HeteroFusion: Dense Scene Reconstruction Integrating Multi-sensors

Sheng Yang, Beichen Li, Minghua Liu, Yu-Kun Lai, Leif Kobbelt, Shi-Min Hu

Abstract—We present a novel approach to integrate data from multiple sensor types for dense 3D reconstruction of indoor scenes in realtime. Existing algorithms are mainly based on a single RGBD camera and thus require continuous scanning of areas with sufficient geometric features. Otherwise, tracking may fail due to unreliable frame registration. Inspired by the fact that the fusion of multiple sensors can combine their strengths towards a more robust and accurate self-localization, we incorporate multiple types of sensors which are prevalent in modern robot systems, including a 2D range sensor, an inertial measurement unit (IMU), and wheel encoders. We fuse their measurements to reinforce the tracking process and to eventually obtain better 3D reconstructions. Specifically, we develop a 2D truncated signed distance field (TSDF) volume representation for the integration and ray-casting of laser frames, leading to a unified cost function in the pose estimation stage. For validation of the estimated poses in the loop-closure optimization process, we train a classifier for the features extracted from heterogeneous sensors during the registration progress. To evaluate our method on challenging use case scenarios, we assembled a scanning platform prototype to acquire real-world scans. We further simulated synthetic scans based on high-fidelity synthetic scenes for quantitative evaluation. Extensive experimental evaluation on these two types of scans demonstrate that our system is capable of robustly acquiring dense 3D reconstructions and outperforms state-of-the-art RGBD and LiDAR systems.

Index Terms—Reconstruction, Sensor fusion, Robotics.

1 Introduction

The continuing development of 3D reconstruction systems [1], [2], [3], [4] and the growth of available dense scenes [5], [6] have significantly improved modern scene understanding and manipulation techniques [7], [8]. However, the current data acquisition process still heavily relies on experienced users to hold and smoothly move the RGBD camera, which involves high labor costs. In order to support cost-effective mass acquisition of 3D scenes, delegating scanning missions to robots is highly demanded.

In order to achieve an automatic acquisition and reconstruction scheme, these modern reconstruction algorithms are also required to be improved for cooperating with modern motion planning strategies [9], [10], since their performance may decline significantly when deployed for vehicle scanning instead of hand-held scanning. This is essentially due to a reduced degree-of-freedom for camera motion that restricts the RGBD sensor from continuously focusing on regions with sufficient geometric details, i.e., the actions of robots are not as flexible as humans for staying at good shooting views containing sufficient registration hints for localization and mapping. For example, when crossing through different regions of interest in indoor scenes, the robot may choose a relatively clear path, where currently available commodity depth sensors, whose precise scanning range is short and field-of-view is narrow, are not able to capture sufficient registration hints for tracking.

In robotics, with an aim to enhance localization, pioneering work integrates multiple sensors for a wide range of diverse robotic perception tasks, as summarized in [11], which shows a feasible strategy to solve the above-mentioned challenge. For indoor scenarios, 2D laser scanners are the preferable choice for localizing the chassis of wheeled robots on account of both cost and effectiveness. Gmapping [12] and the state-of-the-art Cartographer [13] are two profound systems coupling laser frames, inertial measurements, and wheel encoders for reconstructing planar occupancy maps. Recently, several systems considering both visual and laser information were also proposed for a wide variety of scenarios such as Unmanned Aerial Vehicles (UAVs) [14], autonomous driving vehicles [15], [16], and indoor positioning tasks [17], [18]. Comparatively, in the field of reconstruction, both an appropriate data structure for precisely fusing multiple measurements and a higher accuracy of localization are required for generating high-fidelity dense representations, whereas current multi-sensor fusion methods [11] have not put these sensors into a unified optimization process, leading to insufficient utilization of advantages from different types of sensors in both pose estimation and loop closure handling.

