TOWARDS ROBUST INDOOR VISUAL SLAM AND DENSE RECONSTRUCTION FOR MOBILE ROBOTS

W. Zhang[1,2,*], S. Wang[2,3], N. Haala[1]

[1] Institute for Photogrammetry, University of Stuttgart, Germany - (wei.zhang, norbert.haala)@ifp.uni-stuttgart.de
[2] Audiovisual Lab, Huawei Munich Research Center, Germany - (wei.zhang3, sen.wang)@huawei.com
[3] Technical University of Munich, Germany - sen.wang@tum.de

Commission I, ICWG I/IV

KEY WORDS: Visual SLAM, Dense Reconstruction, Mutil-Sensor Fusion, TSDF Map, Mobile Robot.

ABSTRACT:

Mobile robots are being increasingly employed in various indoor scenarios. The fundamental prerequisite is that the robot can reconstruct an accurate and complete map of the observed environment and estimate the track of its movements in this map. Current visual SLAM methods can perform this task reasonably well, but mostly in small spaces, such as a single room, and often tested in well-textured environments. In real-world applications of large indoor scenes, they lack robustness and fail to build a globally consistent map. To this end, we propose a novel system that can robustly solve the problem encountered by existing visual SLAM methods, such as weak texture and long-term drift. By combining information from a wheel odometer, the robot poses can be predicted smoothly in the absence of texture. The geometric cues are leveraged by aligning Truncated Signed Distance Function (TSDF) based submaps to minimize the long-term drift. To reconstruct a more complete and accurate dense map, we refine the sensor depth maps by taking advantage of color information and the optimization result of global bundle adjustment. As a result, the system can provide precise trajectory estimation and a globally consistent map for the downstream tasks. We validate the accuracy and robustness of the proposed method on both public and self-collected datasets and show the complementary nature of each module. Evaluation results based on high precision ground-truth show an improvement in the mean Absolute Trajectory Error (ATE) from 21 cm to 2 cm for the trajectory estimation, and the reconstructed map has a mean accuracy of 8 cm.

1. INTRODUCTION

Simultaneous localization and mapping (SLAM) is an essential technique for robots equipped with environmental sensors to explore and interact with the physical world and provide fast and reliable 3D information of indoor environments (Maboudi et al., 2018). To this end, Blaser et al. (2018) presented the high performance of a Lidar-based SLAM system. We focus on robotic applications that use depth cameras, as they, by contrast, are more affordable and have richer 3D information than Lidar. However, the existing visual SLAM methods lack sufficient robustness and are prone to failure under challenging scenarios. For example, global inconsistency can arise due to the absence of large-baseline loop closures in a multi-room scene. Further example can be a corridor with forward and reverse revisits as present in Shi et al. (2020), where the reverse loop can not be closed due to a large view deviation. In this work, we explore the advantages of the different techniques, which are mutually complementary and reinforcing. Specifically, the wheel odometer supports camera tracking when optical flow matching fails due to weak texture. Fig. 1 a) shows an example where the visual tracking failed due to a textureless wall that took up the entire camera view. After fusing the wheel odometer measurements, camera poses are predicted smoothly, as can be seen in Fig. 1 b). Furthermore, the geometric-based submap registration can fix the long-term drift regardless of significant viewpoint changes, as shown in Fig. 1 c) and d). As a result, accurate dense reconstruction of multi-room scene can be achieved by our proposed system as presented in Fig. 2

As an essential part of the proposed system, we choose the dense optical flow-based method (Teed and Deng, 2021) as a strong baseline. With the increased computing power and deep learning research advances, this optical flow-based approach emerged as a new promising visual SLAM paradigm. Before that, the community generally divided SLAM methods into keypoint-based methods and photometric consistency-based direct methods. Based on the principle and characteristics of each method, both have their merits. On the one hand, the keypoint-based method is easier to optimize for the global

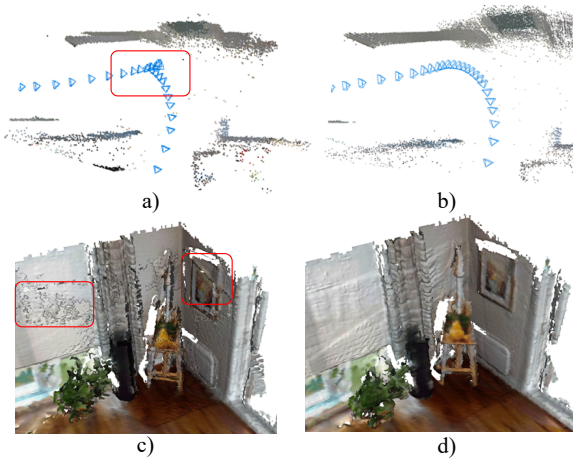

Figure 1. a) Failed camera tracking in front of a white wall; b) Smooth tracking after fusing with wheel odometer measurements; c) Reconstructed map contaminated by the misalignment due to missed large-baseline loop closure; d) Improved dense map with submap registration.

* Corresponding author

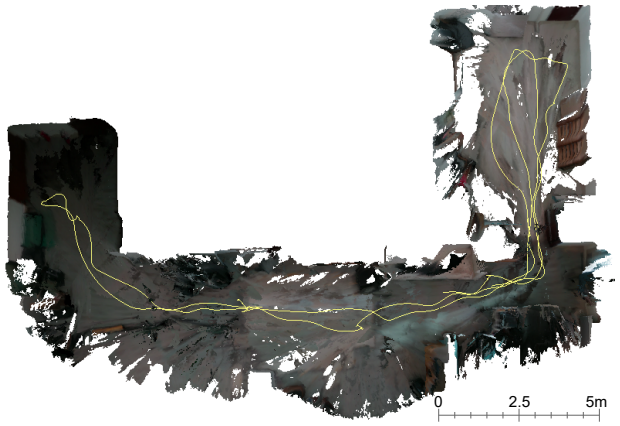

Figure 2. Multi-room reconstruction and the trajectory estimated by the proposed system.

optimum (Campos et al., 2021). However, it is limited to using only a fraction of image information, leaving the rest of the potentially helpful information aside. On the other hand, the direct methods use semi-dense pixel information instead of only sparse keypoints (Engel et al., 2017), but minimizing the photometric consistency loss is a more complex problem being more likely to fall into a local minimum during optimization. In contrast, the dense optical flow-based method combines the advantages of both keypoint-based and direct photometric-based methods by applying reprojection errors for dense pixel information as a stable optimization objective.

