2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

PROFusion: Robust and Accurate Dense Reconstruction via Camera Pose Regression and Optimization

Siyan Dong*, Zijun Wang, Lulu Cai, Yi Ma, and Yanchao Yang
The University of Hong Kong

***Abstract*— Real-time dense scene reconstruction during unstable camera motions is crucial for robotics, yet current RGB-D SLAM systems fail when cameras experience large viewpoint changes, fast motions, or sudden shaking. Classical optimization-based methods deliver high accuracy but fail with poor initialization during large motions, while learning-based approaches provide robustness but lack sufficient accuracy for dense reconstruction. We address this challenge through a combination of learning-based initialization with optimization-based refinement. Our method employs a camera pose regression network to predict metric-aware relative poses from consecutive RGB-D frames, which serve as reliable starting points for a randomized optimization algorithm that further aligns depth images with the scene geometry. Extensive experiments demonstrate promising results: our approach outperforms the best competitor on challenging benchmarks, while maintaining comparable accuracy on stable motion sequences. The system operates in real-time, showcasing that combining simple and principled techniques can achieve both robustness for unstable motions and accuracy for dense reconstruction. Code released: `https://github.com/siyandong/PROFusion`.**

I. INTRODUCTION

Real-time camera tracking and dense scene reconstruction are fundamental problems in robotics and computer vision. For autonomous robots, handling unstable camera motions is both challenging and critical. Current RGB-D SLAM (Simultaneous Localization and Mapping) systems perform well in controlled environments with smooth, typically slow camera movements. However, they struggle with the unstable motions encountered in practical applications like exploration or rescue missions - situations where robust camera pose estimation is essential for dense reconstruction.

Since the pioneering work of KinectFusion [1], [2], the past decade has witnessed significant progress in RGB-D SLAM systems, particularly in the development of scene representations and camera pose estimation methods. Existing research has explored various representations such as volumetric [3], [4], point-based [5], [6], and neural representations [7], [8], [9]. Building upon these representations, camera pose estimation is commonly performed through geometric optimization [10], [11] in combination with photometric losses [12], [7]. While they can achieve high accuracy, they inherently require smooth and relatively slow camera motions - a limitation that restricts their widespread deployment in robotics. ROSEFusion [13] recently introduced randomized optimization to better handle relatively fast camera motions, but it still struggles with rapid motions such as large in-place rotations. When cameras undergo large viewpoint changes, finding suitable initial poses becomes challenging. This makes the optimization process struggle to find the global optimum or even fail to converge.

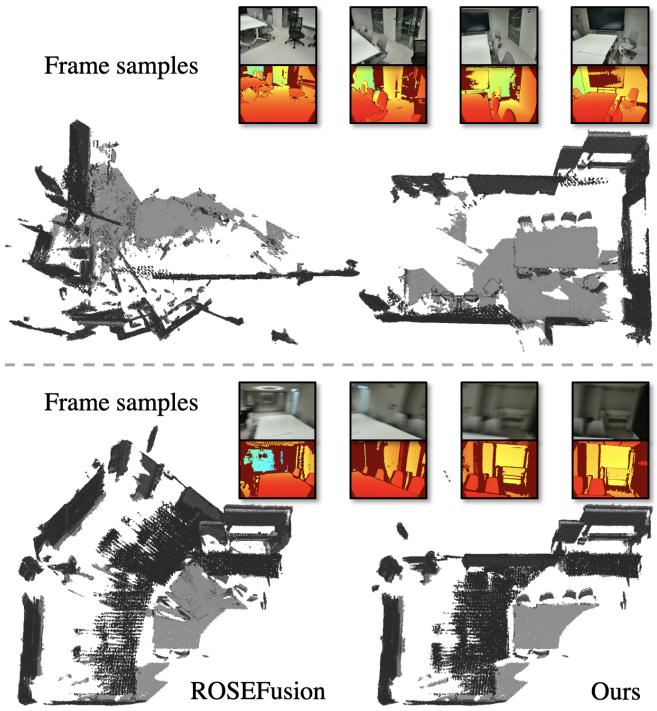

Fig. 1: Dense scene reconstruction under challenging camera motions. Two representative sequences demonstrate failure cases where state-of-the-art methods like ROSEFusion [13] (left) produce corrupted reconstructions due to unstable camera motions involving large translations and fast in-place rotations. Our approach (right), which combines camera pose regression and optimization, successfully reconstructs accurate scene layouts under challenging conditions, demonstrating superior robustness to camera motion instability.

Unlike classical optimization methods, which can be sensitive to large motions, recent learning-based pose estimation approaches powered by large foundation models [14], [15] offer much greater robustness. Thanks to scaling up the training on mixtures of diverse datasets, they demonstrate excellent generalization ability. Among them, camera pose regression networks [16], [17] achieve state-of-the-art success rates while maintaining fast inference times. However, existing works rarely take metric depth images as input, and the predicted poses are accurate only up to unknown scales.

* Email: siyan3d@hku.hk

As a result, they often fall short of the tracking accuracy (e.g., millimeter-level error per frame) that classical optimization methods can achieve.

In this paper, we present a practical 3D reconstruction framework that combines the robustness of learning-based pose estimation with the accuracy of classical optimization. We integrate a metric-aware camera pose regression network with a randomized optimization algorithm to create an effective dense reconstruction system that achieves impressive results, despite its simplicity. We validate our approach through comprehensive experimental evaluations (e.g., real-world samples shown in Figure 1). Our key contributions can be summarized as follows:

- Our findings show that a camera pose regression network can reliably predict an initial coarse pose, which then serves as a starting point for further refinement using randomized optimization.
- Based on our findings, we have developed a real-time dense scene reconstruction system that provides robust and accurate camera tracking and scene reconstruction, regardless of camera motion stability.
- Extensive experiments demonstrate that our system achieves accuracy on par with state-of-the-art dense reconstruction systems under stable camera motions, while offering substantially better robustness when handling camera shake, fast, and large motions.

