LRSLAM: Low-rank Representation of Signed Distance Fields in Dense Visual SLAM System

Hongbeen Park[1], Minjeong Park[2], Giljoo Nam[3], and Jinkyu Kim[1]

[1] Dept of Computer Science and Engineering, Korea University, Seoul, Korea
[2] Dept of Electrical and Electronic Engineering, Yonsei University, Seoul, Korea
[3] Meta Reality Labs, Pittsburgh, PA 15222, USA

Abstract. Simultaneous Localization and Mapping (SLAM) has been crucial across various domains, including autonomous driving, mobile robotics, and mixed reality. Dense visual SLAM, leveraging RGB-D camera systems, offers advantages but faces challenges in achieving real-time performance, robustness, and scalability for large-scale scenes. Recent approaches utilizing neural implicit scene representations show promise but suffer from high computational costs and memory requirements. ESLAM introduced a plane-based tensor decomposition but still struggled with memory growth. Addressing these challenges, we propose a more efficient visual SLAM model, called LRSLAM, utilizing low-rank tensor decomposition methods. Our approach, leveraging the Six-axis and CP decompositions, achieves better convergence rates, memory efficiency, and reconstruction/localization quality than existing state-of-the-art approaches. Evaluation across diverse indoor RGB-D datasets demonstrates LRSLAM's superior performance in terms of parameter efficiency, processing time, and accuracy, retaining reconstruction and localization quality. Our code will be publicly available upon publication.

Keywords: Dense Visual SLAM · Low Rank Representation · Six-axis Decomposition

1 Introduction

Simultaneous Localization and Mapping (SLAM) has been an essential technology in various domains, such as autonomous driving [1, 5], indoor/outdoor mobile robotics [8, 18, 21], and mixed reality [6, 13, 22]. Recently, dense visual SLAM approaches based on an RGB-D camera system with additional depth information have been actively explored due to the advantages of simple sensor configuration. Despite promising results, their high computational costs make it challenging to achieve (i) real-time performance, (ii) robustness, and (iii) scalability to deal with large-scale scenes [14, 25, 27, 32]. These are crucial factors in rendering a SLAM system truly effective for real-world applications.

* Corresponding author: J. Kim (jinkyukim@korea.ac.kr)

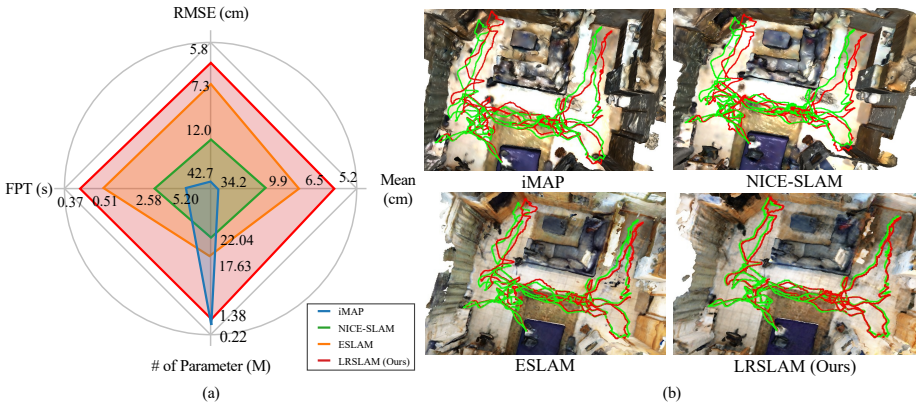

Fig. 1: (a) **Comparison with SOTA Approaches.** Our model is clearly positioned as an efficient yet effective dense visual SLAM method. Our proposed LRSLAM requires fewer parameters, faster frame processing time (FPT), and better reconstruction accuracy (regarding ATE mean and RMSE). This is further validated by our (b) **Qualitative Comparison of Scene Reconstruction and Localization** between our proposed LRSLAM and the state-of-the-art approaches, including iMAP [25], NICE-SLAM [32] and ESLAM [14]. Ours shows comparable or better reconstruction and localization accuracy with highly compact representations.

Learning underlying representations of scene geometry and appearance is pivotal in building such a visual SLAM system. With remarkable success with Neural Radiance Fields (NeRF) techniques [20], recent work [25, 32] suggests that neural implicit scene representation can be utilized to learn geometry and appearance representations, optimizing a 3D map and camera poses for a visual SLAM system. Yet, their cubic memory growth rate necessitates employing voxel grids with reduced resolutions, sacrificing intricate geometric details. More recently, ESLAM [14] leverages plane-based tensor decomposition to achieve efficient and accurate localization and reconstruction. Despite its promising outcomes, it still has a quadratic memory growth rate, which is still challenging for a real-time visual SLAM system.

Following this stream of visual SLAM models, we propose a more efficient model with a linear memory growth rate, thereby improving both the efficiency and accuracy of SLAM tasks, i.e., localization and reconstruction. To this end, we focus on compactly factorizing the 3D geometry and appearance of a scene into parameterized low-rank components (more compact than ESLAM's plane-based representation), enabling a compact yet expressive scene representation. Specifically, we propose a new tensor decomposition method, called Six-axis decomposition, which factorizes the three planes in the tri-plane representation into six axis-aligned feature tensors, thus holding an efficient memory complexity of $O(n)$. Also, we propose a hybrid scene representation using both the conventional CP decomposition [2] and our new Six-axis decomposition to further improve the overall performance of SLAM.

In summary, we propose an efficient visual SLAM method, called LRSLAM, which leverages a combination of low-rank tensor decomposition methods (i.e., our proposed Six-axis decomposition and CP decomposition) to provide a better convergence rate, memory efficiency, and reconstruction/localization quality. We observe that a hybrid use of these two tensor decomposition methods provides a notable benefit in the following two aspects: (1) CP decomposition allows compact and fast encoding of the geometry features (once converged, it also helps to learn the appearance features) and (2) Six-axis (SA) decomposition allows to learn detailed appearance features with more expressive yet efficient decomposition, which is essential for the tracking task. We conduct thorough evaluations across diverse indoor RGB-D datasets, including ScanNet [7], TUM RGB-D [24], and Replica [23]. In our experiments, our model uses remarkably fewer parameters (87.3%–90.1% fewer than ESLAM [14]) and shows faster processing time (4.3%–73.2% than ESLAM [14]), retaining reconstruction and localization accuracy.