Inspired by the general idea that multi-sensor fusion is beneficial for promoting the quality and robustness of localization and mapping [11], we present a robust real-time dense reconstruction system coupling information from an RGBD camera, a horizontally placed 2D laser scanner, an inertial measurement unit (IMU), and wheel encoders, which are typically equipped on indoor robots. The main contributions of this paper are:

- *Sheng Yang and Minghua Liu are with the Department of Computer Science and Technology, Tsinghua University, China. E-mail: {shengyang93fs,liumh413}@gmail.com*
- *Beichen Li is with the Computational Fabrication Group, Massachusetts Institute of Technology, Cambridge MA, USA. E-mail: beichen@mit.edu*
- *Yu-Kun Lai is with the School of Computer Science & Informatics, Cardiff University, UK. E-mail: Yukun.Lai@cs.cardiff.ac.uk*
- *Leif Kobbelt is with Computer Graphics Group, RWTH Aachen University, Germany. E-mail: kobbelt@cs.rwth-aachen.de*
- *Shi-Min Hu is with the Department of Computer Science and Technology, Tsinghua University, China. Email: shimin@tsinghua.edu.cn*

- We present a novel real-time dense scene reconstruction system for robotic scanning using multimodal sensors, which outperforms the generic way of multi-sensor fusion [11], [19]. Specifically, we replace the occupancy grid by the truncated signed distance field (TSDF) representation for 2D laser frames, so as to reformulate the cost function in the pose estimation stage for maintaining better accuracy.
- We propose a new pose evaluation classifier considering features derived from both sensor readings and the progress of pose estimation. Such a classifier helps to determine correct loop closures used for reducing the cumulated drifts from the sequential frame-to-model registration.
- A benchmark for evaluating the quality of mesh reconstruction is developed, where the robotic scanning process is simulated using synthetic scenes for quantitative evaluation. The benchmark will be made publicly available for facilitating future research.

In order to test the performance of our proposed algorithm, we also assembled a simple robot platform (Fig. 7) for scanning real-world scenes. Extensive evaluations on both real and simulated scans demonstrate that our proposed system, which tightly couples laser and RGBD measurements for a unified tracking and loop optimization process, is capable of maintaining sufficient accuracy for indoor scenarios, outperforming several state-of-the-art reconstruction methods [2], [3], [13], [20], even when they are enhanced with initial pose hints directly provided by a classical probabilistic approach [19] for coupling multiple sensors.

2 Related Work

In this section, we first review dense reconstruction techniques for indoor scenes, and further discuss relevant multi-sensor systems in the field of simultaneous localization and mapping (SLAM).

Dense Scene Reconstruction. As a milestone in real-time dense reconstruction, KinectFusion [1] using consumer level RGBD cameras has aroused great interests in the graphics community. This system uses TSDF volumes to store the reconstructed scenes, achieving real-time tracking and integration with the help of GPU.

In order to enlarge the size of reconstruction, the original KinectFusion has to reduce the resolution of the volumes due to the limited capacity of the graphical memory. Whelan et al. [21] present a strategy to perform memory swapping according to the current sensor pose, where distant voxels are stored on the host and nearby voxels are loaded in GPU. Such a strategy is further enhanced by a voxel hashing data structure [22] that significantly improves the utilization of memory, based on the observation that surface voxels are sparsely distributed in common scenes.

In another aspect, the accuracy of sequential tracking is also continually improved. In the field of reconstruction, estimating sensor poses is performed through frame-to-model registration (actually with ray-casted frames), and the geometric cost from depth frames for such registration is enhanced with an additional photometric cost [23] based on color frames. Image pyramids [24] are also involved to present coarse-to-fine registration for faster convergence. Recently, sparse correspondences of keypoints were also taken into consideration [3] to avoid falling into erroneous correspondences.