The SLAM methods applied for mobile robots frequently combine the measurements from a wheel encoder, a standard sensor to support mobile robot navigation. Depending on the ground material, the wheel odometer can encounter varying degrees of sliding, causing different level measurement errors. Integrating the pose over a long period, the error can accumulate to a considerable amount. By contrast, the pose integration over a short time, more specifically, over the period of two adjacent keyframes, is fairly reliable. Therefore, to enhance the robustness of the overall system, the relative 2D pose transformations of the wheel odometer are added as additional constraints between consecutive keyframes. These constraints increase the smoothness of trajectory estimation and are especially useful when the texture is missing from the camera view. Thus from the practical point of view, it is a valuable complement to the visual system.

While moving through the indoor environment, mobile robots need to rely on a dense 3D map for navigation and obstacle avoidance. We use a depth camera to enhance the acquisition of 3D information since the off-the-shelf depth camera systems are becoming increasingly available. Furthermore, depth cameras are beneficial in textureless indoor scenarios as providing geometric information for alignment. Simply accumulating depth point clouds of all frames will lead to extensive information redundancy, as neighbor frames have largely overlapped observations. For this purpose, the TSDF is selected as the representation of the dense 3D map, which is a voxel-based representation (Curless and Levoy, 1996). 3D points falling into the same voxel are fused. While the redundancy is reduced, the impact of observation noise is minimized by weighted averaging. Recent works leverage the voxel hashing method (Nießner et al., 2013) for constructing a TSDF map. A balance of accuracy and efficiency can be achieved by only constructing the observed regions and selecting an appropriate voxel size. Inspired by the submap idea of Reijgwart et al. (2020), we construct separated TSDF submaps within a specific time window, which can be geometrically aligned to each other to provide additional constraints for global bundle adjustment (BA). As a result, the accumulated pose drift can be mitigated with the closed long-term loop. Compared to dense optical flow constraints, which can miss potential loop closures due to significant viewing angle and baseline deviations, the submap registration is not limited by viewing angle changes and can robustly align the map geometrically.

Another key of the proposed system is the depth map refinement strategy to enhance the accuracy and completeness of the dense reconstruction. The depth observations have a limited range due to the restricted power consumption and device size limitation. Moreover, the observation error increases quadratically with respect to the depth range according to Szeliski (2010). When it comes to distant observations, depth measurements suffer from significant observation noise, which can lead to numerous ghosting and artifacts in the reconstructed map, affecting the map accuracy and the robot navigation. To improve the depth map quality, we use the off-the-shelf results of global BA from the SLAM backend. The optimized depth maps of the keyframes benefit from the dense matching of optical flow between neighbor frames and have higher accuracy due to the joint optimization of increased information. Based on the degree of over-determination, a confidence value of each optimized depth pixel can be derived, based on which the optimized depth is fused adaptively with raw sensor depth. Furthermore, we apply the bilateral solver (Barron and Poole, 2016) to enhance the smoothness of the sensor depth map, which takes advantage of the appearance information in the color image to keep the sharpness of object edges while smoothing the measurements on the object surface. Finally, the resulting refined depth maps are fused into a combined 3D map, which proves to be more accurate and has higher completeness by comparing against a reference model scanned by a high-end laser scanner. The main contributions of this work are summarized as follows:

- A novel robust indoor visual SLAM framework, overcoming the challenges of short-period texture missing and long-term large-baseline loop closuring by exploring the mutually complementary nature of different constraints.
- A depth map refinement strategy based on the analysis of sensor raw depth errors and combining color cues with bilateral solver and optimization results of global bundle adjustment.
- Experiments on multiple datasets and over hundreds of sequences to validate the effectiveness of the proposed system. The final reconstruction is evaluated against the high-end laser scanner model.

2. RELATED WORK

Indoor Visual-SLAM problem has been extensively studied for decades for both the works based on monocular and RGBD cameras. Despite the remarkable advances, there is still considerable room for improving the robustness in challenging scenarios (Bujanca et al., 2021). We notice two major trends emerging in recent years. One is to fuse the multi-sensor information, and the other is to combine the deep learning techniques from other related fields. Concerning multi-sensor fusion, one typical complementary sensor to the camera is the Inertial Measurement Unit (IMU), which has the advantage of

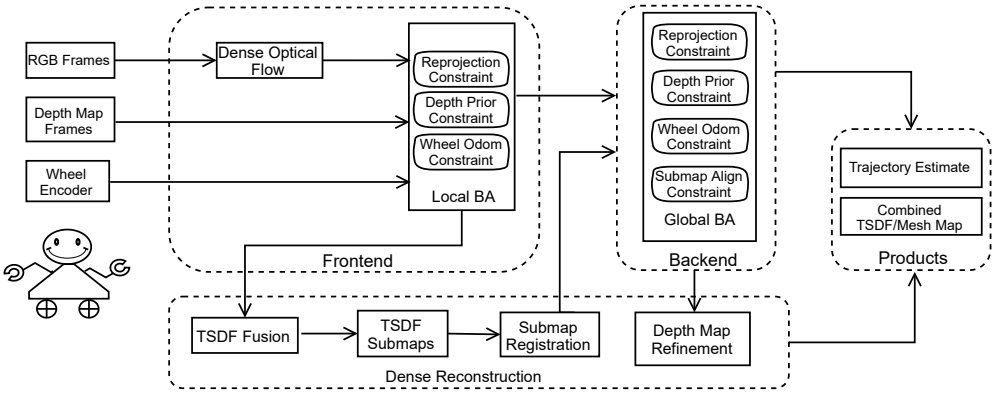

Figure 3. Overall pipeline mainly consists of three modules. The SLAM frontend and backend modules are detailed in Sec. 4, while the details of the dense reconstruction module are illustrated in Sec. 5.

adding scale information and providing reliable gravity direction but can suffer from unobservability problems under particular motion (Qin et al., 2018). Another is the combination with wheel odometer information to provide 2D motion information (Wu et al., 2017), as this is the typical motion of a ground robot. For the methods applying deep learning, some seek for applying deep neural networks to estimate camera pose and depth map directly (Zhou et al., 2017). Nevertheless, it does not seem to be an area where deep learning specializes too well through its end-to-end learning strategy. By contrast, it is believed that a better application area of deep learning for visual-SLAM is to solve feature extraction and matching problems such as in Sarlin et al. (2020). Our work benefits from recent deep learning advances in this trend. More specifically, this work applies the pre-trained network by Teed and Deng (2021), which can predict robust optical flows between overlapped frames.