II. RELATED WORKS

A. Dense scene reconstruction with RGB-D SLAM

KinectFusion [2], [1] pioneered real-time dense scene reconstruction with camera tracking using depth sensors. Subsequent research have introduced scalable scene representations [3], [4], improved registration algorithms [11], [18], and integrated color image feature matching [19], [20], [9]. A core problem in RGB-D based reconstruction is accurate camera pose estimation. Registration errors can accumulate and cause drift in large scenes. Classical systems like InfiniTAM [21], ElasticFusion [5], and BundleFusion [20] address this problem by integrating loop closure and bundle adjustment to achieve globally consistent reconstruction.

Recent SLAM systems have adopted neural scene representations [7], [8], [9], enabling photo-realistic novel view synthesis. The photometric losses used in these systems can refine camera poses and reduce error, but they may be sensitive to noise like motion blur or exposure changes during fast motions. As a result, their robustness still depends on bundle adjustment or additional backend systems [22].

There are also non-rigid [23] or semantic reconstruction [24] approaches that are beyond our focus in this paper.

B. Randomized optimization for fast camera motions

A key limitation of the approaches reviewed in Section II-A is their requirement for smooth and typically very slow camera motions. This poses challenges for robotic applications, such as exploration and rescue scenarios, where cameras often shake on rugged terrain and move fast. Such conditions cause motion blur and significant viewpoint changes between frames, making camera tracking difficult. ROSEFusion [13] uses randomized optimization to address nonlinearity from viewpoint changes. The intuition is to apply a wide range of random search to find solutions. However, as relative poses increase in magnitude, the required search space grows proportionally, making the method impractical for large motions. A typical failure case occurs with fast in-place rotation - a common motion in robots.

More recent works like MIPS-Fusion [25] and Remix-Fusion [26] combine randomized optimization with neural representations. While achieving promising scalability and rendering quality, these approaches sacrifice robustness. In this paper, we focus on achieving robustness and accuracy for unstable camera movements - particularly the large viewpoint changes caused by fast motion, low frame rates, or signal losses. Our experiments indicate that ROSEFusion [13] remains the strongest competitor in this field.

C. Large 3D foundation models

DUSt3R [14] was the first to leverage a large mixture of public datasets to train a large geometric foundation model, demonstrating that high-quality 3D reconstruction and remarkable generalization capability can emerge through the scaling laws. It triggered a surge of neural networks for geometric reasoning, quickly producing numerous research outcomes in areas such as keypoint matching [15], camera pose estimation [16], and multi-view 3D reconstruction [27], [28], [29], [30], [31], [32], [33], [17], [34], [35].

Among these works, SLAM3R [32] is the only one compatible with point cloud input, and Reloc3r [16] provides the simplest yet most efficient approach for camera pose estimation. However, their predictions are only accurate up to unknown scale factors. This paper draws inspiration from point cloud embedding [32] but utilizes metric point clouds obtained from depth scanning. By integrating this approach with camera pose regression [16], we create a simple yet robust metric camera pose estimation network.

III. METHOD

Problem setting. Our system takes an RGB-D video as input, consisting of consecutive frames $\{F_t = (C_t, D_t)\}_{t=1}^{T}$ captured by a moving camera in a static scene. We assume aligned color $C_t \in \mathbb{R}^{H\times W\times 3}$ and depth $D_t \in \mathbb{R}^{H\times W\times 3}$ images for each frame, with known camera intrinsic parameters. Therefore, each depth image D_t can be projected to a metric point cloud $M_t \in \mathbb{R}^{H\times W\times 3}$. Our goal is to recover the camera poses $\{P_t = [\mathbf{R}_t|\mathbf{t}_t]\}_{t=1}^{T}$ (where $\mathbf{R} \in SO(3)$ is the rotation matrix and $\mathbf{t} \in \mathbb{R}^3$ represents the translation in the world coordinate system) of the frames and reconstruct the dense 3D geometry of the captured scene. To simplify notation, we denote the camera pose in the world coordinate system as P_i and relative pose from frame F_i to F_j as $P_{(i,j)}$.

Method overview. Figure 2 illustrates the pipeline of our system. Following KinectFusion [1], [2], we represent the scene using truncated signed distance function (TSDF) [36]. This is implemented with a tensor that records the distance value in each 3D grid, denoting the distance to its closest

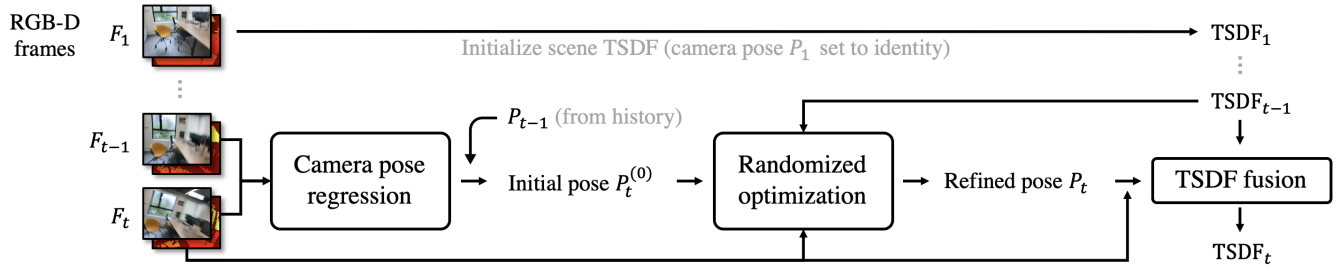

Fig. 2: System overview. We use the first frame (set to identity pose) to initialize the scene represented by TSDF grids. The following frames are incrementally fused to the scene through a two-step process: first, a coarse registration with the previous frame via camera pose regression, and second, a fine-grained alignment to the TSDF via a randomized optimization algorithm. The aligned frames then update the TSDF values by modifying known grids and filling in new ones. Through this process, both camera motion and scene geometry are progressively reconstructed.