We summarize our contributions as follows:

- We propose a novel tensor decomposition method called Six-axis decomposition. This method compactly factorizes the tri-planes into six axis-aligned feature tensors with a linear memory growth rate.
- We leverage the novel Six-axis decomposition together with the traditional CP decomposition to achieve compact yet effective RGB-D SLAM performance.
- Our extensive experiments with three public datasets (i.e., ScanNet, TUM RGB-D, and Replica) validate the effectiveness of our proposed approach, which significantly reduces the demand for parameters while showing a faster convergence rate with matched or outperforming reconstruction and localization performance.

2 Related Work

Visual SLAM. Visual SLAM techniques can mainly be categorized into three types depending on the data source: (i) Visual-only SLAM [6,9,26,29], which utilizes a mono (or multiple) camera system and thus needs to accurately estimate depth from cameras only, which is still technically challenging. (ii) Visual-inertial SLAM [10, 11], which relies on additional inertial measurement units (IMUs) to improve the overall accuracy. However, their system is prone to noise in calibration and inertial measurements. (iii) RGB-D SLAM [14, 25, 31, 32] utilizes additional depth information and thus provides reliable and improved performance. Recent approaches have significantly improved with RGB-D SLAM, reporting better accuracy and robustness, albeit with drawbacks such as increased memory and power requirements. In this paper, we follow the stream of RGB-D SLAM and aim to improve its accuracy, satisfying low memory and power requirements.

Neural Scene Representation for Visual SLAM. Neural Radiance Fields (NeRF, [20]) have had a significant impact on various applications such as large-

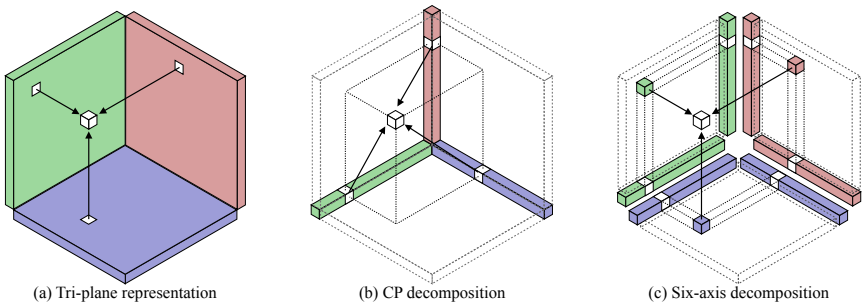

Fig. 2: (a) **Tri-plane representation** factorizes a 4D tensor into three feature planes (Eq. 1). (b) **CP decomposition** factorizes a 4D tensor into a sum of outer products of three axis-aligned low-rank feature tensors (Eq. 2). (c) Our proposed **Six-axis decomposition** factorizes a 4D tensor into a set of six axis-aligned low-rank feature tensors (Eq. 3).

scale 3D reconstructions by utilizing neural implicit representations. Recent work suggests that such neural implicit representations can be applied to dense visual SLAM systems, showing promising localization performance by accurately representing the scene. A landmark work is iMAP [25], which utilizes an implicit neural scene representation in a real-time SLAM system using an RGB-D camera, proving its capability to optimize a 3D map and camera poses. However, its performance is often limited due to the model's capacity for encoding a wide scene. To solve this, NICE-SLAM [32] extends iMAP by representing the scene with voxel grid features and converting them into occupancies using pre-trained MLPs. However, their model's cubic memory growth rate leads to the use of low-resolution voxel grids and the loss of fine geometric details. Recently, ESLAM [14] has been introduced for efficient yet accurate localization and reconstruction by leveraging plane-based tensor decomposition. Despite their promising results, their model still exhibits quadratic memory growth, leading to constraints and sub-optimal performance for real-time visual SLAM systems. Thus, in this work, we propose a more efficient model with a linear memory growth rate, further enhancing the efficiency and accuracy of localization and reconstruction.

Concurrent Work using 3D Gaussian Splatting. 3D Gaussian Splatting (3DGS) [16] is gaining increasing attention as a new scene representation. 3DGS is well-known for its faithful 3D reconstruction quality and the efficient rendering paradigm. As concurrent work of ours, several SLAM systems are proposed to utilize 3DGS as a scene representation [12,15,17,19,28,30]. However, as discussed in [17,28], the large memory consumption is also an issue in 3DGS-based SLAM systems, because they often fail to manage the number of Gaussians in unseen areas [27].

3 Efficient Scene Representations

The 3D geometry of a scene can be depicted using a signed distance field (SDF), denoted as $f_{SDF} : \mathbb{R}^3 \rightarrow \mathbb{R}$. This function maps a 3D location $\mathbf{p}$ to a scalar

value s, which is the distance from $\mathbf{p}$ to the nearest surface in the scene. This function can be efficiently implemented using a combination of factorized tensors and a small neural network [3,4]. In formal terms, $f_{SDF} = \texttt{MLP}(f(\mathbf{p}))$, where $f : \mathbb{R}^3 \rightarrow \mathbb{R}^C$ is a function that takes a 3D location $\mathbf{p}$ as input and outputs a length-C feature vector, and $\texttt{MLP} : \mathbb{R}^C \rightarrow \mathbb{R}$ is a multilayer perceptron that decodes this feature vector into a scalar value s. The compactness and accuracy of these representations depend on the specific method employed for the function f. In this section, we discuss two preliminary methods, i.e., tri-plane representation and CP decomposition and then introduce our novel tensor decomposition method, the Six-axis (SA) decomposition.

Tri-plane Representation. The tri-plane representation [3] employs three 2D feature planes, i.e., $f_{xy}, f_{yz}, f_{zx} \in \mathbb{R}^{L \times L \times C}$ each with a spatial resolution of $L \times L$ and C feature channels. To query the feature vector at a 3D location $\mathbf{p}$, we first project $\mathbf{p}$ onto each axis-aligned plane and aggregate the three retrieved features from respective planes. Chan et al. [3] suggested that summation can serve as an efficient feature aggregation method, yielding the tri-plane representation as follows:

$$f_{\text{tri-plane}}(\mathbf{p}) = f_{xy}(\mathbf{p}) + f_{yz}(\mathbf{p}) + f_{zx}(\mathbf{p}). \tag{1}$$

This representation has $\mathrm{O}(n^2)$ space complexity where n is the side length of the scene. See Fig. 2 (a).