Dense Reconstruction from the input data of a moving camera requires, on the one hand, the accurate camera pose estimation by a SLAM system, and more importantly, the sequence of depth map estimates to be fused into a global model. With a 3D depth camera, the depth map of each view can be obtained, but the observations of depth cameras suffer from varying degree errors. Depth cameras based on Time-of-Flight or structured-light principles are prone to blurring due to camera motion and multipath interference (Giancola et al., 2018). We choose the active stereo camera as an alternative between accuracy and reliability. Compared to passive stereo vision, the active stereo camera equipped with a projector adds additional random patterns to the environment to still work in dark and low-texture environments. However, like stereo matching, the depth error increases quadratically with respect to the depth of object (Szeliski, 2010), and also other effects such as occlusion, holes, edge flattening. Zhang et al. (2018) analyzed these effects and leveraged deep neural networks to improve the accuracy of depth maps. Barron and Poole (2016) enhanced the depth smoothness based on the color image. The proposed work differs from existing works by combing the smoothing techniques with the off-the-shelf optimized depth results of dense SLAM system to improve both the smoothness and completeness of the depth map estimation.

3. SYSTEM OVERVIEW

Enabling mobile robots to interact seamlessly with the physical world and help people in daily lives, the goal is to reconstruct an accurate and complete map of the observed environment and estimate the precise robot trajectory in the unknown environment. As illustrated in Fig. 3, the color image and depth map along with wheel odometer measurements are fed into the system. On the SLAM frontend, the incoming image is matched densely with nearby frames within a local window based on an optical flow network (Teed and Deng, 2021). Three types of constraints are jointly optimized in a local window BA. As for the dense reconstruction module, depth maps are fused incrementally into timely separated submaps with the tracked camera poses from the SLAM frontend. In addition to the same constraints as the frontend, we compute the submap registration constraint to enhance the consistency in the areas observed multiple times from far viewpoints. The camera poses and depth maps at a lower resolution are jointly optimized by global BA to reach a maximum global consistency. Subsequently, before constructing the final 3D map, the sensor raw depth maps are refined based on the optimization results of global BA, and the bilateral solver is applied to improve the spatial smoothness.

4. ROBUST INDOOR VISUAL SLAM

This section presents the methodology of the proposed SLAM system in detail. First, the optical flow calculation (Sec. 4.1) is an important step performed at the arrival of each frame and recalculated iteratively. For the wheel odometer observation, the equation for converting the raw wheel angular velocity to robot motion will be formulated, then added as wheel odometer constraints to the optimization objective (Sec. 4.2). On the backend of global BA, we append the submap alignment term (Sec. 4.3) to facilitate further consistency between long-term views, which have a common observation area. Finally, the strategy of keyframe selection is discussed in Sec.4.4. The above three constraints of all keyframes are jointly optimized with local and global BA on the SLAM frontend and backend respectively.

4.1 Dense Optical Flow

The dense optical flow network predicts the pixel-wise dense correspondences between adjacent frames. Following recent advances in this field, we apply the optical flow network of Teed and Deng (2021) due to its excellent generalization ability. The network model pre-trained on TartanAir datasets (Wang et al., 2020) can still have great performance without refining on the test dataset. Based on Teed and Deng (2021), we elaborate on the principle of the optical flow prediction step. First, a prior optical flow can be derived from the current pose and depth estimates. Therefore, more specifically, the task of the network is

to predict an update of the optical flow based on current optical flow estimate. The updated optical flow is then considered as the reference to derive the reprojection error.

For each newly arrived frame i, the pose and depth map are initialized with the pose predicted by the wheel odometer and the raw depth measurement from the depth camera, respectively. Then this latest frame is paired with the nearby N frames which are most similar within the local window. The 2D image coordinates as grid array $\mathbf{p}_i$ are transformed to 3D coordinates $\hat{\mathbf{P}}_i$ with the camera intrinsic parameter $\mathbf{K}$ and current depth map $\hat{\mathbf{d}}_i$ in the local coordinate system of the current frame as follows:

$$\hat{\mathbf{P}}_i = \hat{\mathbf{d}}_i \mathbf{K}^{-1} \mathbf{p}_i \tag{1}$$

Based on the current pose estimate, the relative transformation $\hat{\mathbf{T}}_{ij}$ between current frame *i* and the adjacent frame *j* to be matched can be derived as $\hat{\mathbf{T}}_{ij} = \hat{\mathbf{T}}_j^{-1} \hat{\mathbf{T}}_i$. With that, the 3D point arrays are transformed into the coordinate system of adjacent frame:

$$\hat{\mathbf{P}}_{ij} = \hat{\mathbf{T}}_{ij} \hat{\mathbf{P}}_i \tag{2}$$

Meanwhile, the 3D coordinates $\hat{\mathbf{P}}_{ij}$ are now in the local frame of the adjacent frame *j*. They can be projected again onto the 2D image plane of the adjacent frame. Note that this time the projection is with respect to the adjacent frame to finally obtain the positions of the correspondences based on the current estimates.

$$\hat{\mathbf{p}}_{ij} = \mathbf{K} \hat{\mathbf{P}}_{ij} \tag{3}$$

To chain the three formulas above together, the original 2D image coordinates in frame i can be transformed to the corresponding 2D coordinates on the adjacent frame j as follows:

$$\hat{\mathbf{p}}_{ij} = \mathbf{K} \hat{\mathbf{T}}_{ij} \hat{\mathbf{d}}_i \mathbf{K}^{-1} \mathbf{p}_i \tag{4}$$

The optical flow network takes the current flow motion as the initial value, based on which the local perceptive field are matched to predict the update motion $\boldsymbol{\delta}_{ij}$. As a result, revised optical motion $\breve{\mathbf{p}}_{ij}$ is obtained as follows:

$$\breve{\mathbf{p}}_{ij} = \hat{\mathbf{p}}_{ij} + \boldsymbol{\delta}_{ij} \tag{5}$$

In order to tackle the false matching problem caused by occlusion for example, a confidence value w_i is also predicted by the network for each pixel so that the pixels without a counterpart in another image have a lower weight. The revised flow positions updated by the network play the role as the reference. The reprojection error $\mathbf{r_{ij}}$ is now computed by taking the residual between revised flow $\breve{\mathbf{p}}_{ij}$ and the current flow estimate $\hat{\mathbf{p}}_{ij}$.

$$\mathbf{r}_{ij} = \|\breve{\mathbf{p}}_{ij} - \hat{\mathbf{p}}_{ij}\|^2_{w_i} = \|\breve{\mathbf{p}}_{ij} - \mathbf{K} \hat{\mathbf{T}}_{ij} \hat{\mathbf{d}}_i \mathbf{K}^{-1} \mathbf{p}_i\|^2_{w_i} \tag{6}$$

In addition to the currently tracked frame, the same operation is performed on all other frames located in the frontend window so that all reprojection errors are optimized in batch. For the backend, based on the covisibility graph, the keyframe pairs that share common observations are selected, and the reprojection errors are applied and computed as before. In total, over the all frame pairs denoted as the set $\mathcal{R}$, the total reprojection error can be summarized as follows:

$$E_{reproj} = \sum_{(i,j) \in \mathcal{R}} \|\breve{\mathbf{p}}_{ij} - \mathbf{K} \hat{\mathbf{T}}_{ij} \hat{\mathbf{d}}_i \mathbf{K}^{-1} \mathbf{p}_i\|^2_{w_i} \tag{7}$$

whereby the $\hat{\mathbf{T}}_{ij}$ is computed from $\hat{\mathbf{T}}_i$ and $\hat{\mathbf{T}}_j$, which along with the depth map $\hat{\mathbf{d}}_i$ are the target variables to the overall optimization function.