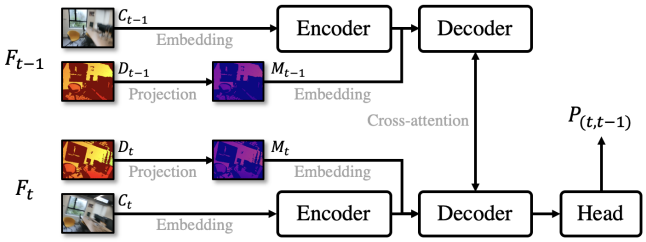

Fig. 3: Network architecture. It takes a pair of consecutive RGB-D frames as input and outputs the relative camera pose that aligns the second frame to the first one. The color and depth (converted to metric point clouds) images are divided into tokens and fed into a Transformer backbone with a pose regression head to infer the relative transformation matrix.

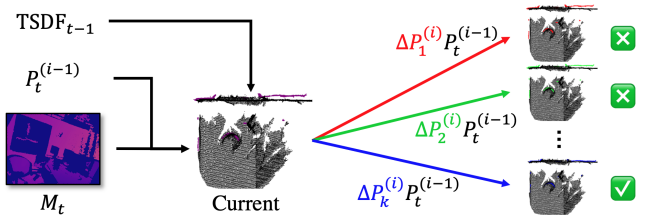

Fig. 4: Illustration of the camera pose searching process in iteration i in our randomized optimization. We multiply a set of delta poses $\{\Delta P_k^{(i)}\}_{k=1}^K$ to the current pose $P_t^{(i-1)}$ and evaluate their fitness to TSDF_{t-1}. Delta poses with better alignment are collected in an advantage set to update the current pose and the search size for the next iteration.

surface. The first input frame is set to the identity pose, and its depth image initializes the TSDF values. The key feature of our system is robust and accurate camera pose estimation, which combines learning-based initialization with optimization-based refinement. Starting from the second frame, each frame F_t is paired with its previous frame F_{t-1} and fed into a neural network to estimate an initial camera pose, denoted by $P_t^{(0)}$. This pose serves as the starting point for aligning the depth image D_t to the scene TSDF_{t-1} through randomized optimization. The optimized pose, denoted by P_t, is recorded for future pose estimation, and the scene is updated to TSDF_t after fusing F_t. The fusion process remains the same as KinectFusion [1], [2]. In the following sections, we first present our camera pose regression network in Section III-A. Next, we introduce our adapted randomized optimization algorithm in Section III-B. Finally, we report implementation details in Section IV-A.

A. Camera Tracking Initialization via Pose Regression

Our pose regression network takes a pair of consecutive RGB-D frames (F_{t-1}, F_t) as input and outputs the relative camera pose $P_{(t,t-1)}$. The absolute pose $P_t^{(0)}$ in the world coordinate system is then calculated by multiplying this relative pose with the previous pose estimation P_{t-1}, denoted as: $P_t^{(0)} = P_{t-1}P_{(t,t-1)}$.

Our network architecture draws inspiration from recent foundation models [14], [32], [16]. For the sake of simplicity, we adopt DUSt3R [14]'s backbone with minimal modifications. The network architecture is illustrated in Figure 3. It consists of a two-branch Vision Transformer (ViT) [37]. The ViT encoder extracts features from each image, and the decoder exchanges information between branches. Finally, a regression head outputs a matrix as the relative pose.

Frame embedding. Each color image C_i undergoes the default ViT patch embedding process to produce color tokens $\mathbf{C}_i$. Each depth image is first projected to a metric point cloud M_i using known camera intrinsics. This point cloud then undergoes the same patch embedding process (without normalization) to extract metric-aware geometry tokens $\mathbf{M}_i$.

ViT backbone. The color tokens are fed into m ViT encoder blocks for feature extraction: $\mathbf{C}'_i = \text{Encoder}(\mathbf{C}_i)$. To preserve the metric information, we do not apply an encoding process to geometry tokens $\mathbf{M}_i$. These tokens are directly added to the encoded color tokens to form feature tokens $\mathbf{Z}_i = \mathbf{C}'_i + \mathbf{M}_i$ for the following decoding process. Unlike the encoder blocks that operate on tokens in each frame separately, the n decoder blocks incorporate cross-attention layers to reason the spatial transformation between the two sets of tokens. We denote the decoded tokens as: $\mathbf{Z}'_t = \text{Decoder}(\mathbf{Z}_t, \mathbf{Z}_{t-1})$.

Pose regression. Our pose head remains similar to Reloc3r [16]'s regression head. The architecture is the same; the only difference is that our translation is set to be metric,

while theirs only predicts direction. Our pose regression process can be denoted as: $P_{(t,t-1)} = \text{Head}(\mathbf{Z}_t^l)$.

Supervision. We train our network using supervision on metric relative poses. The loss function minimizes the angular error [16] in rotation and the distance in translation:

$$\mathcal{L} = \ell_{\mathbf{R}} + \ell_{\mathbf{t}},$$
$$\ell_{\mathbf{R}} = \arccos(\frac{\text{tr}(\bar{\mathbf{R}}^{-1}\mathbf{R}) - 1}{2}), \quad \text{and} \quad \ell_{\mathbf{t}} = \text{norm}(\bar{\mathbf{t}} - \mathbf{t}).$$

Here, $\bar{\mathbf{R}}$ and $\bar{\mathbf{t}}$ represent the ground truth rotation and translation of the relative pose, respectively. Our network is trained on a mixture of public datasets and generalizes well. Details are reported in Section IV-A.

B. Pose Refinement via Randomized Optimization

Our randomized optimization algorithm takes the current scene TSDF_{t-1}, point cloud M_t, and the initial pose $P_t^{(0)}$ as input, then iteratively searches for incremental relative poses (which we call delta poses) that can better align M_t to TSDF_{t-1}. Note that we do not use color images in this process. Under unstable camera motions, color images can become blurred, thereby reducing accuracy. While depth images can be incomplete, their valid pixels remain accurate. Building on this property, our algorithm draws inspiration from ROSEFusion [13], but is simplified as detailed below.