For these pipelines, the most common way of integrating heterogeneous sensors is to use them for providing a good initial estimation of relative transformations. This strategy has been used in reconstruction with mobile devices [25], where 3 degree-of-freedom prediction of consecutive rotations inferred via a gyroscope is used to initialize the pose iteration. However, since they are only used for initialization, there is no guarantee that such hints will eliminate poor alignment and tracking loss during the optimization process (see Sec. 4.2 for a comparison of derived systems utilizing such a strategy). Therefore, it is necessary to enhance the prediction in a tightly-coupled manner, which we study in this paper. This is especially appropriate for deploying such systems on robots, where the scanning path can easily contain regions with limited features.

Another crucial feature for maintaining global consistency of the reconstruction is the loop closure handling, which involves loop detection and optimization. In the detection stage, two methods are commonly used for detecting loops with RGBD sensors, namely bag-of-words [26] and Randomized Ferns [27]. Regarding optimization strategies, Whelan et al. [2] propose to use scene deformation of surfels according to both temporal and fern constraints, while Dai et al. [3] perform factor graph optimization among submaps and implement scene re-integration for updating scenes. From another point of view, Kähler et al. [20] simultaneously track a single frame on all its related submaps to construct constraints for pose graph factors.

In fact, the robustness of back-end optimization relies heavily on the correctness of these constraints. Simply thresholding the percentage of inliers and the RMSE (root-mean-square error) of a registration attempt [2] does not generalize well for repetitive structures in a scene. Kähler et al. [20] address this problem via training a classifier considering features among the registration process. We propose to enhance this by further considering features from heterogeneous sensors. Moreover, additional information provided by these sensors is also able to filter out erroneously detected loops with reliable, despite coarse, global positioning hints.

Multi-sensor SLAM Systems. In the fields of robotics and computer vision, reconstructing maps under pose uncertainty is often addressed as a Simultaneous Localization and Mapping (SLAM) problem. We kindly refer readers to an insightful survey [11], which divides those available systems into several categories, where the two most prevalent categories are probabilistic approaches such as extended Kalman filter (EKF) [19], and maximum-a-posteriori (MAP) estimation such as factor graph formulation [28], [29].

The classical probabilistic approaches are still popular for multi-sensor fusion, such as visual-inertial systems [30] for indoor scenes and integrated navigation systems (INS, mostly consisting of GPS/RTK and inertial sensors) [31] for outdoor scenarios. These systems incorporate inertial measurements into the ego-motion estimation with pre-calibrated extrinsics or online calibration [32], and the uncertainty of sensor measurements quantified as covariances is also involved in confidence-based pose prediction. For example, Chow et al. [33] propose to use implicit iterative extended Kalman filter (IEKF) for coupling sensor states from a 2D laser scanner, an IMU, and two RGBD cameras. Deilamsalehy et al. [14] assemble a 2D laser scanner, an IMU, and a camera on a UAV and use the EKF for robust indoor navigation. However, these approaches fuse available sensors by concluding their motion state based on multiple individually estimated odometry, while our method chooses to simultaneously and densely track RGBD and laser frames (see Sec. 3.2).

From the perspective of the MAP estimation, variables are estimated by computing the assignment of variables that attains the maximum of the posterior. In this method, odometry estimation is solved through a cost function integrating information from multiple sensors [34], where measurements are regarded as priors and camera as well as laser frames are used for scan-matching based on ICP (iterative closest point) algorithms [35]. For instance, Wen et al. [17] perform a cascaded ICP based on sparse and dense visual correspondences, where laser scans are fed into Gmapping [12] for a 2D initial guess. Another capability of the MAP estimation is that such methods are capable of alleviating accumulated drifts through detected loops: In the factor graph approaches, sensor poses as well as landmark positions, and even calibration parameters, are continuously optimized under the constraints of odometry, frame registration, and landmark/loop-closure observations [11]. But factor graph approaches are not suitable for real-time dense reconstruction systems, since they are hard to be parallelized. Based on the 2D TSDF structure for organizing laser measurements, we can use the MAP strategy in our tracking process by considering both dense visual and laser correspondences.