4.2 Wheel Odometry

The wheel encoder is installed in each robot wheel and measures the angular velocity ω_l and ω_r of the left and right wheel. The moving speed v of the robot can be derived with the known wheel radius r_l and r_r. The rotation speed ω while steering can be obtained based on the wheel speed difference between the two wheels and the wheel spacing b.

$$v = \frac{r_l \omega_l + r_r \omega_r}{2}, \quad \omega = \frac{r_r \omega_r - r_l \omega_l}{b} \tag{8}$$

Within the brief time period of two consecutive measurements and short distance of movement, the movement between two measurement timestamps can be considered as a straight line and has a constant speed. As a result, the movement can be computed directly as $v\Delta t$ and the 2D pose $[x, y, \theta]^\mathsf{T}$ is derived as follows:

$$\begin{aligned} x_{t+1} &= x_t + v\Delta t \cos(\theta_t), \\ y_{t+1} &= y_t + v\Delta t \sin(\theta_t), \\ \theta_{t+1} &= \theta_t + \omega \Delta t \end{aligned} \tag{9}$$

Note that the coordinates above are still in the base coordinate system of the robot, whereas the previous optical flow based reprojection errors are located in the camera coordinate system. In order to make the constraints consistent in a common coordinate system, the movement is transformed based on the extrinsic parameter into the camera coordinate system. Here the 2D pose is complemented with zero height, pitch, and roll into the 3D pose so that the 6 DoF transformation can be applied. To acquire the relative transform $\breve{\mathbf{T}}_{ij}$ as observation, the wheel odometer measurements lie within the period of two consecutive keyframes are integrated and subsequently transformed to the camera coordinate system. Over the keyframes in history, let $\mathcal{W}$ denote the set containing all the wheel odometer edges, the total error of wheel odometer constraints is as follows:

$$E_{wheel} = \sum_{(i,j) \in \mathcal{W}} \|\log(\breve{\mathbf{T}}_{ij}^{-1} \hat{\mathbf{T}}_{ij})\|^2, \tag{10}$$

4.3 Submap Registration

The submaps are created at regular time intervals and conditioned by a minimum traveled distance in case of slow motion. Following the practice in Reijgwart et al. (2020), the first frame is used as the anchor frame. Subsequent frames within the current time interval are transformed into the anchor coordinate system. Since the dense optical flow module gives robust prediction when sufficient texture is available, the wheel odometer can support when the texture is missing at short times. The pose sequence estimated by the frontend is considered smooth and highly accurate. Therefore, the pose of the anchor keyframe is the only optimization variable for the submap registration constraint. Note that the rest of the keyframes are still chained by the wheel odometer and reprojection constraints, so when the anchor pose changes, the rest are adapted to it.

As part of the dense reconstruction module, which will be detailed in the next section, the submap is represented as the TSDF model. The same strategy as Reijgwart et al. (2020) is applied to extract iso-surface vertices from each submap using marching cube algorithm (Lorensen and Cline, 1987). In addition, the axis-aligned bounding box (AABB) (Ericson, 2004) is computed for each submap. Based on whether the AABB intersects, the submap pairs with potential overlap are pre-selected. Each

submap pair contributes to a constraint, in which the surface vertices $\mathbf{P}_i$ of one submap are randomly sampled to be projected into the truncated signed distance field $\mathbf{\Omega}_j$ of another submap. Since the distance field reflects the distance to the nearest surface, we directly consider the looked-up truncated distance as the constraint error. When the submaps are aligned well, the surface vertices of one map should be projected onto the surface of another map and result in small or close to zero distance, and big distance otherwise. Intuitively, the distance provides the gradient naturally for the optimization to converge to a point where the submaps are registered well.

$$\mathbf{r}_{ij} = \mathbf{\Omega}_j(\hat{\mathbf{T}}_{ij}\mathbf{P}_i) \tag{11}$$

Due to the lack of direct observation, the negative distance field in the TSDF submap is more uncertain than the positive part. For instance, as illustrated in Fig. 9, the living room and bedroom share the same wall, and the area between the two wall surfaces is unobserved and has a negative distance field. As it is unknown how thick the wall is or whether it is a solid wall, the distance field at this area is unreliable. Based on this observation, the error terms induced by the vertices projected into the negative field are filtered out to avoid the mismatch at such areas. Furthermore, in order to mitigate the impact of moving objects, the Huber kernel is applied to the error function with $\rho_H(x) = x^2$ if $|x| < 1, 2|x|-1$ otherwise. We denote the set $\mathcal{S}$ containing all submap pairs and sample $\alpha\,\%$ of N_i iso-surface vertices for each submap pair i and j. The overall submap registration constraints can be formulated as:

$$E_{submap} = \sum_{(i,j)\in\mathcal{S}} \sum_{k=0}^{\alpha N_i} \rho_H\left(\mathbf{\Omega}_j(\hat{\mathbf{T}}_{ij}P_i^k)\right) \tag{12}$$

Sum of constraints: in addition to the aforementioned three constraints, the distance between the depth map variable $\hat{\mathbf{d}}_i$ and sensor depth observation $\breve{\mathbf{d}}_i$ is added as a prior term to the minimization objective. Since the sensor depth can be noisy especially for far measurements, the inverse depth is used to calculate the depth prior constraint in favor of the close depth observations. With that, the overall bundle adjustment problem can be summarized as:

$$\underset{\mathcal{T},\mathcal{D}}{\arg\min}\; E_{reproj} + E_{wheel} + E_{submap} + \sum_{i\in\mathcal{D}} \| \frac{1}{\breve{\mathbf{d}}_i} - \frac{1}{\hat{\mathbf{d}}_i} \|^2, \tag{13}$$

where $\mathcal{T}$, $\mathcal{D}$ are the collections of the camera pose and depth variables.

4.4 Keyframe Selection

On the SLAM frontend, we select and store only keyframes for efficiency. With the pose prediction of the wheel odometer, a new keyframe is added to the system when the translation is over 20 cm, or the angular change is over a threshold of $\pi/8$. In contrast, for the baseline setting without the usage of wheel odometer, following the practice of Teed and Deng (2021) the criterium is based on the average optical flow with a threshold of 3 pixels. The SLAM backend optimizes all the keyframes selected by the frontend jointly using Gauss-Newton algorithm.