Let's denote the search size as $s = (\omega, v)$, representing ω degree and v cm searching range. Our algorithm iteratively updates the current pose $P_t^{(i-1)}$ and search size $s^{(i)}$. In each iteration i, as illustrated in Figure 4, we have a current pose $P_t^{(i-1)}$ from previous iterations. We uniformly sample a set of delta poses $\{\Delta P_k^{(i)}\}_{k=1}^K$ according to $s^{(i)}$. Next, we multiply each delta pose $\Delta P_k^{(i)}$ to $P_t^{(i-1)}$ to formulate a global transformation, $\Delta P_k^{(i)} P_t^{(i-1)}$, that aligns M_t to TSDF_{t-1}. We then evaluate if this transformation actually improves the alignment. We quantify this by measuring the geometric consistency error between the transformed point cloud and the current scene representation:

$$\mathcal{E}(\Delta P_k^{(i)} P_t^{(i-1)}) = \frac{1}{|X_k|} \sum_{x \in X_k} \frac{|\text{TSDF}_{t-1}(\Delta P_k^{(i)} P_t^{(i-1)} x)|}{\tau}.$$

Here, τ represents the truncated distance, x denotes a 3D points from M_t, and X_k represents the subset of points that fall in TSDF_{t-1}'s valid grids (i.e., grids observed by history frames and within distance τ) when applying the transformation $\Delta P_k^{(i)} P_t^{(i-1)}$. $|\text{TSDF}_{t-1}(\cdot)|$ represents the absolute distance value queried by the input point. This is normalized to the range [0, 1] by multiplying $1/\tau$. $|X_k|$ represents the number of points in X_k. Delta poses with lower errors (i.e., $\mathcal{E}(\Delta P_k^{(i)} P_t^{(i-1)}) < \mathcal{E}(P_t^{(i-1)})$) are collected in an advantage set. We compute an average (rotation in $SO(3)$ and translation in $\mathbb{R}^3$) delta pose within this set, denoted by $\Delta\hat{P}^{(i)}$. The current pose is updated by $P_t^{(i)} = \Delta\hat{P}^{(i)} P_t^{(i-1)}$, and the search size is updated by $s^{(i+1)} = \beta s^{(i)} + (1-\beta)\mathcal{E}(P_t^{(i)})\, s^{(i)}$. Here, $\beta = 0.1$ represents the momentum between iterations. As a result, the search size converges during iterations.

IV. EXPERIMENT

A. Implementation details

Camera pose regression network. This module is implemented in Python with PyTorch. The images input to our network are resized and center-cropped to a fixed resolution of 224×224 for efficiency. Following DUSt3R [14], we employ a 24-block encoder and a 12-block decoder. The network weights are initialized from SLAM3R [32]'s L2W model and then trained with a mixture of public datasets featuring indoor RGB-D scanning with known camera poses. The training data includes ScanNet++ [38], Aria Synthetic Environments [39], and few sequences from 7 Scenes [40] and Replica [41] to ensure diversity. This resulted in around 2 million RGB-D image pairs, with relative camera poses ranging from 0-180 degrees and 0-5 meters. It's important to note that all test scenes were excluded from the training data, and our network was trained only once to handle all tests. The network demonstrates strong generalization capability across various benchmarks and real-world applications.

Randomized optimization algorithm. This module is implemented in C++ with CUDA. The initial search size $s^{(1)}$ is set to (10 degree, 10 cm) by default. Following ROSEFusion [13], we alternate the number of delta poses (1024, 3072, and 10240) across iterations. Similarly, we downsample the input point cloud for efficiency, and alternate the sampling rates (1/8, 1/16, and 1/32). While we also set a maximum of 20 iterations, our optimization typically converges within just a few iterations.

B. Comparisons

For our comparison, we selected six representative dense scene reconstruction systems. The systems include three state-of-the-art classical dense fusion approaches (ElasticFusion [5], BundleFusion [20], and ROSEFusion[13]) and three neural SLAM systems (NICE-SLAM [7], MIPS-Fusion [25], and HERO-SLAM [9]). We test these methods on classical benchmarks with stable camera motions (TUM RGB-D [42]), but primarily focus on benchmarks with unstable motions: camera shaking scenarios from ETH3D [43] and fast motion from FastCaMo benchmarks [13], which include both synthetic and real-world RGB-D scans. The results are produced using publicly available code and papers. In the tables, a minus symbol ("-") indicates cases where the method failed to reconstruct at least 40% of the sequence or produced a completely incorrect trajectory misaligned with ground-truth. Additionally, we conduct real-world applications with an RGB-D camera, ORBBEC Femto Bolt.

TUM RGB-D [42]. We report camera tracking accuracy in Table III. The camera motions in these sequences are smooth and slow. Our method achieves comparable accuracy to the best-performing systems, ElasticFusion [5] and BundleFusion [20], which use global optimization with loop closures to correct accumulated drift. Note that NICE-SLAM [7], HERO-SLAM [9], and MIPS-Fusion [25] also employ bundle adjustment to jointly optimize camera poses across keyframes. Despite using only single-frame tracking,

TABLE I: Camera tracking accuracy evaluation using ATE-RMSE (cm) on FastCaMo-Synth (raw) benchmark.

Method	Apart._1	Apart._2	Frl_a._2	Hotel_0	Office_0	Office_1	Office_2	Office_3	Room_0	Room_1	Avg.
ElasticFusion [5]	23.2	56.8	12.6	36.6	19.2	21.9	-	80.0	-	43.6	-
BundleFusion [20]	3.0	1.1	2.8	54.4	0.8	-	-	-	1.5	-	-
ROSEFusion [13]	0.7	**0.6**	**0.5**	0.8	**0.5**	0.8	8.3	10.1	2.1	2.0	2.6
NICE-SLAM [7]	-	36.7	15.4	4.2	8.4	13.7	14.6	14.3	-	29.7	-
MIPS-Fusion [25]	7.0	1.5	1.9	4.8	3.6	5.6	7.4	17.4	4.4	5.1	5.9
HERO-SLAM [9]	3.7	0.7	0.7	1.7	0.7	1.1	13.4	24.6	7.6	**0.5**	5.5
Ours	**0.5**	**0.6**	**0.5**	**0.7**	**0.5**	**0.5**	**1.8**	**1.0**	**0.4**	**0.5**	**0.7**

TABLE II: Camera tracking accuracy evaluation using ATE-RMSE (cm) on FastCaMo-Synth (noise) benchmark.