Once sensor poses are solved for and optimized, the final map can be constructed according to geometric transformations. Both depth maps and laser scans are typical choices for stitching dense point clouds to acquire a full representation of scenes. Although laser scanners have better accuracy in comparison to commodity depth sensors, covering the whole surface with 2D laser frames is extremely tedious, regardless of how they are placed (vertically [36] or through a flexible spring [37]). In fact, the most common way of assembling such a scanner on a wheeled robot is to secure it with the chassis horizontally, which is able to provide registration hints that largely benefit 2D localization accuracy. Unlike traditional SLAM techniques where the focus is sparse reconstruction and localization, our system aims to produce robust real-time dense reconstruction. Therefore, utilizing information from the RGBD camera for dense 3D reconstruction is the most suitable way for indoor scenes. Moreover, although depth sensors produce noisy data, such noise can be statistically diminished in real-time during the integration of TSDF volumes.

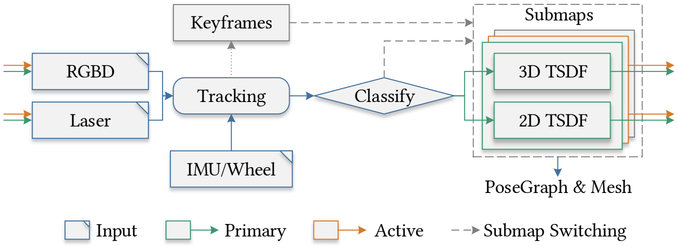


Figure 1. Overview of our proposed system. The backbone of our pipeline is colored in blue and green, while the rest are added for handling cumulative drifts through loop-closures.

3 Method

The data flow of our framework is shown in Fig. 1 with the backbone colored in blue and green. We choose to use the TSDF volumetric representation for representing both 2D and 3D scenes reconstructed through laser and RGBD frames, respectively (see Sec. 3.1 for the data structure of 2D volumes). For input messages captured by different sensors, we synchronize color ($\mathcal{C}$), depth ($\mathcal{D}$), and laser ($\mathcal{L}$) frames as batches for tracking through an *Approximate Time* strategy [38]. These three types of frames are simultaneously registered to their corresponding ray-casted frames generated through these volumetric representations (Sec. 3.2), where possible additional hints from the IMU and wheel encoders can contribute to a better initial status for subsequent iterations. After such an estimated pose is acquired, the system uses an evaluation module (Sec. 3.3) to judge its reliability and decide to either fuse the current batch of frames into these maps or report a tracking failure event. Since such a sequential tracking scheme may suffer from cumulative drifts, we follow the strategy proposed by Kähler et al. [20] to enhance the backbone, which uses the pose graph optimization to handle it through loop-closures.

Loop Closure Optimization Through Submaps. In order to handle cumulative drifts and maintain the global consistency of reconstruction, we choose to use mutliple volumetric representations to fuse both 2D and 3D measurements (Fig. 1-Right). Overall, these synchronized batches of frames are split into multiple groups, where each group has a pair of 3D and 2D TSDF volumes for integrating RGBD and laser frames, respectively. We denote such a pair of volumes as a *submap*, and these submaps are regarded as rigid parts in the global deformation for reducing cumulative drifts during reconstruction. For determining the partition of frames, these submaps are divided into three categories, namely *primary* (green), *active* (orange), and *inactive* (gray), and their states are continuously updated during scanning: The current batch of frames will be simultaneously registered to all submaps that are currently *primary* or *active*, and multiple successful registrations among different submaps will update the constraints of their relative poses in the pose graph [29] for performing non-rigid global deformation and final integration. At any time during reconstruction, there exists at most one *primary* submap which integrates the current batch of frames. Once such a *primary* submap has fused a certain amount of frames (whose accumulated translation/rotation exceeds 0.3m/30° in our implementation), its status will be switched into *active*, and another new *primary* map will be created to continue the integration. A detected tracking failure event from those *active* submaps will turn them into *inactive*. Besides, the system records a set of keyframes during the scanning process through Randomized Ferns [27], and successful retrieval of frames with similar appearance as the current frames is regarded as a loop event, i.e., a revisit of a scanned place, which will *reactivate* the submap it belongs to. We add additional constraints as discussed in Sec. 3.4 in such a multi-sensor scenario for better *reactivation* performance.