CP Decomposition. CP decomposition provides a more compact representation of $f()$ than the tri-plane representation. It factorizes a 4D tensor into a sum of outer products of three axis-aligned rank-one tensors as follows:

$$f_{\text{CP}}(\mathbf{p}) = \{\sum_{i=1}^{k} f_x^{(i)} \otimes f_y^{(i)} \otimes f_z^{(i)}\}(\mathbf{p}) \tag{2}$$

where $f_x^{(i)}, f_y^{(i)}, f_z^{(i)} \in \mathbb{R}^{L \times C}$ are factorized low-rank tensors of three modes for the i-th component. As illustrated in Fig. 2 (b), the CP decomposition has $\mathrm{O}(n)$ space complexity and can provide the most compact representation of a scene. However, as discussed in [4], CP decomposition may cause information loss due to is extreme compactness.

Six-axis Decomposition. Here we present our novel Six-axis (SA) decomposition. The key idea is to factorize the three planes in the tri-plane representation into a set of six axis-aligned low-rank feature tensors. Given three feature planes of rank k, i.e., f_{xy}, f_{yz}, f_{zx}, they are further factorized as the sum of outer products of k rank one tensors: e.g., $f_{xy} = \sum_{i=1}^{k} f_{x_y}^{(i)} \otimes f_{y_x}^{(i)}$ where $f_{x_y}^{(i)}, f_{y_x}^{(i)} \in \mathbb{R}^{L \times C}$ are axis-aligned low-rank tensors. Similarly, other feature planes, f_{yz} and f_{zx}, can be decomposed into the sum of outer products of k axis-aligned low-rank tensors, yielding the Six-axis decomposition as follows:

$$f_{\text{SA}}(\mathbf{p}) = \{\sum_{i=1}^{k} f_{x_y}^{(i)} \otimes f_{y_x}^{(i)}\}(\mathbf{p}) + \{\sum_{i=1}^{k} f_{y_z}^{(i)} \otimes f_{z_y}^{(i)}\}(\mathbf{p}) + \{\sum_{i=1}^{k} f_{z_x}^{(i)} \otimes f_{x_z}^{(i)}\}(\mathbf{p}) \tag{3}$$

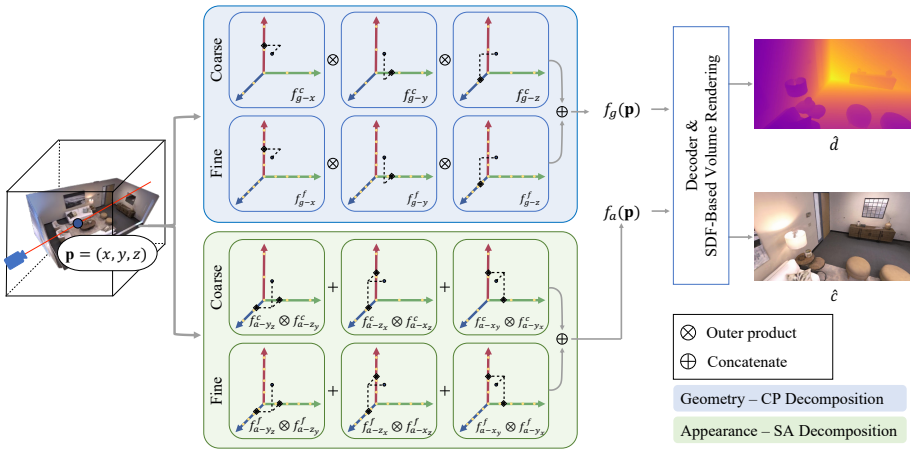

Fig. 3: An overview of our proposed model, called LRSLAM. Our model utilizes a combination of low-rank tensor decomposition methods to provide a better convergence rate, memory efficiency, and reconstruction quality. Specifically, we utilize the CP decomposition to represent the geometry of a scene (see top, f_{g}) and use our Six-axis decomposition for reconstructing color (see bottom, f_{a}).

where $f_{y_z}^{(i)}, f_{z_y}^{(i)}, f_{z_x}^{(i)}, f_{x_z}^{(i)} \in \mathbb{R}^{L \times C}$ are axis-aligned features. As shown in Fig. 2 (c), compared to the tri-plane representation, Six-axis decomposition has advantages in holding the efficient memory complexity of $\mathrm{O}(n)$, retaining the high capability of scene encoding and decoding.

4 Low-rank Representations for RGB-D SLAM

In this section, we propose LRSLAM, a memory-efficient RGB-D SLAM that employs low-rank representations of signed distance fields. We first provide a summary of the baseline approach ESLAM, which is the state-of-the-art RGB-D SLAM method. We then delve into the specifics of employing the novel SA decomposition together with the traditional CP decomposition to achieve optimal RGB-D SLAM performance.

4.1 Baseline Method

We use ESLAM [14] as the baseline approach for the RGB-D SLAM system. Given a sequence of RGB-D frames $\{I_i, D_i\}_{i=1}^{M}$, ESLAM jointly estimates camera poses $\{R_i|t_i\}_{i=1}^{M}$, an SDF $f_{SDF} : \mathbb{R}^3 \to \mathbb{R}$, and its appearance $f_{RGB} : \mathbb{R}^3 \to \mathbb{R}^3$.

Scene Representation. The main technical contribution of ESLAM is the employment of tri-plane representation to efficiently model the SDF of a scene. Specifically, ESLAM models a scene with twelve axis-aligned multi-resolution feature planes, six for geometry and the other six for appearance. Formally, it has

three coarse geometry feature planes $\{F^c_{g\text{-}xy}, F^c_{g\text{-}xz}, F^c_{g\text{-}yz}\}$ and three fine geometry feature planes $\{F^f_{g\text{-}xy}, F^c_{g\text{-}xz}, F^f_{g\text{-}yz}\}$. Appearance feature planes are similarly defined as $\{F^c_{a\text{-}xy}, F^c_{a\text{-}xz}, F^c_{a\text{-}yz}\}$ and $\{F^f_{a\text{-}xy}, F^f_{a\text{-}xz}, F^f_{a\text{-}yz}\}$. To query the feature vector of a 3D location $\mathbf{p}$, they sum the bilinearly interpolated features of each plane and then concatenate the coarse and fine features together. For example, a geometric feature at $\mathbf{p}$ is obtained:

$$\begin{aligned} f^c_g(\mathbf{p}) &= F^c_{g\text{-}xy}(\mathbf{p}) + F^c_{g\text{-}xz}(\mathbf{p}) + F^c_{g\text{-}yz}(\mathbf{p}) \\ f^f_g(\mathbf{p}) &= F^f_{g\text{-}xy}(\mathbf{p}) + F^f_{g\text{-}xz}(\mathbf{p}) + F^f_{g\text{-}yz}(\mathbf{p}) \\ f_g(\mathbf{p}) &= [f^c_g(\mathbf{p}); f^f_g(\mathbf{p})]. \end{aligned} \tag{4}$$

An appearance feature $f^f_a(\mathbf{p})$ can be obtained in the same manner. The features are then decoded into the final SDF value $\phi_g(\mathbf{p})$ and color $\phi_a(\mathbf{p})$:

$$\begin{aligned} \phi_g(\mathbf{p}) &= \texttt{MLP}_g(f_g(\mathbf{p})) \\ \phi_a(\mathbf{p}) &= \texttt{MLP}_a(f_a(\mathbf{p})). \end{aligned} \tag{5}$$

In practice, truncated signed distance field (TSDF) is used because SLAM systems only require geometric information around the surfaces in a scene. The TSDF value $\phi_g(\mathbf{p})$ and its color $\phi_a(\mathbf{p})$ can effectively model a 3D scene.