Method	Apart._1	Apart._2	Frl_a._2	Hotel_0	Office_0	Office_1	Office_2	Office_3	Room_0	Room_1	Avg.
ElasticFusion [5]	40.9	40.7	43.8	43.8	22.3	2.3	-	94.3	-	31.0	-
BundleFusion [20]	4.6	2.2	83.6	65.2	2.7	17.3	-	-	-	-	-
ROSEFusion [13]	1.5	**0.9**	3.0	1.6	0.7	1.8	3.6	9.4	2.9	3.8	2.9
NICE-SLAM [7]	-	20.2	24.8	11.8	29.3	-	16.4	29.8	-	24.9	-
MIPS-Fusion [25]	6.6	3.1	**2.6**	5.2	7.6	17.4	24.9	6.0	4.4	3.6	8.1
HERO-SLAM [9]	3.7	1.2	7.5	1.6	1.2	1.7	14.7	23.7	11.7	**1.3**	6.8
Ours	**1.2**	**0.9**	2.9	**1.5**	**0.5**	**1.1**	**1.8**	**1.9**	**1.7**	**1.3**	**1.5**

TABLE III: Camera tracking accuracy evaluation using ATE-RMSE (cm) on stable motions from TUM RGB-D dataset. The methods marked with * represent only single-frame camera tracking w/o bundle adjustment or loop optimization.

Method	fr1_desk	fr2_xyz	fr3_office	Avg.
ElasticFusion [5]	2.0	**1.1**	**1.7**	**1.6**
BundleFusion [20]	**1.6**	**1.1**	2.2	**1.6**
ROSEFusion [13] *	2.3	3.3	3.9	3.1
NICE-SLAM [7]	2.7	1.8	3.0	2.5
MIPS-Fusion [25]	3.0	1.4	4.6	3.0
HERO-SLAM [9]	2.5	2.1	2.7	2.4
Ours *	2.0	2.3	2.5	2.3

TABLE IV: Camera tracking accuracy evaluation using ATE-RMSE (cm) on camera shaking scenes from ETH3D dataset.

Method	camera_shake_1	2	3	Avg.
ElasticFusion [5]	8.4	-	-	-
BundleFusion [20]	5.2	3.5	-	-
ROSEFusion [13]	0.9	1.8	5.0	2.6
NICE-SLAM [7]	1.0	-	-	-
MIPS-Fusion [25]	4.6	-	7.7	-
HERO-SLAM [9]	**0.7**	1.5	-	-
Ours	**0.7**	**1.2**	**3.6**	**1.8**

our system achieves slightly lower tracking errors. This validates the accuracy of our approach.

ETH3D [43]. Our method demonstrates its advantage when dealing with unstable camera motions. To evaluate this, we conduct a comparison on three challenging sequences from the ETH3D benchmark that feature camera shaking. Camera shake results in sudden speed changes and blurred color images. This variation in motion between frames can be significant, with some frames experiencing large viewpoint changes. The results are reported in Table IV. Among all compared methods, only ROSEFusion [13] and our approach successfully track all sequences, with our method consistently achieving superior results. Our lower errors demonstrate the robustness of our approach.

TABLE V: Evaluation of reconstruction completeness and accuracy on FastCaMo-Real benchmark.

Method	Apart._1	Apart._2	Gym	Lab	Avg.
ROSEFusion [13]	84.4% 4.4cm	**84.4%** 3.1cm	43.0% 4.9cm	**84.3%** **2.9cm**	74.0% 3.8cm
MIPS-Fusion [25]	76.2% 4.2cm	70.6% 4.2cm	- -	71.8% 3.4cm	- -
HERO-SLAM [9]	73.2% 5.1cm	71.1% 4.7cm	- -	50.5% 4.6cm	- -
Ours	**89.7%** **3.3cm**	84.0% **3.0cm**	**56.0%** **4.5cm**	84.2% 3.0cm	**78.5%** **3.5cm**

TABLE VI: Evaluation of reconstruction completeness and accuracy on FastCaMo-Real benchmark. To mimic unstable motion, we uniformly drop 20% of frames for each sequence.

Method	Lou._1	Lou._2	Meet.	Office	Avg.
ROSEFusion [13]	76.8% 2.2cm	80.2% 4.1cm	**76.8%** 3.8cm	44.7% 3.3cm	69.6% 3.4cm
Ours	**77.8%** **2.1cm**	**94.3%** **2.9cm**	75.7% **3.7cm**	**54.9%** **2.1cm**	**75.7%** **2.7cm**

TABLE VII: Evaluation of reconstruction completeness and accuracy on FastCaMo-Real benchmark. We drop 50%-80% of frames for each sequence, making it more challenging.

Method	Stair.	Studio	Work._1	Work._2	Avg.
ROSEFusion [13]	- -	- -	77.2% 3.2cm	52.2% 4.2cm	- -
Ours	**66.8%** **2.8cm**	**67.8%** **2.6cm**	**82.8%** **3.1cm**	**64.2%** **3.5cm**	**70.4%** **3.0cm**