Definitions and Coordinates. During reconstruction, we have several sensors and submaps as well as their local coordinates, as shown in Fig. 2. On the robot (Fig. 2-left), Raw sensor readings (green circles) including registered RGBD frames ($\mathbb{D}$), laser scans ($\mathbb{L}$), and inertial measurements ($\mathbb{I}$) are represented in their device coordinate systems. For each submap (Fig. 2-right), two coordinate systems (red circles) are used to represent the volume, namely in 3D for RGBD ($\mathbb{W}$) and 2D for laser scans ($\mathbb{P}$). During the initialization of each submap i, $\mathbb{W}_i$ is assigned the same as the current $\mathbb{D}$, while $\mathbb{P}_i$ is assigned the same as a rectified coordinate system $\mathbb{G}$ (the yellow circle), which is obtained by reprojecting $\mathbb{L}$ based on the estimated gravity direction. Such rectification is helpful for registering laser frames based on the Manhattan assumption [39], i.e., indoor scenes consist of dominantly surfaces with orthogonal normal directions. To acquire the gravity orientation, we calculate the average of recent acceleration readings [13], where the result is formulated as a transformation $\mathrm{T}_i^{\mathbb{IG}} \in SO(3)$. For convenience, we denote by $\mathrm{T}^{\mathbb{XY}}$ the transformation that converts points represented in the coordinate system $\mathbb{X}$ to $\mathbb{Y}$.

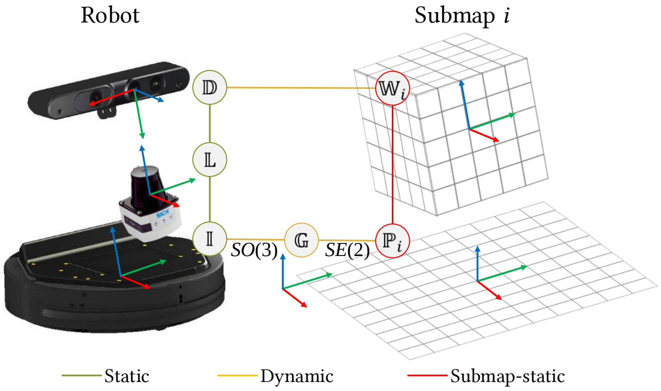


Figure 2. Coordinate systems involved in our pipeline, where x, y and z axes are colored in red, green and blue, respectively. We use such a rectified coordinate system $\mathbb{G}$ for better establishing 2D volumes.

Hence, from these established coordinates, we can divide all transformations into three types, namely *static* (green), *submap-static* (red), and *dynamic* (yellow): *Static* transformations, including extrinsic transformations between different sensors, are pre-calibrated before scanning. *Submap-static* transformations for each pair of 3D and 2D volumes $\mathrm{T}_i^{\mathbb{WP}}$ can be calculated using the estimated orientation of gravity during its initialization, as:

$$\mathrm{T}_i^{\mathbb{WP}} = \mathrm{T}^{\mathbb{DI}} \oplus \mathrm{T}_i^{\mathbb{IG}}, \tag{1}$$

where $\oplus$ denotes the motion composition operator [40]. *Dynamic* transformations include the two which are most concerned: $\mathrm{T}_{i,t}^{\mathbb{DW}}$ and $\mathrm{T}_{i,t}^{\mathbb{LP}}$, i.e., the poses between sensors and their corresponding maps, where t is the timestamp.