SDF-based Volume Rendering. Given the current camera pose estimates, random pixels are chosen and the corresponding rays are obtained. For each ray, N points $\{\mathbf{p}\}^N_{i=1}$ are sampled and their corresponding TSDF $\phi_g(\mathbf{p}_i)$ and raw color $\phi_c(\mathbf{p}_i)$ are computed using Eq. 5. Specifically, N points are composed of N_s points from stratified sampling and N_t points from importance sampling. For volume rendering, TSDF values are converted into volume densities via a mapping function $\phi_d(\mathbf{p}_i) = \beta \cdot \text{Sigmoid}(-\beta \cdot \phi_g(\mathbf{p}_i))$, where β is a learnable parameter to control the thickness of surface boundaries. The color $\hat{\mathbf{c}}$ and depth $\hat{d}$ of the ray can be computed as follows:

$$\begin{aligned} \hat{c} = \sum_{i=1}^{N} w_i \phi_a(\mathbf{p}_i), \quad \hat{d} = \sum_{i=1}^{N} w_i z_i \\ w_i = \exp(-\sum_{j=1}^{i-1} \phi_d(\mathbf{p}_j))(1 - \exp(-\phi_d(\mathbf{p}_i))) \end{aligned} \tag{6}$$

where z_i is the depth of a given point $\mathbf{p}_i$. Please refer to [14] for more details about the pixel and point sampling strategies.

Simultaneous Localization and Mapping. Using a set of generated rays, ESLAM optimizes camera poses and scene parameters by minimizing a loss function that consists of several terms: depth loss, color loss, free-space loss, and TSDF loss. The depth loss minimize the difference between rendered depth $\hat{d}$ and sensor-measured depth. The color losses works similarly for the rendered color $\hat{\mathbf{c}}$. The free-space loss prevents empty spaces form having valid SDF values, and the TSDF loss makes the SDF values $\phi_g(\mathbf{p})$ confirm to the sensor-measured depth. We refer to [14] for more implementation details about the loss functions and overall SLAM pipeline.

4.2 LRSLAM

We propose to use low-rank representations of 4D tensors to enhance the efficiency of RGB-D SLAM. Our approach involves the use of CP decomposition to depict the geometric aspect of a scene via a truncated signed distance field (TSDF). At the same time, we employ our SA decomposition to represent the scene's appearance. In the following, we provide a detailed explanation of our proposal and discuss the reasoning behind our choice of this hybrid representation. All the experiments shown in this paper base on the ESLAM [14] framework, but with a different scene representation.

CP Decomposition for Geometry. We use a set of low-rank feature tensors to represent the geometry of a scene with the CP decomposition. Formally, the geometry feature vector $f_{\mathrm{g}}(\mathbf{p}) \in \mathbb{R}^C$ at a point $\mathbf{p}$, is computed as the sum of outer products of three axis-aligned vectors as follows:

$$\begin{aligned} f_g^c(\mathbf{p}) &= \{\sum_{i=1}^{k_g} f_{g\text{-}x}^{c(i)} \otimes f_{g\text{-}y}^{c(i)} \otimes f_{g\text{-}z}^{c(i)}\}(\mathbf{p}) \\ f_g^f(\mathbf{p}) &= \{\sum_{i=1}^{k_g} f_{g\text{-}x}^{f(i)} \otimes f_{g\text{-}y}^{f(i)} \otimes f_{g\text{-}z}^{f(i)}\}(\mathbf{p}) \\ f_g(\mathbf{p}) &= [f_g^c(\mathbf{p}); f_g^f(\mathbf{p})], \end{aligned} \tag{7}$$

where $f_{g\text{-}x}^{c(i)}$ is the i-th coarse-level rank-one tensor for geometry. Other features are similarly defined. The final TSDF value is decoded via a small MLP:

$$\phi_g(\mathbf{p}) = \mathtt{MLP}_g(f_g(\mathbf{p})). \tag{8}$$

SA Decomposition for Appearance. For appearance, we use our SA decomposition. The appearance feature vector $f_{\mathrm{a}}(\mathbf{p}) \in \mathbb{R}^C$ at a point $\mathbf{p}$ is computed as the sum of outer products of six axis-aligned vectors:

$$f_a^c(\mathbf{p}) = \{\sum_{i=1}^{k_a} f_{a\text{-}x_y}^{c(i)} \otimes f_{a\text{-}y_x}^{c(i)}\}(\mathbf{p}) + \{\sum_{i=1}^{k_a} f_{a\text{-}y_z}^{c(i)} \otimes f_{a\text{-}z_y}^{c(i)}\}(\mathbf{p}) + \{\sum_{i=1}^{k_a} f_{a\text{-}z_x}^{c(i)} \otimes f_{a\text{-}x_z}^{c(i)}\}(\mathbf{p}) \tag{9}$$

$$f_a^f(\mathbf{p}) = \{\sum_{i=1}^{k_a} f_{a\text{-}x_y}^{f(i)} \otimes f_{a\text{-}y_x}^{f(i)}\}(\mathbf{p}) + \{\sum_{i=1}^{k_a} f_{a\text{-}y_z}^{f(i)} \otimes f_{a\text{-}z_y}^{f(i)}\}(\mathbf{p}) + \{\sum_{i=1}^{k_a} f_{a\text{-}z_x}^{f(i)} \otimes f_{a\text{-}x_z}^{f(i)}\}(\mathbf{p}) \tag{10}$$

$$f_a(\mathbf{p}) = [f_a^c(\mathbf{p}); f_a^f(\mathbf{p})], \tag{11}$$

where $f_{a\text{-}x_y}^{c(i)}$ is the i-th coarse-level rank-one tensor for appearance. Remaining features can be similarly defined. The final color is decoded via a small MLP:

$$\phi_a(\mathbf{p}) = \mathtt{MLP}_a(f_a(\mathbf{p})). \tag{12}$$

Rationale Behind the Hybrid Representation. As shown in Sec. 3, both CP and SA decomposition has a significant advantage in space complexity, which is our main motivation of using such low-rank representations in SLAM. While the SA decomposition has better representational power than CP decomposition, we empirically find that using the hybrid representation, i.e., CP for geometry and SA for appearance, yields better performance in RGB-D SLAM, as shown in Fig. 6. We identify two key reasons for this. Firstly, the geometry of a scene typically contains lower frequency information than its appearance, making the CP decomposition sufficient for representing geometries in many cases. Second, during optimization, CP decomposition converges faster than SA decomposition due to its simplicity. We find that the early convergence of geometry helps the appearance optimization because of the geometric dependency of color volume rendering in Eq. 6. Fig. 6 shows the ablation study that supports our choice of hybrid representation.

5 Experiments

We evaluate the effectiveness of our novel compact decomposition method for dense visual SLAM systems on various publicly available real and synthetic datasets. We also conduct a detailed ablation study to support the feasibility of our design choice in terms of speed and accuracy.

5.1 Experimental Setup

Datasets. We conduct our experiments on the following three widely-used dense visual SLAM benchmarks: Replica [23], ScanNet [7], and TUM RGB-D [24]. We evaluate the localization performance, i.e., camera tracking errors, for all the three datasets. In addition, we evaluate the reconstruction performance using the Replica dataset which provides the ground-truth geometries. We preprocess the datasets in the same way used in recent work [14, 32].

Implementation Details. In our experiments, we use two-layer MLPs (with 64 input channels and 16 hidden layer channels) for our decoders. For SDF-based volume rendering on a small-scale Replica [23], we set $N_s = 32$ and $N_t = 8$, which are sampled by stratified and importance sampling, respectively. We use $N_s = 48$ and $N_t = 8$, which are sampled similarly for the other datasets. Our coarse axis-aligned feature tensors use a resolution of 24 cm for both geometry and appearance, while the fine feature tensors use 6 cm and 3 cm for geometry and appearance, respectively. All axis-aligned features are set to have 32-dimensional tensors. Lastly, we set $k_g = 2$ for CP decomposition and $k_a = 16$ for SA decomposition. We provide more implementation details in the supplemental material.

Evaluation Metrics. We follow recent work [14, 32] for evaluation metrics. For the scene geometry evaluation, we use both 2D and 3D metrics, where we use the L1 loss on depth maps from ground truth and reconstructed meshes across 1000

randomly sampled camera poses. Further, for 3D metrics, we use reconstruction accuracy (in cm), reconstruction completion (in cm), and completion ratio. For a fair comparison, the volume resolution is set to 1cm, and we remove unseen regions that are not visible from any camera frustum. In addition, we use the Absolute Trajectory Error (ATE, [24]) RMSE and Mean to evaluate localization. By default, we run five independent runs and report the average results unless otherwise stated.

5.2 Analysis of Computation and Memory Efficiency

We start by evaluating the computation cost and memory efficiency. As we summarize in Table 1, we report the average frame processing time (FPT), the number of parameters for scene geometry and appearance feature planes, and memory complexity regarding big-O notation. We compare ours with other state-of-the-art approaches, such as NICE-SLAM [32] and ESLAM [14].

Table 1: Computation and Memory Efficiency. We compare runtime and memory efficiency between other state-of-the-art approaches, including NICE-SLAM [32] and ESLAM [14]. We measure the frame processing time (FPT), number of parameters for scene geometry and appearance feature planes. We assume a spatial resolution of $L \times L \times L$.

Data	Method	FPT(s)	# of Parameters			Complexity
			f_g	f_a	Total	
Replica	NICE-SLAM [32]	2.27	6.42M	5.70M	12.18M	O(L^3)
	ESLAM [14]	0.23	1.40M	5.38M	6.79M	O(L^2)
	LRSLAM (ours)	0.22	0.03M	0.83M	0.86M	O(L)
		(4.3%↓)	(97.9%↓)	(84.6%↓)	(87.3%↓)	
ScanNet	NICE-SLAM [32]	2.58	11.63M	10.36M	22.04M	O(L^3)
	ESLAM [14]	0.51	3.66M	13.98M	17.63M	O(L^2)
	LRSLAM (ours)	0.37	0.05M	1.33M	1.38M	O(L)
		(27.5%↓)	(98.6%↓)	(90.5%↓)	(90.1%↓)	
TUM RBG-D	NICE-SLAM [32]	13.21	23.56M	21.02M	44.64M	O(L^3)
	ESLAM [14]	4.56	1.40M	5.36M	6.77 M	O(L^2)
	LRSLAM (ours)	1.22	0.03M	0.81M	0.84M	O(L)
		(73.2%↓)	(97.9%↓)	(84.9%↓)	(87.6%↓)	

Note that we report scores for a scene `room0` of Replica [23], `scene0000` of ScanNet [7], and `fr1/desk` in TUM RGB-D [24] datasets. We use a single NVIDIA A100 GPU to measure such scores. As expected, our proposed LRSLAM uses a remarkably reduced number of parameters (87.3%–90.1% fewer parameters than ESLAM), while its processing time becomes faster (4.3%–73.2%) than ESLAM. Such gains become more apparent with real-world scenes (i.e., ScanNet and TUM RGB-D) than synthetic scenes (i.e., Replica).

5.3 Evaluation of Mapping and Localization Performance

Evaluation on ScanNet [7]. We compare reconstruction and localization accuracy on large real scenes from ScanNet [7] dataset with existing state-of-the-art approaches, including NICE-SLAM [32] and ESLAM [14]. In Fig. 4, we provide a qualitative analysis of camera localization and geometry reconstruction.