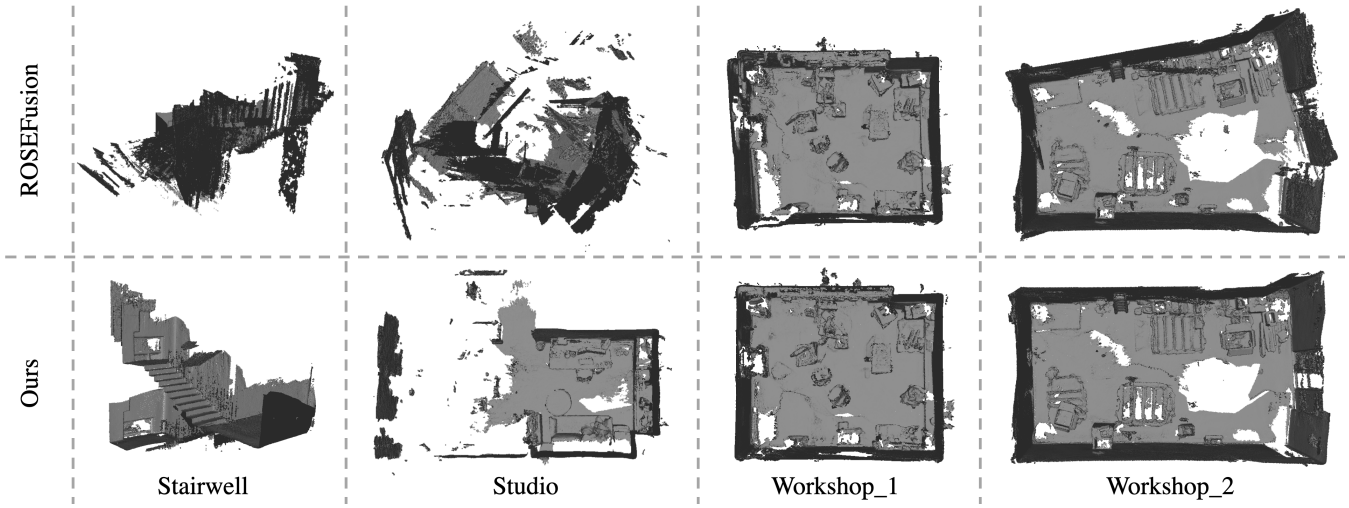

Fig. 5: Visual comparison between ROSEFusion (the most robust competitor) and our system. We present dense reconstruction results from the four most challenging sequences from FastCaMo-Real. For each sequence, we drop 50%-80% of frames to mimic unstable motion. Our system performs only single-frame camera tracking without bundle adjustment or loop closure, yet the reconstructed layout demonstrates both robustness (no wrong registration) and accuracy (minimal drift).

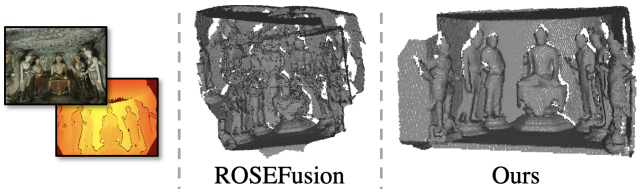

Fig. 6: Despite being trained only on daily indoor scenes, our system generalizes well to novel environments such as cave sculptures. While ROSEFusion can handle moderately fast camera motion, sudden shaking can cause wrong registration and disrupt the reconstruction.

FastCaMo-Synth [13]. This benchmark contains 10 room-scale synthetic scenes. The RGB-D sequences are rendered from manually crafted fast camera motions. The average camera speed is 54.43 degrees/s and 1.68 m/s. The benchmark provides two sets of sequences for each scene: one with perfect color and depth images, and another with simulated motion blur and depth noise. The camera tracking results are reported in Table I and II, respectively.

Despite perfect RGB-D frames in the raw sequences, several systems struggle with tracking robustness. ElasticFusion [5] and NICE-SLAM [7] often produce incorrect registrations and messy reconstructions, while BundleFusion [20] usually drops frames and loses tracking capability. HERO-SLAM [9] successfully tracks all sequences using learning-based keypoint matching and bundle adjustment, but exhibits noticeable errors and operates $<$10 FPS. MIPS-Fusion [25] tracks all sequences with randomized optimization, but its neural representation and hybrid optimization strategy compromise robustness compared to ROSEFusion [13]'s pure randomized optimization approach. Our system outperforms all competing methods with superior robustness and accuracy, delivering the lowest camera tracking error.

When using noisy RGB-D frames, tracking accuracy degrades for all methods. Blur in color images specifically affects photometric losses in neural representations, while noisy depth data introduces errors in geometric-based optimization. These imperfect inputs also reduce the performance of our pose regression network and randomized optimization algorithm. Despite these challenges, our method still achieves the best overall accuracy compared to competing approaches. This further validates our robustness.

FastCaMo-Real [13]. This benchmark contains 12 real-world scanning sequences featuring large-scale scenes. The benchmark doesn't provide ground-truth camera poses. Instead, it offers high-quality laser-scanned meshes as reference models. We align the scene reconstructions to these reference models using CloudCompare and then evaluate the completeness and accuracy of the 3D models. We calculate completeness by iterating through each point in the ground-truth model and determining the percentage of points successfully reconstructed (10cm threshold). Accuracy is measured as the average error of the reconstructed points.

By combining pose regression and randomized optimization, our system provides both robustness and accuracy in camera pose estimation, ultimately yielding more favorable reconstruction results compared to competitors. The numerical results for the first 4 scenes are reported in Table V. The methods not included in the table failed in most of the scenes. We can observe that ROSEFusion [13] remains the most robust competitor in the literature. Overall, our reconstruction results achieve the best completeness and accuracy.

To further validate our robustness, we make the remaining scenes more challenging by dropping some frames in the sequences to simulate more unstable RGB-D streams. As shown in Table VI, our advantage over ROSEFusion [13] becomes evident with 20% of frames dropped. When dropping 50%-80% of frames, ROSEFusion [13] produces noticeable

errors and fails in two scenes, while our method remains reliable. Figure 5 provides a visual comparison, and the numerical results are reported in Table VII.

Real-world applications. We captured several RGB-D videos in different environments and at various frame rates. We first visualize the reconstruction results from two representative sequences in Figure 1. The first sequence is at 5 FPS, featuring large viewpoint changes but less motion blur, while the second is at 30 FPS with fast in-place rotation. We compared our system with the strongest competitor, ROSEFusion [13]. While ROSEFusion [13] produces noticeable incorrect registrations, our method demonstrates robust performance. In Figure 6, we visualize the reconstruction results from a sequence of sculptures scanned in a cave with sudden camera shaking. Our clean reconstruction results demonstrate the robustness and generalizability of our approach, despite the network being trained only on daily indoor scenes.

C. Analyses

In this section, we showcase why both pose regression and randomized optimization are necessary components of our system. We report runtime statistics and also discuss limitations along with directions for future work.

Ablation study. The key design of our system combines pose regression (PR) and randomized optimization (RO). To verify their effectiveness, we run experiments on FastCaMo-Synth (raw) and analyze the statistics of relative pose estimates. As shown in Table VIII, PR alone lacks accuracy (higher minimum error), while RO alone lacks robustness (higher maximum error). Figure 7 illustrates this trade-off: PR exhibits drift, and RO produces incorrect registration. Combining them achieves both robustness and accuracy.