5. DENSE RECONSTRUCTION

Another essential aspect of the mobile robot platform is modeling the environment accurately and densely. In this section, the details of the dense reconstruction module will be elaborated. For depth map refinement, we first analyze the noise level of the sensor depth measurements and show the importance of taking measures to improve the accuracy of input depth maps (Sec. 5.1). As indoor scenes contain frequent planes and smooth object surfaces, the fast bilateral solver (Barron and Poole, 2016) is leveraged to smooth the measurements while keeping the sharpness at object boundaries. On the other hand, as the results from the global BA, the optimized depth maps can be obtained. We fuse the optimized and smoothed depth map adaptively based on dense optical flow matching weights to derive the final depth map. Further, the refined depth maps are fused into a 3D map representation (Sec. 5.2). To mitigate the remaining noise of depth maps, the TSDF is applied to fuse depth maps.

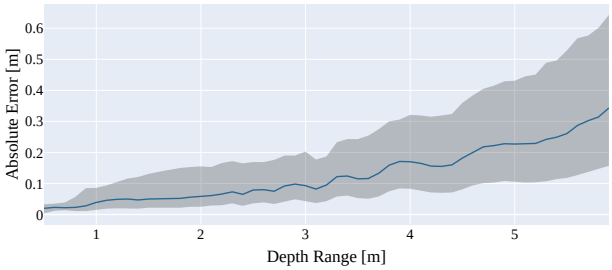

Figure 4. Depth error analysis of the depth camera Realsense D435. The solid line corresponds to the median and the lower and upper bounds are the first and third quantiles.

5.1 Depth Map Refinement

The active stereo-based depth camera, equipped with an infrared projector, is more robust than passive light-based stereo matching and can work under low light conditions. However, its accuracy is also limited by the camera baseline length, and the observation error increases quadratically with respect to the depth range. We render reference depth maps from the high-precision model by Laser scanner to analyze the depth measurement quality. Fig. 4 shows the depth error computed based on the rendered reference for the depth camera Realsense D435, which has a relatively short baseline of 5 cm. Note that the depth error can be typically further decoupled into bias and precision as presented in Nebiker et al. (2021), but due to the uncontrolled environment, the depth error here is a compound of bias and precision. It can be seen that the errors increase rapidly with depth range, for example, reaching an error level of 5 cm to 20 cm at 3 m range.

To eliminate the depth errors, the depth map is firstly smoothed using the fast bilateral solver (Barron and Poole, 2016). The color image is used as the reference to calculate the affinity between adjacent pixels. At object boundaries, due to the gradient in the color image, the sharpness in these areas is preserved, as can be seen in Fig. 5. In addition, from the global BA, the optimized depth maps are obtained. However, for efficiency reasons, the depth map is optimized at a 1/8 resolution of the original depth map. The CNN feature map is leveraged to upsample them back to the original resolution while remaining the fine details. After upsampling, another significant strength of the optimized depth map is its high completeness. It can complement the voids in the sensor depth map as shown in Fig. 5.

Although the joint optimization of dense optical flow matching over multiple frames, the optimized depth maps have high uncertainty at the areas of weak texture. Thus the final refined

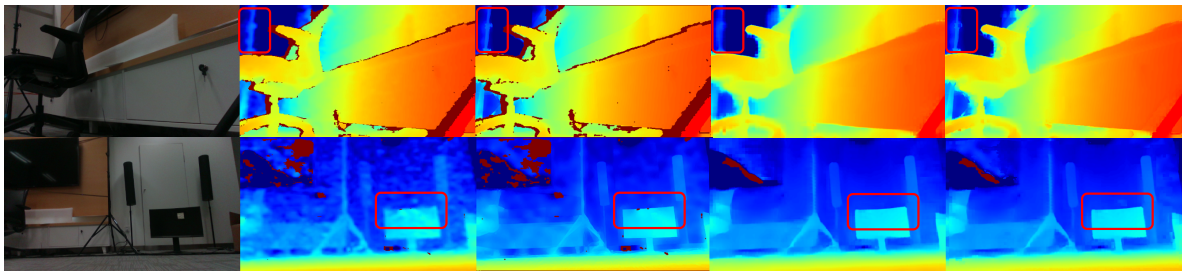

Figure 5. Left to right: color image, sensor depth map, smoothed depth map, optimized depth map, and final refined depth map. Areas with improvement are marked with red rectangles

depth map is derived via the fusion of optimized depth map $\hat{\mathbf{d}}_{opt}$ and the smoothed depth map $\hat{\mathbf{d}}_{smooth}$ based on the per-pixel confidence weight $\boldsymbol{\omega}$ of optical flow prediction and a confidence threshold δ as follows:

$$\hat{\mathbf{d}}_{refined} = \begin{cases} \hat{\mathbf{d}}_{opt}, & \text{if } \boldsymbol{\omega} > \delta \text{ or } \hat{\mathbf{d}}_{smoothed} = 0, \\ \hat{\mathbf{d}}_{smoothed}, & \text{otherwise.} \end{cases} \tag{14}$$

5.2 TSDF Fusion

The TSDF representation is applied as the 3D map representation to further reduce the impact of depth map errors. The 3D space is divided into a minimum unit of 4 cm voxels. A hash table is applied to store the distances of observed voxels. Naively treating each observation equally with a constant weight of 1 is not ideal for our case with an increasing error to observation distance. We adapt the weighting scheme with $\lambda = 1/z^2$ to have a higher reliance on closer observation, which is statistically more accurate. For each depth measurement d_i and corresponding weight λ_i, we update the signed distance D and weight Λ of all voxels within a truncation distance of 20 cm via ray-casting as follows:

$$D_{new} = \frac{\Lambda_{old} D_{old} + \lambda_i d_i}{\Lambda_{old} + \lambda_i}, \Lambda_{new} = \Lambda_{old} + \lambda_i \tag{15}$$

To further mitigate the impact of depth noise and outliers, we apply voxel filtering based on voxel weights W. Empirically a minimum weight of 0.2 is found suitable for the collected data. Once all depth maps are fused into a global TSDF map, a map represented as the mesh can be extracted from the TSDF voxels via the marching cube algorithm.

6. EXPERIMENTS

To prove the effectiveness of the proposed system, we conduct experiments on three datasets, the self-collected lab dataset, the simulated data on synthetic HM3D dataset, and the public OpenLORIS-Scene dataset. As the results of the SLAM system, the trajectory estimate is quantitatively evaluated on both the self-collected lab dataset and simulated data based on the ground-truth provided by the high-precise motion capture system (MCS) and the simulated ground-truth. For the OpenLORIS-scene dataset, we notice the original ground-truth provided by the dataset is not accurate enough, and misalignment artifacts can be noticed when reconstructing the map with the ground-truth poses. Thus we inspect the visual quality of the reconstructed map on this dataset. By evaluating various combinations of different constraints, the improvement of each extension can be verified.

