and other neural-rendering-based environment reconstruction methods, adding NeRF models into SLAM has enabled SLAM systems to make great progress in continuous mapping: improving the robustness and generalization of systems and overcoming more of the weaknesses of the original SLAM models. NeRF-based SLAM has become a new development direction in the field of SLAM. This paper analyzes 30 papers in the field of NeRF-based SLAM; studies the evolution, advantages, and disadvantages of SLAM methods and NeRF methods; analyzes the improvements to NeRF-based SLAM systems as a result of having two parallel running threads; and finds some highly mature frameworks. Most follow-up work will be based on what has already been built. Finally, the problems existing in NeRF-based SLAM systems at the present stage are pointed out, and future development is prospected. Through this review, we hope to introduce more related practitioners to this field, to stimulate future research on NeRF-based SLAM, and to promote SLAM system development in the future.

Author Contributions: Conceptualization, K.Y., Y.C., Z.C. and J.W.; Conceptualization, K.Y.; Formal analysis, Y.C. and Z.C.; Funding acquisition, J.W.; Investigation, K.Y., and J.W.; Methodology, K.Y., and J.W.; Project administration, Z.C. and J.W.; Writing—original draft, K.Y. and Y.C.; Writing—review & editing, Z.C. and J.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Natural Science Foundation of China: 62103393.

Data Availability Statement: No applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Khairuddin, A.R.; Talib, M.S.; Haron, H. Review on simultaneous localization and mapping (SLAM). In Proceedings of the 2015 IEEE International Conference on Control System, Computing and Engineering (ICCSCE), Penang, Malaysia, 27–29 November 2015; pp. 85–90.
2. Vallivaara, I.; Haverinen, J.; Kemppainen, A.; Röning, J. Magnetic field-based SLAM method for solving the localization problem in mobile robot floor-cleaning task. In Proceedings of the 2011 15th International Conference on Advanced Robotics (ICAR), Tallinn, Estonia, 20–23 June 2011; pp. 198–203.
3. Yang, T.; Li, P.; Zhang, H.; Li, J.; Li, Z. Monocular vision SLAM-based UAV autonomous landing in emergencies and unknown environments. *Electronics* **2018**, *7*, 73. [CrossRef]
4. Liu, Z.; Chen, H.; Di, H.; Tao, Y.; Gong, J.; Xiong, G.; Qi, J. Real-time 6d lidar slam in large scale natural terrains for ugv. In Proceedings of the 2018 IEEE Intelligent Vehicles Symposium (IV), Changshu, China, 26–30 June 2018; pp. 662–667.
5. Yeh, Y.J.; Lin, H.Y. 3D reconstruction and visual SLAM of indoor scenes for augmented reality application. In Proceedings of the 2018 IEEE 14th International Conference on Control and Automation (ICCA), Anchorage, AL, USA, 12–15 June 2018; pp. 94–99.
6. Strasdat, H.; Montiel, J.M.; Davison, A.J. Visual SLAM: Why filter? *Image Vis. Comput.* **2012**, *30*, 65–77. [CrossRef]
7. Taheri, H.; Xia, Z.C. SLAM; definition and evolution. *Eng. Appl. Artif. Intell.* **2021**, *97*, 104032. [CrossRef]
8. Macario Barros, A.; Michel, M.; Moline, Y.; Corre, G.; Carrel, F. A comprehensive survey of visual slam algorithms. *Robotics* **2022**, *11*, 24. [CrossRef]
9. Mildenhall, B.; Srinivasan, P.P.; Tancik, M.; Barron, J.T.; Ramamoorthi, R.; Ng, R. Nerf: Representing scenes as neural radiance fields for view synthesis. *Commun. ACM* **2021**, *65*, 99–106. [CrossRef]
10. Kazerouni, I.A.; Fitzgerald, L.; Dooly, G.; Toal, D. A survey of state-of-the-art on visual SLAM. *Expert Syst. Appl.* **2022**, *205*, 117734. [CrossRef]
11. 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.
12. 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.
13. Sandström, E.; Li, Y.; Van Gool, L.; Oswald, M.R. Point-slam: Dense neural point cloud-based slam. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Vancouver, BC, Canada, 18–22 June 2023; pp. 18433–18444.
14. Sunderhauf, N.; Lange, S.; Protzel, P. Using the unscented kalman filter in mono-SLAM with inverse depth parametrization for autonomous airship control. In Proceedings of the 2007 IEEE International Workshop on Safety, Security and Rescue Robotics, Rome, Italy, 27–29 September 2007; pp. 1–6.