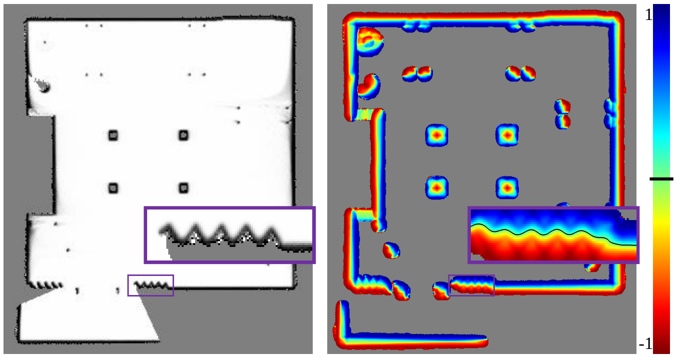


Figure 3. Comparisons between two candidate data structures for storing laser measurements. In each cell, the occupancy grid (left) stores the possibility of having an obstacle, whereas the TSDF volume (right) stores the signed distance to the scanned surface. Both maps are fused through the same inputs and trajectory at the same resolution.

We optimize $\mathrm{T}_{i,t}^{\mathbb{DW}}$ with multi-sensor tracking (see Sec. 3.2 for details). Once this is solved, $\mathrm{T}_{i,t}^{\mathbb{LP}}$ can be calculated using:

$$\mathrm{T}_{i,t}^{\mathbb{LP}} = \mathrm{T}^{\mathbb{LI}} \oplus \mathrm{T}_t^{\mathbb{IG}} \oplus \pi(\mathrm{T}_t^{\mathbb{GI}} \oplus \mathrm{T}^{\mathbb{ID}} \oplus \mathrm{T}_{i,t}^{\mathbb{DW}} \oplus \mathrm{T}_i^{\mathbb{WP}}), \tag{2}$$

where the transformation chain in $\pi(\cdot)$ is the actual 3D transformation from $\mathbb{G}$ to $\mathbb{P}_i$, and $\pi(\cdot)$ is a projection operation for a 3D transformation, which only keeps the original (x, y, ϕ) as the position and yaw for generating a planar transformation. Such an approximation $\mathrm{T}_{i,t}^{\mathbb{GP}} \in SE(2)$ is made because the horizontally placed laser scanner can only be used to reveal the planar motion of the robot chassis.

3.1 2D volume representation for laser scans

There are several strategies for integrating planar laser scans. The most prevalent choice is the occupancy grid [12], [13], which has proven its strength for localization and navigation purposes. However, in the 3D reconstruction scenario which requires higher accuracy, this kind of representation is not capable of conveying the exact location of scanned surfaces. An illustrative figure (Fig. 3) for comparing the data structures of the occupancy grid (left) and the TSDF volume (right) shows their difference: The occupancy grid stores the probability of the existence of an obstacle in each cell, with a binary Bayes filter to update its value through multiple measurements. Hence, such a data structure limits the accuracy of the precision of surfaces to the *grid* level, and it is also unable to model relations between neighboring cells, i.e., noisy measurements that have been mistakenly categorized into erroneous cells cannot correctly contribute to the reconstruction statistically. In comparison, the TSDF volume we choose to use stores the truncated signed distance to the surface in each cell, which helps us find the actual location of the surface (as the zero level line).

Therefore, we replace the occupancy grid by the 2D TSDF volume representation derived from its 3D version [1] along with a voxel hashing strategy [22]. Similar to 3D representations for depth information, this TSDF volume stores the exact location of obstacles and is able to integrate and ray-cast laser frames. In our implementation, each voxel block is composed of 256 voxels and mapped to the hash table with its size set as 2048. Such a configuration is sufficient in most cases for managing stored measurements.

Frame Integration. Given a laser scan $\mathcal{L}_t$ and a pose prediction $\mathrm{T}_{i,t}^{\mathbb{DW}}$ after tracking and evaluation, those hit points are transformed into $\mathbb{P}_i$ using Equ. 2, where their z values are dropped in the integration process. For each hit point, it will affect a certain range of cells. We set the band of its influence as $\sigma_u = 8$cm and the grid resolution as $\sigma_r = 2$cm according to the distribution of noise and the density of hit points, respectively. For updating the status of each cell, it is assigned the average TSDF value computed using all its corresponding measurements.