TABLE VIII: Relative pose errors with different methods.

Method	Median	Minimum	Maximum
PR	0.26cm	0.16mm	2.60cm
RO	0.27cm	**0.06mm**	5.55cm
Full (PR+RO)	**0.25cm**	**0.06mm**	**2.12cm**

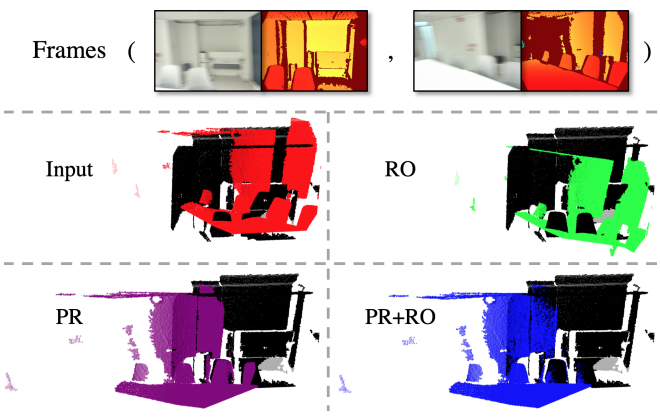

Fig. 7: Visual comparison with different methods.

Runtime statistics. Our experiments are conducted on a server with an Intel® Xeon® Silver 4314 CPU (2.40GHz ×16) and an NVIDIA RTX 4090 GPU. In Table IX, we report our running statistics for reconstructing the scene Frl_apartment_2 from FastCaMo-Synth (noise) benchmark. We use a TSDF resolution of 2cm and a grid size of $500 \times 300 \times 750$ (sufficient to cover 150 m^2). Our pose regression network takes the main GPU memory consumption, while TSDF memory usage varies with resolution and grid size. Across all experiments, the total GPU memory consumption of our system remains under 10GB. Our network is purely feed-forward, achieving inference times under 20ms. Our randomized optimization runs in parallel using CUDA, requiring only a few milliseconds. Overall, our system can deliver real-time performance over 30 FPS.

TABLE IX: Resource consumption and running time.

Peak memory (GB)		Running time (ms)		Frames per second
PR	TSDF+RO	PR	RO	
6.76	1.19	<20	<10	>30

Limitations. A noticeable limitation of our system is its reliance on single-frame tracking without integrating bundle adjustment or loop closure. This lack of global optimization can lead to accumulated drift in very large scenes. In addition, our method typically fails when input frames lack features, such as completely blurred images or frames with no overlap due to extremely fast motion. These conditions make the registration problem ill-posed. Integrating IMU data could potentially solve this challenge. Addressing these limitations will be our focus in future work.

V. CONCLUSION

In this paper, we present a robust and accurate dense scene reconstruction system. This is achieved through a simple combination of a camera pose regression network and a randomized optimization algorithm. Despite its simplicity, the system provides high-quality reconstruction results in real-time, regardless of camera motion stability. This makes it suitable for robot applications such as exploration and rescue scenarios that potentially involve unstable motions.

Acknowledgment. This work is supported by the Early Career Scheme of the Research Grants Council (grant # 27207224), the Hong Kong STEM Professorship program, and the JC STEM Lab of Autonomous Intelligent Systems funded by The Hong Kong Jockey Club Charities Trust. We thank the editors and reviewers for their valuable suggestions and Yuzheng Liu, Jiazhao Zhang, and Shuzhe Wang for their help with experiments and proofreading.

REFERENCES

[1] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon, "Kinectfusion: Real-time dense surface mapping and tracking," in *2011 10th IEEE international symposium on mixed and augmented reality*. Ieee, 2011, pp. 127–136.

[2] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, *et al.*, "Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera," in *Proceedings of the 24th annual ACM symposium on User interface software and technology*, 2011, pp. 559–568.

[3] M. Nießner, M. Zollhöfer, S. Izadi, and M. Stamminger, "Real-time 3d reconstruction at scale using voxel hashing," *ACM Transactions on Graphics (ToG)*, vol. 32, no. 6, pp. 1–11, 2013.

[4] J. Chen, D. Bautembach, and S. Izadi, "Scalable real-time volumetric surface reconstruction." *ACM Trans. Graph.*, vol. 32, no. 4, pp. 113–1, 2013.

[5] T. Whelan, S. Leutenegger, R. F. Salas-Moreno, B. Glocker, and A. J. Davison, "Elasticfusion: Dense slam without a pose graph." in *Robotics: science and systems*, vol. 11, no. 3. Rome, 2015.

[6] E. Sandström, Y. Li, L. Van Gool, and M. R. Oswald, "Point-slam: Dense neural point cloud-based slam," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 18 433–18 444.

[7] Z. Zhu, S. Peng, V. Larsson, W. Xu, H. Bao, Z. Cui, M. R. Oswald, and M. Pollefeys, "Nice-slam: Neural implicit scalable encoding for slam," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 12 786–12 796.

[8] Z. Peng, T. Shao, Y. Liu, J. Zhou, Y. Yang, J. Wang, and K. Zhou, "Rtg-slam: Real-time 3d reconstruction at scale using gaussian splatting," in *ACM SIGGRAPH 2024 Conference Papers*, 2024, pp. 1–11.

[9] Z. Xin, Y. Yue, L. Zhang, and C. Wu, "Hero-slam: Hybrid enhanced robust optimization of neural slam," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 8610–8616.

[10] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[11] S. Rusinkiewicz and M. Levoy, "Efficient variants of the icp algorithm," in *Proceedings third international conference on 3-D digital imaging and modeling*. IEEE, 2001, pp. 145–152.

[12] E. Sucar, S. Liu, J. Ortiz, and A. J. Davison, "imap: Implicit mapping and positioning in real-time," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 6229–6238.

[13] J. Zhang, C. Zhu, L. Zheng, and K. Xu, "Rosefusion: random optimization for online dense reconstruction under fast camera motion," *ACM Transactions on Graphics (TOG)*, vol. 40, no. 4, pp. 1–17, 2021.