15. Klein, G.; Murray, D. Parallel tracking and mapping for small AR workspaces. In Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality, Nara, Japan, 13–16 November 2007; pp. 225–234.
16. Mendes, E.; Koch, P.; Lacroix, S. ICP-based pose-graph SLAM. In Proceedings of the 2016 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR), Lausanne, Switzerland, 23–27 October 2016; pp. 195–200.
17. Endres, F.; Hess, J.; Sturm, J.; Cremers, D.; Burgard, W. 3-D mapping with an RGB-D camera. *IEEE Trans. Robot.* **2013**, *30*, 177–187. [CrossRef]
18. Forster, C.; Pizzoli, M.; Scaramuzza, D. SVO: Fast semi-direct monocular visual odometry. In Proceedings of the 2014 IEEE international conference on robotics and automation (ICRA), Hong Kong, China, 31 May–7 June 2014; pp. 15–22.
19. Engel, J.; Schöps, T.; Cremers, D. LSD-SLAM: Large-scale direct monocular SLAM. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; Springer: Berlin/Heidelberg, Germany, 2014; pp. 834–849.
20. Tateno, K.; Tombari, F.; Laina, I.; Navab, N. Cnn-slam: Real-time dense monocular slam with learned depth prediction. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 6243–6252.
21. Mur-Artal, R.; Montiel, J.M.M.; Tardos, J.D. ORB-SLAM: A versatile and accurate monocular SLAM system. *IEEE Trans. Robot.* **2015**, *31*, 1147–1163. [CrossRef]
22. Mur-Artal, R.; Tardós, J.D. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Trans. Robot.* **2017**, *33*, 1255–1262. [CrossRef]
23. Campos, C.; Elvira, R.; Rodríguez, J.J.G.; Montiel, J.M.; Tardós, J.D. Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam. *IEEE Trans. Robot.* **2021**, *37*, 1874–1890. [CrossRef]
24. A detailed map of Higgs boson interactions by the ATLAS experiment ten years after the discovery. *Nature* **2022**, *607*, 52–59. [CrossRef]
25. Yen-Chen, L.; Florence, P.; Barron, J.T.; Rodriguez, A.; Isola, P.; Lin, T.Y. inerf: Inverting neural radiance fields for pose estimation. In Proceedings of the 2021 IEEE RSJ International Conference on Intelligent Robots and Systems (IROS), Prague, Czech Republic, 27 September–1 October 2021; pp. 1323–1330.
26. Lin, C.H.; Ma, W.C.; Torralba, A.; Lucey, S. Barf: Bundle-adjusting neural radiance fields. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 5741–5751.
27. Yu, A.; Li, R.; Tancik, M.; Li, H.; Ng, R.; Kanazawa, A. Plenoctrees for real-time rendering of neural radiance fields. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 5752–5761.
28. Hedman, P.; Srinivasan, P.P.; Mildenhall, B.; Barron, J.T.; Debevec, P. Baking neural radiance fields for real-time view synthesis. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, BC, Canada, 11–17 October 2021; pp. 5855–5864.
29. Sun, C.; Sun, M.; Chen, H.T. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 5459–5469.
30. Fridovich-Keil, S.; Yu, A.; Tancik, M.; Chen, Q.; Recht, B.; Kanazawa, A. Plenoxels: Radiance fields without neural networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 5501–5510.
31. 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.
32. 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.
33. 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.
34. 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.
35. 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.
36. Yan, D.; Lyu, X.; Shi, J.; Lin, Y. Efficient Implicit Neural Reconstruction Using LiDAR. *arXiv* **2023**, arXiv:2302.14363.
37. 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. [CrossRef]
38. 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. [CrossRef]
39. 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.

40. Liu, J.; Chen, H. Towards Real-time Scalable Dense Mapping using Robot-centric Implicit Representation. *arXiv* **2023**, arXiv:2306.10472.
41. 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.
42. 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.
43. 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. [CrossRef]
44. 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.
45. 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.
46. 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.
47. 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.
48. 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.
49. Moad, G.; Rizzardo, E.; Thang, S.H. Living radical polymerization by the RAFT process. *Aust. J. Chem.* **2005**, *58*, 379–410. [CrossRef]
50. 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.
51. Straub, J.; Whelan, T.; Ma, L.; Chen, Y.; Wijmans, E.; Green, S.; Engel, J.J.; Mur-Artal, R.; Ren, C.; Verma, S. The Replica dataset: A digital replica of indoor spaces. *arXiv* **2019**, arXiv:1906.05797.
52. Sturm, J.; Engelhard, N.; Endres, F.; Burgard, W.; Cremers, D. A benchmark for the evaluation of RGB-D SLAM systems. In Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, Vilamoura, Portugal, 7–12 October 2012; pp. 573–580.
53. Dai, A.; Chang, A.X.; Savva, M.; Halber, M.; Funkhouser, T.; Nießner, M. Scannet: Richly-annotated 3D reconstructions of indoor scenes. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 5828–5839.
54. Abou-Chakra, J.; Dayoub, F.; Sünderhauf, N. Implicit object mapping with noisy data. *arXiv* **2022**, arXiv:2204.10516.
55. Vizzo, I.; Chen, X.; Chebrolu, N.; Behley, J.; Stachniss, C. Poisson surface reconstruction for LiDAR odometry and mapping. In Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 30 May–5 June 2021; pp. 5624–5630.
56. Liao, Y.; Xie, J.; Geiger, A. KITTI-360: A novel dataset and benchmarks for urban scene understanding in 2D and 3D. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *45*, 3292–3310. [CrossRef]
57. Fang, J.; Yi, T.; Wang, X.; Xie, L.; Zhang, X.; Liu, W.; Nießner, M.; Tian, Q. Fast dynamic radiance fields with time-aware neural voxels. In Proceedings of the SIGGRAPH Asia 2022, Daegu, Republic of Korea, 6–9 December 2022; pp. 1–9.
58. Fridovich-Keil, S.; Meanti, G.; Warburg, F.R.; Recht, B.; Kanazawa, A. K-planes: Explicit radiance fields in space, time, and appearance. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancuver, BC, Canada, 18–22 June 2023; pp. 12479–12488.
59. Cao, A.; Johnson, J. Hexplane: A fast representation for dynamic scenes. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancuver, BC, Canada, 18–22 June 2023; pp. 130–141.