Ray Casting. In the ray-casting stage, corresponding voxels under a given pose are retrieved for constructing the ray-casted laser frame in $\mathbb{L}$. Here, rectification is required to recover those abandoned z values, which is achieved via the following relation:

$$l^{\mathbb{L}} = (l^{\mathbb{P}} + \lambda_l \cdot z) \oplus \mathrm{T}_{i,t}^{\mathbb{PL}}, \tag{3}$$

where $z = (0,0,1,0)^\top$ in homogeneous coordinates is added for such compensation, and λ_l can be solved using the constraint that $l_z^{\mathbb{L}} = 0$. During the scanning process, the horizontally placed laser sensor only touches a narrow slice of the indoor scene at its height. Similarly, rectification of its corresponding normal $n^{\mathbb{L}}$ is performed by assuming $n_z^{\mathbb{L}} = 0$. Fig. 4 illustrates the rectification process of points (left) and their normals (right). Such rectification can be considered valid under the Manhattan assumption [39] as discussed before.

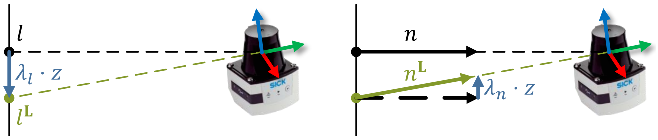


Figure 4. Recovery of the location (left) and normal (right) of a laser hit point through the Manhattan assumption [39].

3.2 Pose estimation with multimodal sensors

In our scenario, we have two types of sensors: (1) perceptual sensors such as the laser scanner and the RGBD camera whose ego-motion estimation is calculated through scanned scene components, and (2) motion sensors such as the IMU and the wheel encoders that provide distance, speed, and acceleration measurements. A common approach for tracking is to compute the odometry from each perceptual sensor individually, and fuse their output through EKF [17], [19], [41]. However, such a strategy cannot fully utilize the characteristics of each sensor: Horizontally placed laser sensors with a wide field-of-view can produce high-quality range information for robust 2D localization, whereas RGBD sensors, on the other hand, produce precise 3D localization when both a good initial solution is given and sufficient surface details are captured. Based on the 2D TSDF volumetric representation (Sec. 3.1), we are able to combine the laser and RGBD measurements in a unified registration process and achieve reciprocity. Fig. 5 presents a visual comparison between different sensor fusion strategies.