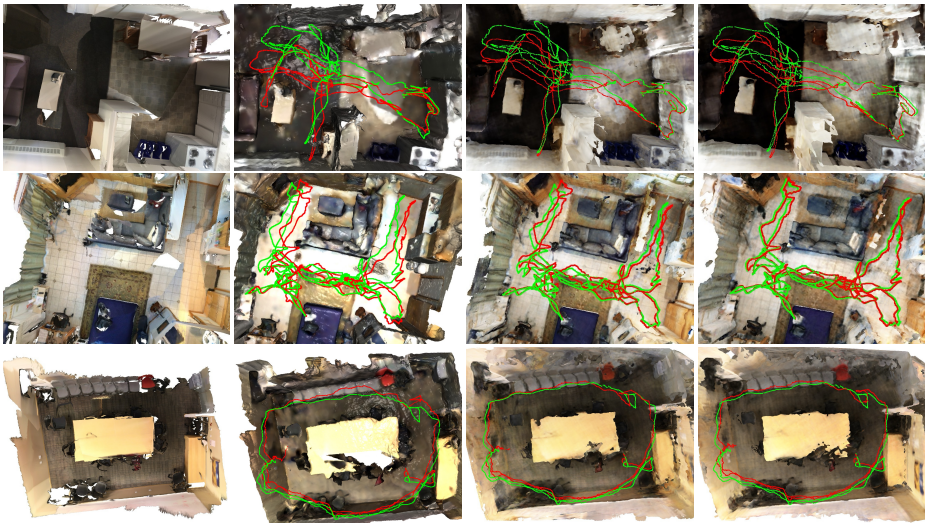

Fig. 4: Reconstruction and Localization on ScanNet [7]. We qualitatively compare the quality of 3D reconstruction and localization between our LRSLAM and existing approaches, including NICE-SLAM [32] and ESLAM [14]. For localization, the green trajectory is from the ground truth, and the red is the tracking results from each method. Despite using ten times more compact representation, our approach provides matched reconstruction and localization performance.

Table 2: Localization Results on ScanNet [7]. We compare with existing approaches, including NICE-SLAM [32] and ESLAM [14] in terms of ATE Mean and ATE RMSE. Our proposed method shows generally better results than the others.

Method	ATE	Sc. 0000	Sc. 0059	Sc. 0106	Sc. 0169	Sc. 0181	Sc. 0207	Avg.
NICE-SLAM [32]	Mean↓	9.9 ± 0.4	11.9 ± 1.8	7.0 ± 0.2	9.2 ± 1.0	12.2 ± 0.3	5.5 ± 0.3	9.3 ± 0.7
	RMSE↓	12.0 ± 0.5	14.0 ± 1.8	7.9 ± 0.2	10.9 ± 1.1	13.4 ± 0.3	6.2 ± 0.4	10.7 ± 0.7
ESLAM [14]	Mean↓	6.5 ± 0.1	6.4 ± 0.4	6.7 ± 0.1	5.9 ± 0.1	8.3 ± 0.2	5.4 ± 0.1	6.5 ± 0.2
	RMSE↓	7.3 ± 0.2	8.5 ± 0.5	7.5 ± 0.1	6.5 ± 0.1	9.0 ± 0.2	5.7 ± 0.1	7.4 ± 0.2
LRSLAM (ours)	Mean↓	5.2 ± 0.2	6.1 ± 0.1	6.7 ± 0.1	5.6 ± 0.1	7.6 ± 0.1	5.2 ± 0.1	6.1 ± 0.1
		(20.0%↓)	(4.7%↓)	(0.0%↓)	(5.1%↓)	(8.4%↓)	(3.7%↓)	(6.2%↓)
	RMSE↓	5.8 ± 0.4	8.2 ± 0.1	7.6 ± 0.1	6.5 ± 0.1	8.4 ± 0.1	5.6 ± 0.1	7.0 ± 0.2
		(20.5%↓)	(3.5%↓)	(1.3%↑)	(0.0%↓)	(6.7%↓)	(1.8%↓)	(5.4%↓)

Compared to the ground truth trajectory (green lines), the tracking results from our method are comparable or better performance without showing any large drifting, which confirms that our model can reconstruct precise geometry and detailed appearance with a ten times compact representation. Further, in Table 2, we provide quantitative analysis of localization on the same dataset in terms of ATE Mean and ATE RMSE for six large real scenes. We run five times for each

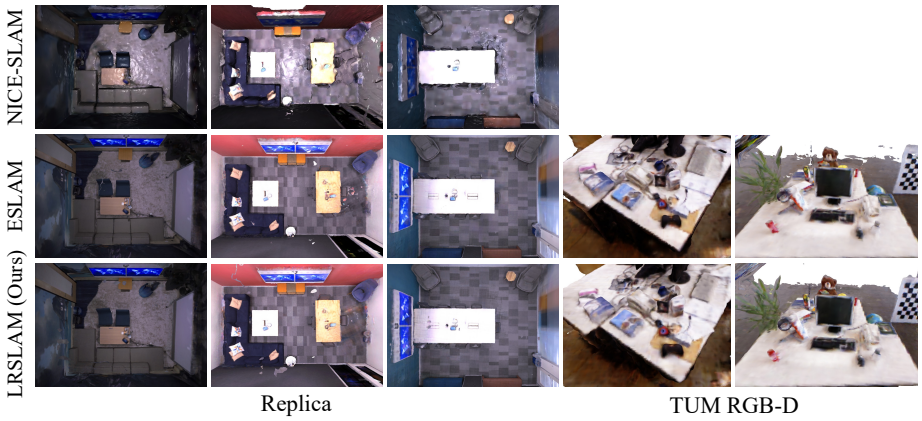

Fig. 5: Reconstruction on (a) TUM RGB-D [24] and (b) Replica [23]. We qualitatively compare the quality of 3D reconstruction between our LRSLAM, NICE-SLAM [32], and ESLAM [14]. Despite using ten times more compact representation, our approach provides comparable reconstruction performance in both small-scale datasets.

method, reporting the average and standard deviation. Note that we only report localization results as the dataset's ground truth meshes are incomplete. Our quantitative analysis also validates that our proposed LRSLAM generally outperforms the other state-of-the-art methods in localization, with a much lower standard deviation (thus, more stable). This confirms that our compact representation can provide significant gains in computations and memory efficiency without any degradation in localization and reconstruction (in fact, ours achieves better localization).

Evaluation on TUM RGB-D [24]. Further, we compare the localization accuracy on small real-world scenes from the TUM RGB-D [24] dataset. As shown in Table 3, we compare ATE RMSE scores with existing approaches, NICE-SLAM [32] and ESLAM [14]. In our experiments, all methods show reasonable reconstruction performance (i.e., ATE RMSE $\leq$ 3), where ours outperforms the others in two scenes (0.8%–13.5% improvements). In addition, as the dataset does not provide ground truth mesh, we provide a qualitative analysis of geometry reconstruction in Fig. 5 (a). Importantly, as we reported earlier in Table 1, our LRSLAM requires fewer parameters (12.4% of ESLAM), but shows comparable (or better in some scenes) geometry reconstruction quality.