6.1 Datasets

Self-collected lab dataset is collected with a turtlebot robot platform equipped with a Realsense D435 camera. The extrinsic between the camera and wheel odometer is determined with the method presented in Heng et al. (2013). The data was captured in a laboratory scenario. To provide a highly accurate reference trajectory of the robot motion, we used the OptiTrack MCS, which can capture the motion within a medium-size room. A Faro Focus3D laser scanner was used to provide a reference model of the environment. Since multiple scans can be registered together, a model of multiple rooms can be captured. From the entire 7 sequences, 6 sequences have both trajectory and map reference but are limited to a medium-sized room. The last sequence is of multi-room recording with a map reference, its reconstruction is presented in Fig. 2.

HM3D dataset (Ramakrishnan et al., 2021) is a collection of building scale real-world reconstruction and consists of different indoor spaces. The scenes were scanned using a Matterport Pro2 tripod-based depth sensor. Based on the simulation platform (Savva et al., 2019), a wheeled robot is simulated to navigate through five randomly generated coordinates. These coordinates are filtered to be located on a same height level. Otherwise, the simulated wheeled robot will get stuck in front of the stairs, for example. In total, 153 data sequences were sampled from the HM3D models. Fig. 1 c) and d) show the reconstruction from one of the sequences. The sampled data includes color images, depth maps, and ground-truth poses. To strive for more realistic data, motion blur and luminance variation noise are added to color images, noise model as described in Handa et al. (2014) applied for depth maps, and 2D pose disturbance to mimic the measurement noise of wheel odometer.

OpenLORIS-Scene dataset (Shi et al., 2020) was designed to test SLAM algorithms in challenging indoor scenes, e.g. the white wall scenario presented in Fig. 1 a) and b). The poses estimated by a 2D laser SLAM system are considered as the ground-truth. Presumably, because of a faulty calibration between camera and lidar, we noticed a misaligned map when reconstructed using the 2D lidar "ground-truth". As illustrated in Fig. 9, the reconstructed map by the proposed system presents better visual quality.

Trajectory Metric	Depth Map Metric
ATE: $\sqrt{\frac{1}{n}\sum_{i=1}^{n} \|trans(Q_i^{-1} P_i)\|^2}$	AbsDiff: $\frac{1}{n}\sum \lvert d - d^* \rvert$ AbsRel: $\frac{1}{n}\sum \lvert d - d^* \rvert / d^*$ SqRel: $\frac{1}{n}\sum \lvert d - d^* \rvert^2 / d^*$ RMSE: $\sqrt{\frac{1}{n}\sum \lvert d - d^* \rvert^2}$ Comp: % valid predictions

Table 1. Definition of evaluation metrics.

Setting	07-32-52	07-39-39	07-45-16	07-51-34	07-55-25	08-00-04	Mean
BAD-SLAM	0.229	0.919	0.231	0.338	0.416	0.964	0.516
VINS-RGBD	0.167	0.728	0.857	0.816	0.101	0.395	0.511
ORB-SLAM3(RGBD)	0.030	0.025	0.041	0.043	0.100	0.035	0.046
ORB-SLAM3(RGBD+IMU)	0.037	0.043	0.041	0.038	0.041	0.044	0.041
Wheel	0.207	0.259	0.180	0.100	0.272	0.164	0.186
Baseline	0.027	0.023	**0.017**	0.054	0.034	0.024	0.026
Baseline+wheel	0.024	**0.021**	**0.017**	0.031	**0.023**	**0.023**	0.024
Baseline+wheel+submap	**0.023**	**0.021**	**0.017**	**0.030**	**0.023**	**0.023**	**0.023**

Table 2. ATE Results in meter on the lab dataset of 6 sequences. Bold font indicates the best result for each sequence.

6.2 Metrics

The evaluation metrics are shown in Tab. 1. For the trajectory metric ATE (Sturm et al., 2012), Q_i and P_i denote the estimated and ground-truth pose at the timestamp i, $trans()$ refers to the translational part of the pose difference. For evaluating the quality of depth map, the metrics defined in Eigen et al. (2014) are followed, which are the Absolute Difference (AbsDiff), the Absolute Relative error (AbsRel), the Squared Relative error (SqRel), and the Root Mean Square Error (RMSE). Furthermore, the metric Comp is computed to indicate the completeness of depth estimates. In the formulas of depth map metrics, the estimated and ground-truth depth are denoted as d and d^*.

6.3 Quantitative Evaluation on Trajectory Accuracy

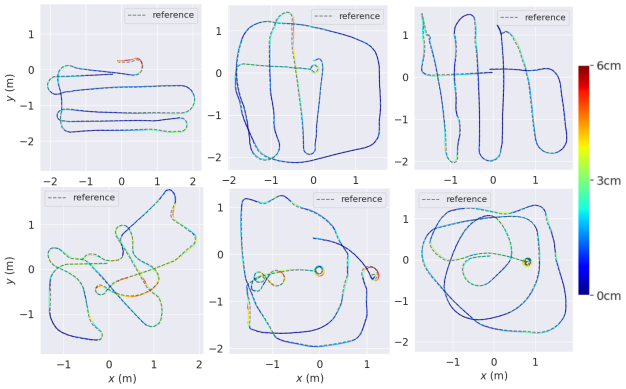

Figure 6. Estimated trajectories on lab dataset overlaid with reference. Colors correspond to the amount of translational error.

First, we validate the effectiveness of the proposed system on the lab dataset. As can be seen from Tab. 2, the ATE of the wheel odometer is the largest because the measurement errors gradually accumulate over the whole movement track. The optical-flow-based baseline achieves a remarkable improvement with the average ATE decreased from 18.6 cm to 2.6 cm. By combining the wheel odometer and submap registration, the error decreases further from 2.6 cm to 2.4 cm and 2.3 cm, respectively. To show the complexness of the robot trajectory, they are shown in Fig. 6.

Furthermore, we compare with the recent state-of-the-art visual SLAM methods as shown in Tab. 2. Since monocular methods failed to complete the tracking in our challenging dataset, only the RGBD based methods succeeded and their results are shown. BAD-SLAM (Schops et al., 2019) uses both color images and depth measurements to minimize geometric and photometric residuals via BA optimization. However, it has difficulty in our collected data due to the high noise level in the depth measurement. VINS-RGBD (Shan et al., 2019) fuses further the IMU measurements but also suffers from the high depth noise. ORB-SLAM3 (Blaser et al., 2018) has the closest performance as ours, especially the RGBD+IMU mode. We believe it is thanks to its robust keypoint matching module and soft incorporation of depth priors. Nevertheless, our proposed system, including the baseline, outperforms the ORB-SLAM3 by a large margin, showing the superiority of the dense optical-flow approach over the sparse keypoint-based approach.