[14] S. Wang, V. Leroy, Y. Cabon, B. Chidlovskii, and J. Revaud, "Dust3r: Geometric 3d vision made easy," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 20 697–20 709.

[15] V. Leroy, Y. Cabon, and J. Revaud, "Grounding image matching in 3d with mast3r," in *European Conference on Computer Vision*. Springer, 2024, pp. 71–91.

[16] S. Dong, S. Wang, S. Liu, L. Cai, Q. Fan, J. Kannala, and Y. Yang, "Reloc3r: Large-scale training of relative camera pose regression for generalizable, fast, and accurate visual localization," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 16 739–16 752.

[17] J. Wang, M. Chen, N. Karaev, A. Vedaldi, C. Rupprecht, and D. Novotny, "Vggt: Visual geometry grounded transformer," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 5294–5306.

[18] M. Halber and T. Funkhouser, "Fine-to-coarse global registration of rgb-d scans," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1755–1764.

[19] B. Glocker, S. Izadi, J. Shotton, and A. Criminisi, "Real-time rgb-d camera relocalization," in *2013 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, 2013, pp. 173–179.

[20] A. Dai, M. Nießner, M. Zollhöfer, S. Izadi, and C. Theobalt, "Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration," *ACM Transactions on Graphics (ToG)*, vol. 36, no. 4, p. 1, 2017.

[21] O. Kähler, V. A. Prisacariu, C. Y. Ren, X. Sun, P. Torr, and D. Murray, "Very high frame rate volumetric integration of depth images on mobile devices," *IEEE transactions on visualization and computer graphics*, vol. 21, no. 11, pp. 1241–1250, 2015.

[22] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE transactions on robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[23] R. A. Newcombe, D. Fox, and S. M. Seitz, "Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 343–352.

[24] J. McCormac, A. Handa, A. Davison, and S. Leutenegger, "Semanticfusion: Dense 3d semantic mapping with convolutional neural networks," in *2017 IEEE International Conference on Robotics and automation (ICRA)*. IEEE, 2017, pp. 4628–4635.

[25] Y. Tang, J. Zhang, Z. Yu, H. Wang, and K. Xu, "Mips-fusion: Multi-implicit-submaps for scalable and robust online neural rgb-d reconstruction," *ACM Transactions on Graphics (TOG)*, vol. 42, no. 6, pp. 1–16, 2023.

[26] Y. Lan, C. Zhu, S. Zhi, J. Zhang, Z. Wang, R. Yi, Y. Wang, and K. Xu, "Remixfusion: Residual-based mixed representation for large-scale online rgb-d reconstruction," *arXiv preprint arXiv:2507.17594*, 2025.

[27] H. Wang and L. Agapito, "3d reconstruction with spatial memory," in *2025 International Conference on 3D Vision (3DV)*. IEEE, 2025, pp. 78–89.

[28] Z. Tang, Y. Fan, D. Wang, H. Xu, R. Ranjan, A. Schwing, and Z. Yan, "Mv-dust3r+: Single-stage scene reconstruction from sparse views in 2 seconds," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 5283–5293.

[29] J. Yang, A. Sax, K. J. Liang, M. Henaff, H. Tang, A. Cao, J. Chai, F. Meier, and M. Feiszli, "Fast3r: Towards 3d reconstruction of 1000+ images in one forward pass," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 21 924–21 935.

[30] S. Elflein, Q. Zhou, and L. Leal-Taixé, "Light3r-sfm: Towards feed-forward structure-from-motion," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 16 774–16 784.

[31] Q. Wang, Y. Zhang, A. Holynski, A. A. Efros, and A. Kanazawa, "Continuous 3d perception model with persistent state," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 10 510–10 522.

[32] Y. Liu, S. Dong, S. Wang, Y. Yin, Y. Yang, Q. Fan, and B. Chen, "Slam3r: Real-time dense scene reconstruction from monocular rgb videos," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 16 651–16 662.

[33] R. Murai, E. Dexheimer, and A. J. Davison, "Mast3r-slam: Real-time dense slam with 3d reconstruction priors," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 16 695–16 705.

[34] N. Keetha, N. Müller, J. Schönberger, L. Porzi, Y. Zhang, T. Fischer, A. Knapitsch, D. Zauss, E. Weber, N. Antunes, *et al.*, "Mapanything: Universal feed-forward metric 3d reconstruction," *arXiv preprint arXiv:2509.13414*, 2025.

[35] C. Cheng, S. Yu, Z. Wang, Y. Zhou, and H. Wang, "Outdoor monocular slam with global scale-consistent 3d gaussian pointmaps," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025, pp. 26 035–26 044.

[36] B. Curless and M. Levoy, "A volumetric method for building complex models from range images," in *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, 1996, pp. 303–312.

[37] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[38] C. Yeshwanth, Y.-C. Liu, M. Nießner, and A. Dai, "Scannet++: A high-fidelity dataset of 3d indoor scenes," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 12–22.

[39] A. Avetisyan, C. Xie, H. Howard-Jenkins, T.-Y. Yang, S. Aroudj, S. Patra, F. Zhang, D. Frost, L. Holland, C. Orme, *et al.*, "Scenescript: Reconstructing scenes with an autoregressive structured language model," in *European Conference on Computer Vision*. Springer, 2024, pp. 247–263.

[40] J. Shotton, B. Glocker, C. Zach, S. Izadi, A. Criminisi, and A. Fitzgibbon, "Scene coordinate regression forests for camera relocalization in rgb-d images," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013, pp. 2930–2937.

[41] J. Straub, T. Whelan, L. Ma, Y. Chen, E. Wijmans, S. Green, J. J. Engel, R. Mur-Artal, C. Ren, S. Verma, *et al.*, "The replica dataset: A digital replica of indoor spaces," *arXiv preprint arXiv:1906.05797*, 2019.

[42] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of rgb-d slam systems," in *2012 IEEE/RSJ international conference on intelligent robots and systems*. IEEE, 2012, pp. 573–580.

[43] T. Schops, T. Sattler, and M. Pollefeys, "Bad slam: Bundle adjusted direct rgb-d slam," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 134–144.