Table 3: Localization Results on TUM RGB-D [24]. We compare with existing approaches, i.e., NICE-SLAM [32] and ESLAM [14] in terms of ATE RMSE. All methods generally show reasonable performance, while our proposed method shows reasonable or better (in two scenes) results.

	fr1/desk	fr2/xyz	fr3/office
NICE-SLAM [32]	2.85	2.39	3.02
ESLAM [14]	2.47	1.11	**2.42**
LRSLAM (ours)	**2.45**	**0.96**	2.79
	(0.8%↓)	(13.5%↓)	(15.3%↑)

Table 4: Reconstruction and Localization Results on Replica [23]. We compare with existing approaches, including NICE-SLAM [32] and ESLAM [14] regarding models' reconstruction and localization performance. The Replica dataset uses synthetic scenes with ground truth depth information, which might be unrealistic for real-world conditions. Thus, we also report scores of models with noisy depth inputs, where we add Gaussian noise $\mathcal{N}(0, 0.05^2)$ in scene `room0`. Our method shows robust performance in both reconstruction and localization.

Method	Depth Noise Added	Reconstruction Error (in cm)				Localization Error (in cm)	
		Depth L1↓	Acc.↓	Comp.↓	Comp. Ratio (%)↑	ATE Mean↓	ATE RMSE↓
NICE-SLAM [32]	-	3.29 ± 0.33	1.66 ± 0.07	1.63 ± 0.05	96.74 ± 0.36	1.56 ± 0.29	2.05 ± 0.45
ESLAM [14]	-	1.18 ± 0.05	0.97 ± 0.02	1.05 ± 0.01	98.60 ± 0.07	0.52 ± 0.03	0.63 ± 0.05
LRSLAM (ours)	-	1.58 ± 0.11	1.00 ± 0.03	1.07 ± 0.03	98.94 ± 0.12	0.61 ± 0.04	0.79 ± 0.05
ESLAM [14]	$\mathcal{N}(0, 0.05^2)$	3.22	2.04	1.99	98.35	3.00	2.64
LRSLAM (ours)	$\mathcal{N}(0, 0.05^2)$	1.20	1.06	1.18	98.28	2.94	2.57
		(62.7%↓)	(48.0%↓)	(40.7%↓)	(0.1%↓)	(2.0%↓)	(2.7%↓)

Evaluation on Replica [23]. We additionally compare the reconstruction and localization performance with other existing approaches, i.e., ESLAM [14] and NICE-SLAM [32], on small-scale synthetic scenes from the Replica [23] dataset. We observe in Table 4 (see top three rows) that ESLAM and ours show reasonably good reconstruction and localization accuracy, though ours shows slightly lower scores than ESLAM, which is probably due to using a ground truth depth (i.e., synthetic) in optimizing a small-scale scene. ESLAM is more expressive than ours, tending to be overfitted to each synthetic scene easily. Thus, to make the problem more realistic, we also conduct the same experiment but with depth noise added. We observe ESLAM suffers from that noise, but ours shows robustness in both reconstruction and localization, clearly outperforming ESLAM. From these experiments, we can reason that our low-rank representations of the scene has the ability to remove or filter the sensor noise, which is critical in RGB-D SLAM systems.

5.4 Ablation Study

Comparison between Different Combinations of Scene Representation. In this paper, we advocate for using a combination of our proposed Six-axis decomposition and conventional compact CP decomposition, which offer an efficient memory complexity of $O(n)$, while the Tri-plane representation provides $O(n^2)$. In Fig. 6, we experiment to compare localization accuracy between variant models with different combinations of scene representations. In this experiment, we use large-scale real-world scenes from ScanNet [7] and evaluate their performance regarding ATE RMSE, the number of learnable parameters, and Frame Processing Time (FPT). As expected, Six-axis decomposition and CP decomposition clearly win in terms of representational compactness (their parameters are 4%–6% and 8%–12% less than the Tri-plane representation, re-

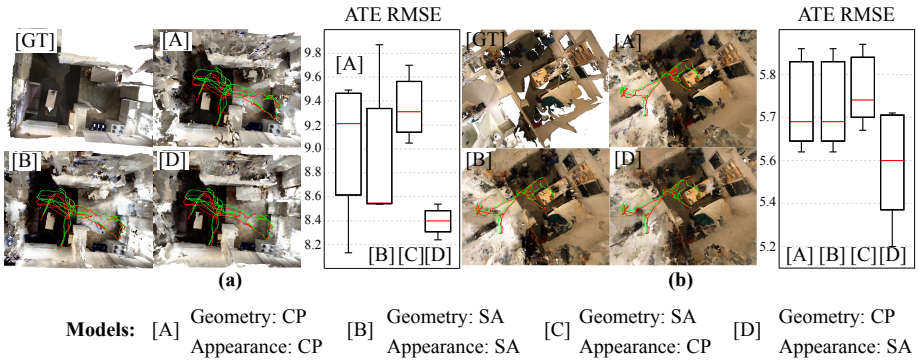

Fig. 6: Ablation Study. Localization and reconstruction accuracy comparison between variants of our model with different combinations of scene geometry and appearance representations, i.e., CP decomposition and Six-axis (SA) decomposition. For example, the model [A] uses CP decomposition both for scene geometry and appearance representation. Note that the model [D] is ours. We visualize the results of two notable scenes from ScanNet [7] dataset as well as their box plots of ATE RMSE for independent five runs. More examples are provided as supplemental material.

spectively), mostly showing better or matched performance in localization and reconstruction accuracy. However, CP decomposition suffers from robustly reconstructing complex scenes (see variances of CP-CP model, i.e., model [A]), which is probably due to its high compactness. This may necessitate a hybrid decomposition to compensate for this trade-off relation.

6 Conclusion

In this paper, we presented a novel dense visual SLAM approach called LRSLAM. This approach leverages a compact scene representation based on a combination of our newly proposed Six-axis decomposition (which factorizes the three planes in the tri-plane representation into six axis-aligned feature vectors, thus holding an efficient memory complexity of $O(n)$) and conventional CP decomposition. Our experiments on three widely-used public benchmarks (i.e., ScanNet, TUM RGB-D, and Replica) validate that our proposed method indeed uses remarkably fewer parameters and shows faster processing time than existing state-of-the-art approaches, retaining matched or improved reconstruction and localization accuracy.