Setting	Max	Mean	Median
Wheel	0.212	0.094	0.091
Baseline	10.88	0.211	0.022
Baseline+wheel	0.246	0.024	0.019
Baseline+wheel+submap	**0.183**	**0.023**	**0.018**

Table 3. Overall ATE results in meter on the 153 sequences synthesized on HM3D reconstruction models.

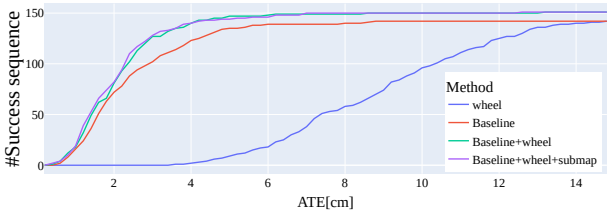

Figure 7. Cumulative error distributions of different settings on the 153 sequences synthesized on HM3D models. Upper-left means more robust, i.e. more sequences with low errors.

Further evaluations are conducted on the synthesized HM3D data. Because the robot is simulated to pass through various rooms of the scene, this data has more short-term and long-term loops, and higher diversity. As the evaluation results shown in Tab. 3, the trajectory estimated by the wheel odometer suffers from accumulated error with max and mean ATE of 21 cm and 9.4 cm. In contrast, the dense optical flow based baseline performs much better in the median ATE decreased from 9 cm to 2.2 cm and same improvement can be seen in the cumulative error plot in Fig. 7, about 120 sequences were run successfully within an ATE error of 4cm. However, the baseline has only a maximum and average ATE of 11 m and 21 cm, as it fails to track the trajectory on many sequences due to the lack of texture. Thus for more robust, a notable improvement is brought by combining the baseline with the wheel odometer. Because the wheel odometer provides reliable measurements under low-texture periods (see Fig. 1), the number of success sequences at ATE of 4cm is further increased from about 120 to 140. Besides, this setting also improved the accuracy, with mean and median ATE decreasing from 21 cm to 2.4 cm, and 2.2 cm to 1.9 cm. As for our complete system, based on the submap registration closing the long-term loop closure, the performance has a further improvement, especially concerning the robustness with the maximum ATE of all sequences of 18cm.

6.4 Investigation of Submap Registration

To further address the strength of the submap registration module, we show more qualitative results in this section. Fig. 8

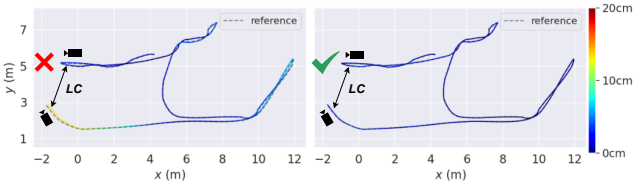

Figure 8. Estimated trajectories before and after submap registration with the amount of translational error color-coded.

shows the estimated trajectories before and after submap registration and the cross-marked place where the robot revisits after a long trip. Due to the significant view deviation at the upper-left corner, the loop closure (LC) can not be closed by the baseline method. By contrast, the submap registration is based on the object geometry and thus independent from the viewing angle. As a result, the trajectory errors at this area decrease significantly. Further, the improvement in reconstructed map for this case can be seen in Fig. 1 c) and d).

Figure 9. Visual quality comparison of reconstructed map. Left: without submap registration; Middle: with naïve submap registration; Right: with proposed submap registration method.

On the OpenLORIS-scene dataset, we conduct further qualitative evaluation on a challenging sequence of the home scene. The robot has traveled across different rooms and corridors resulting in multiple long-term LCs. Evaluations in Shi et al. (2020) depict the existing visual SLAM methods failed to work in this data due to the weak texture and large-baseline LC in many areas. As shown in Fig. 9, the map reconstructed by the baseline has the problem of unclosed long-term loop. Combining naïve submap registration method can solve the long-term loop problem but leads to a new mismatching at the wall of living room and bedroom. With the extension described in Sec. 4.3, our full system avoids the misalignment and reconstructs a globally consistent map.

6.5 Evaluation on depth map

Depth Type	AbsDiff	RMSE	AbsRel	SqRel	Comp
Sensor	0.2334	0.5385	8.899	13.060	97.94
Smoothed	0.2047	0.3742	7.907	5.678	95.94
Optimized	0.1916	**0.3614**	7.643	5.594	99.61
Refined	**0.1912**	0.3618	**7.602**	**5.578**	**99.66**

Table 4. Evaluation results of depth map refinement on lab dataset. The AbsDiff and RMSE are given in meter, and other metrics are in percentage.

As described in Sec. 5.1, the sensor depth contaminates significant noise and has many holes without depth measurement. From Tab. 4, we conduct the depth map evaluation on the lab dataset with the reference provided by a high-accuracy laser scanner. The sensor depth has the AbsRel of 8.89 % and completeness of only 97.94 %. After smoothed by the bilateral solver, the accuracy improves to 7.91 %, but the completeness remains at a low level. By contrast, the BA optimization of depth maps improves the accuracy and predicts the depth values for the void parts, resulting in improved completeness from 95.94 % to 99.61 %. The final refined depth map has the best evaluation results with an AbsRel of 7.60 % and completeness of 99.66 %.

6.6 Evaluation on reconstructed 3D map

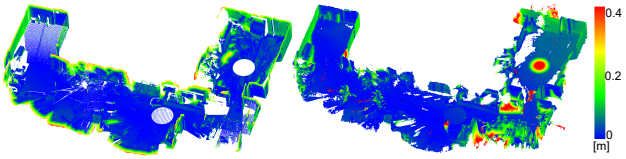

Figure 10. Reference model and the reconstructed map with color as the closest distance to each other.

Based on the trajectory estimate by the proposed SLAM system and the refined depth maps, a global map can be finally reconstructed, which is a mesh derived from the TSDF map via marching cube. The reconstructed map is evaluated on the lab dataset, which contains a sequence recorded in a multi-room environment with a millimeter-accurate reference. To enable the comparison, the reconstructed map is aligned to the reference model by selecting corresponding point pairs followed by ICP. Fig. 10 presents the reference model, and the reconstructed map with colors indicate the closest distance to the other. Following the distance metrics presented in Aanæs et al. (2016), the accuracy is measured as the mean distance from the reconstructed map to the reference model, which is 7.8 cm on average. Further, the completeness of the reconstructed map is evaluated by the distance from the reference to the reconstructed map, which is 6.0 cm on average. Both metrics together are considered as the end-to-end performance of the proposed complete system, incorporating the results of both the SLAM module and dense reconstruction module.