Acknowledgment. This work was partly supported by IITP under the Leading Generative AI Human Resources Development(IITP-2024-RS-2024-00397085, 30%) grant, IITP grant (RS-2022-II220043, Adaptive Personality for Intelligent Agents, 30% and IITP-2024-2020-0-01819, ICT Creative Consilience program, 10%). This work was also partly supported by Basic Science Research Program through the NRF funded by the Ministry of Education(NRF-2021R1A6A1A13044830, 30%).

References

1. Bresson, G., Alsayed, Z., Yu, L., Glaser, S.: Simultaneous localization and mapping: A survey of current trends in autonomous driving. IEEE Transactions on Intelligent Vehicles **2**(3), 194–220 (2017)
2. Carroll, J.D., Chang, J.J.: Analysis of individual differences in multidimensional scaling via an n-way generalization of "eckart-young" decomposition. Psychometrika **35**(3), 283–319 (1970)
3. Chan, E.R., Lin, C.Z., Chan, M.A., Nagano, K., Pan, B., De Mello, S., Gallo, O., Guibas, L.J., Tremblay, J., Khamis, S., et al.: Efficient geometry-aware 3d generative adversarial networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16123–16133 (2022)
4. Chen, A., Xu, Z., Geiger, A., Yu, J., Su, H.: Tensorf: Tensorial radiance fields. In: European Conference on Computer Vision. pp. 333–350. Springer (2022)
5. Chen, X., Milioto, A., Palazzolo, E., Giguere, P., Behley, J., Stachniss, C.: Suma++: Efficient lidar-based semantic slam. In: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 4530–4537. IEEE (2019)
6. Covolan, J.P.M., Sementille, A.C., Sanches, S.R.R.: A mapping of visual slam algorithms and their applications in augmented reality. In: 2020 22nd Symposium on Virtual and Augmented Reality (SVR). pp. 20–29. IEEE (2020)
7. 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. pp. 5828–5839 (2017)
8. Dworakowski, D., Thompson, C., Pham-Hung, M., Nejat, G.: A robot architecture using contextslam to find products in unknown crowded retail environments. Robotics **10**(4), 110 (2021)
9. Fuentes-Pacheco, J., Ruiz-Ascencio, J., Rendón-Mancha, J.M.: Visual simultaneous localization and mapping: a survey. Artificial intelligence review **43**, 55–81 (2015)
10. Gui, J., Gu, D., Wang, S., Hu, H.: A review of visual inertial odometry from filtering and optimisation perspectives. Advanced Robotics **29**(20), 1289–1301 (2015)
11. Huang, G.: Visual-inertial navigation: A concise review. In: 2019 international conference on robotics and automation (ICRA). pp. 9572–9582. IEEE (2019)
12. Huang, H., Li, L., Cheng, H., Yeung, S.K.: Photo-slam: Real-time simultaneous localization and photorealistic mapping for monocular, stereo, and rgb-d cameras. arXiv preprint arXiv:2311.16728 (2023)
13. Jinyu, L., Bangbang, Y., Danpeng, C., Nan, W., Guofeng, Z., Hujun, B.: Survey and evaluation of monocular visual-inertial slam algorithms for augmented reality. Virtual Reality & Intelligent Hardware **1**(4), 386–410 (2019)
14. Johari, M.M., Carta, C., Fleuret, F.: Eslam: Efficient dense slam system based on hybrid representation of signed distance fields. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 17408–17419 (2023)
15. Keetha, N., Karhade, J., Jatavallabhula, K.M., Yang, G., Scherer, S., Ramanan, D., Luiten, J.: Splatam: Splat, track & map 3d gaussians for dense rgb-d slam. arXiv preprint arXiv:2312.02126 (2023)
16. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. ACM Transactions on Graphics **42**(4) (2023)
17. Li, M., Liu, S., Zhou, H.: Sgs-slam: Semantic gaussian splatting for neural dense slam. arXiv preprint arXiv:2402.03246 (2024)
18. Liu, C., Zhou, C., Cao, W., Li, F., Jia, P.: A novel design and implementation of autonomous robotic car based on ros in indoor scenario. Robotics **9**(1), 19 (2020)

19. Matsuki, H., Murai, R., Kelly, P.H., Davison, A.J.: Gaussian splatting slam. arXiv preprint arXiv:2312.06741 (2023)
20. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. Communications of the ACM **65**(1), 99–106 (2021)
21. Ruan, K., Wu, Z., Xu, Q.: Smart cleaner: A new autonomous indoor disinfection robot for combating the covid-19 pandemic. Robotics **10**(3), 87 (2021)
22. Singandhupe, A., La, H.M.: A review of slam techniques and security in autonomous driving. In: 2019 third IEEE international conference on robotic computing (IRC). pp. 602–607. IEEE (2019)
23. Straub, J., Whelan, T., Ma, L., Chen, Y., Wijmans, E., Green, S., Engel, J.J., Mur-Artal, R., Ren, C., Verma, S., et al.: The replica dataset: A digital replica of indoor spaces. arXiv preprint arXiv:1906.05797 (2019)
24. Sturm, J., Engelhard, N., Endres, F., Burgard, W., Cremers, D.: A benchmark for the evaluation of rgb-d slam systems. In: 2012 IEEE/RSJ international conference on intelligent robots and systems. pp. 573–580. IEEE (2012)
25. 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. pp. 6229–6238 (2021)
26. Taketomi, T., Uchiyama, H., Ikeda, S.: Visual slam algorithms: A survey from 2010 to 2016. IPSJ Transactions on Computer Vision and Applications **9**(1), 1–11 (2017)
27. Tosi, F., Zhang, Y., Gong, Z., Sandström, E., Mattoccia, S., Oswald, M.R., Poggi, M.: How nerfs and 3d gaussian splatting are reshaping slam: a survey. arXiv preprint arXiv:2402.13255 (2024)
28. Yan, C., Qu, D., Wang, D., Xu, D., Wang, Z., Zhao, B., Li, X.: Gs-slam: Dense visual slam with 3d gaussian splatting. arXiv preprint arXiv:2311.11700 (2023)
29. Yousif, K., Bab-Hadiashar, A., Hoseinnezhad, R.: An overview to visual odometry and visual slam: Applications to mobile robotics. Intelligent Industrial Systems **1**(4), 289–311 (2015)
30. Yugay, V., Li, Y., Gevers, T., Oswald, M.R.: Gaussian-slam: Photo-realistic dense slam with gaussian splatting. arXiv preprint arXiv:2312.10070 (2023)
31. Zhang, S., Zheng, L., Tao, W.: Survey and evaluation of rgb-d slam. IEEE Access **9**, 21367–21387 (2021)
32. 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. pp. 12786–12796 (2022)