7. CONCLUSION AND FUTURE WORK

In this work, we proposed a robust indoor SLAM and dense reconstruction system capable of tracking the camera consistently under weak texture situations and constructing a globally consistent map. By combining the optical flow module and wheel odometer, the system can keep on tracking the camera under short-term texture lacking. Further improvement is performed by registering the TSDF submap, which ultimately constitutes a globally consistent map. The extensive experiments show that the proposed system has the best trajectory estimation result on both the self-collected dataset and synthesized data on the HM3D dataset. The mean ATE errors are reduced from 2.6 cm to 2.3 cm and 22 cm to 2 cm on these two datasets. The final reconstructed map has an average accuracy of 7.8 cm which presents the compound error of both the SLAM and dense reconstruction modules. In the future, we intend to enhance the depth estimation capability of the framework by leveraging the synthesized depth map ground-truth to train an end-to-end depth estimation network. In addition, we aim to extract semantic information from the reconstructed map, which we believe will be fairly useful to many robot tasks such as interactive navigation.

ACKNOWLEDGEMENTS

We would like to thank Rongwei Guo and Junya Hu for supporting the collection of the lab dataset. We also thank the authors of Teed and Deng (2021) for open-sourcing their excellent work and the generous release of the pre-trained model.

REFERENCES

Aanæs, H., Jensen, R. R., Vogiatzis, G., Tola, E., Dahl, A. B., 2016. Large-scale data for multiple-view stereopsis. *International Journal of Computer Vision*, 120(2), 153–168.

Barron, J. T., Poole, B., 2016. The fast bilateral solver. *European Conference on Computer Vision*, Springer, 617–632.

Blaser, S., Cavegn, S., Nebiker, S., 2018. Development of a portable high performance mobile mapping system using the robot operating system. *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*, 4(1).

Bujanca, M., Shi, X., Spear, M., Zhao, P., Lennox, B., Lujan, M., 2021. Robust SLAM Systems: Are We There Yet? *arXiv preprint arXiv:2109.13160*.

Campos, C., Elvira, R., Rodriguez, J. J. G., M. Montiel, J. M., D. Tardos, J., 2021. ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual–Inertial, and Multimap SLAM. *IEEE Transactions on Robotics*, 37(6), 1874–1890.

Curless, B., Levoy, M., 1996. A volumetric method for building complex models from range images. *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*.

Eigen, D., Puhrsch, C., Fergus, R., 2014. Depth map prediction from a single image using a multi-scale deep network. *Advances in neural information processing systems*, 27.

Engel, J., Koltun, V., Cremers, D., 2017. Direct sparse odometry. *IEEE transactions on pattern analysis and machine intelligence*, 40(3), 611–625.

Ericson, C., 2004. *Real-Time Collision Detection*. CRC Press, Inc., USA.

Giancola, S., Valenti, M., Sala, R., 2018. *A survey on 3D cameras: Metrological comparison of time-of-flight, structured-light and active stereoscopy technologies*. Springer.

Handa, A., Whelan, T., McDonald, J., Davison, A. J., 2014. A benchmark for rgb-d visual odometry, 3d reconstruction and slam. *2014 IEEE international conference on Robotics and automation (ICRA)*, IEEE, 1524–1531.

Heng, L., Li, B., Pollefeys, M., 2013. Camodocal: Automatic intrinsic and extrinsic calibration of a rig with multiple generic cameras and odometry. *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE.

Lorensen, W. E., Cline, H. E., 1987. Marching cubes: A high resolution 3D surface construction algorithm. *ACM siggraph computer graphics*, 21(4), 163–169.

Maboudi, M., Bánhid, D., Gerke, M., 2018. Investigation of geometric performance of an indoor mobile mapping system. *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences*, 42(2).

Nebiker, S., Meyer, J., Blaser, S., Ammann, M., Rhyner, S., 2021. Outdoor Mobile Mapping and AI-Based 3D Object Detection with Low-Cost RGB-D Cameras: The Use Case of On-Street Parking Statistics. *Remote Sensing*, 13(16), 3099.

Nießner, M., Zollhöfer, M., Izadi, S., Stamminger, M., 2013. Real-time 3D reconstruction at scale using voxel hashing. *ACM Transactions on Graphics (ToG)*, 32(6), 1–11.

Qin, T., Li, P., Shen, S., 2018. VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator. *IEEE Transactions on Robotics*, 34(4), 1004-1020.

Ramakrishnan, S. K., Gokaslan, A., Wijmans, E., Maksymets, O., Clegg, A., Turner, J., Undersander, E., Galuba, W., Westbury, A. et al., 2021. Habitat-Matterport 3D Dataset (HM3D): 1000 Large-scale 3D Environments for Embodied AI. *arXiv preprint arXiv:2109.08238*.

Reijgwart, V., Millane, A., Oleynikova, H., Siegwart, R., Cadena, C., Nieto, J., 2020. Voxgraph: Globally Consistent, Volumetric Mapping Using Signed Distance Function Submaps. *IEEE Robotics and Automation Letters*.

Sarlin, P.-E., DeTone, D., Malisiewicz, T., Rabinovich, A., 2020. Superglue: Learning feature matching with graph neural networks. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 4938–4947.

Savva, M., Kadian, A., Maksymets, O., Zhao, Y., Wijmans, E., Jain, B., Straub, J., Liu, J. et al., 2019. Habitat: A Platform for Embodied AI Research. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

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

Shan, Z., Li, R., Schwertfeger, S., 2019. RGBD-inertial trajectory estimation and mapping for ground robots. *Sensors*, 19(10), 2251.

Shi, X., Li, D., Zhao, P., Tian, Q., Tian, Y., Long, Q., Zhu, C. et al., 2020. Are we ready for service robots? the OpenLORIS-Scene datasets for lifelong SLAM. *2020 International Conference on Robotics and Automation (ICRA)*.

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

Szeliski, R., 2010. *Computer vision: algorithms and applications*. Springer Science & Business Media.

Teed, Z., Deng, J., 2021. DROID-SLAM: Deep Visual SLAM for Monocular, Stereo, and RGB-D Cameras. *Advances in neural information processing systems*.

Wang, W., Zhu, D., Wang, X., Hu, Y., Qiu, Y., Wang, C., Hu, Y., Kapoor, A., Scherer, S., 2020. Tartanair: A dataset to push the limits of visual slam. *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE.

Wu, K., Guo, C. X., Georgiou, G. A., Roumeliotis, S. I., 2017. VINS on wheels. *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 5155-5162.

Zhang, Y., Khamis, S., Rhemann, C., Valentin, J., Kowdle, A., Tankovich, V. et al., 2018. Activestereonet: End-to-end self-supervised learning for active stereo systems. *Proceedings of the European Conference on Computer Vision (ECCV)*.

Zhou, T., Brown, M., Snavely, N., Lowe, D. G., 2017. Unsupervised learning of depth and ego-motion from video. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1851–1858.