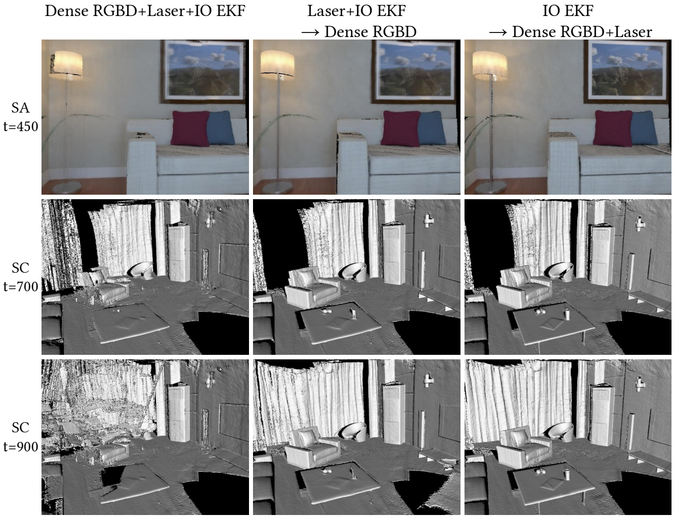


Figure 5. Comparisons of candidate sensor fusion strategies. Left: a generic way that fuses all the sensors by EKF [17], [19], [41]. Middle: fusing the laser, IMU, and wheel encoders by EKF, and then using its output as an initial guess for dense RGBD tracking. 'IO' stands for the IMU and wheel encoders together. Right: our method. Details such as the wall picture in SA and the bottle in SC can be successfully recovered during our reconstruction. Performance of those loosely-coupled strategies becomes worse when visual odometry provides erroneous results (Bottom-Left), whereas an initial guess (Bottom-Middle) cannot fully recover the problem. Tab. 2 further presents a quantitative comparison of different publicly available reconstruction algorithms, those methods combined with the fusion strategy in the middle column and our method.

Our tracking process is applied to all active submaps for each group of synchronized messages. In most cases, it contains two stages, i.e., initial prediction and iterative refinement, except when a submap is just reactivated by loop closure events: In such a case since the last activity of the submap is temporally distant, neither inertial nor wheel encoder measurements can be taken into consideration. Hence, the tracker will directly start with the latter stage. For simplicity, we ignore the notation of submap index i here in the equations below.

Initial estimation. The initial pose $\widehat{\mathrm{T}}_t^{\mathbb{DW}}$ for the frame at time t is estimated using inertial and wheel encoder measurements. We use extended Kalman filter (EKF) [19] with a 6D model (3D position and 3D orientation) for discrete-time prediction in the primary submap, and then the predicted relative transformation is propagated to other active submaps. Such integration benefits the subsequent optimization with a better initial guess.

Iterative refinement. Starting from the initial guess, we retrieve the ray-casted depth ($\mathcal{D}_{t-1}'$) and color ($\mathcal{C}_{t-1}'$) images from the 3D volume as well as the laser scan ($\mathcal{L}_{t-1}'$) from the 2D volume for each submap. We then iteratively refine the final prediction of the current pose $\mathrm{T}_t^{\mathbb{DW}}$ via the following optimization:

$$\underset{\mathrm{T}_t^{\mathbb{DW}}}{\arg\min}\, E(\mathrm{T}_t^{\mathbb{DW}}) = E_{\mathcal{D}}(\mathrm{T}_t^{\mathbb{DW}}) + w_c E_{\mathcal{C}}(\mathrm{T}_t^{\mathbb{DW}}) + w_l E_{\mathcal{L}}(\mathrm{T}_t^{\mathbb{LP}}), \tag{4}$$

where the balancing weight between color and depth frames w_c is set to 0.1 as previous works [2], [24], and w_l is continuously adjusted over iterations which will be discussed later. Each $E_{\mathcal{X}}$ ($\mathcal{X} \in \{\mathcal{C}, \mathcal{D}, \mathcal{L}\}$) takes its corresponding input $\mathcal{X}_t$ and the ray-casted frame $\mathcal{X}_{t-1}'$ into consideration. $E_{\mathcal{C}}$ and $E_{\mathcal{D}}$ are the traditional costs for registering RGBD frames, while $E_{\mathcal{L}}$ is introduced for measuring planar point-to-model geometric error utilizing laser scans in robotic scenarios.

Laser scan cost $E_{\mathcal{L}}$. Previous 2D LiDAR SLAM methods use the occupancy grid representation [12], [13], so the laser scan cost is measured by the overlap of hit points with the occupancy map. As a consequence of using this data structure, the resolution of such grids will affect the surface matching progress in frame registration. Our approach instead uses the 2D TSDF representation, so each hit point can find the exact position of its corresponding surface point, achieving registration at the *sub-grid* level. The cost is defined as:

$$E_{\mathcal{L}}(\mathrm{T}_t^{\mathbb{LP}}) = \sum_{(l_p^{\mathbb{P}}, l_q^{\mathbb{L}}) \in \mathcal{K}_{\mathcal{L}}} \left\| (l_p^{\mathbb{P}} - l_q^{\mathbb{L}} \oplus \mathrm{T}_t^{\mathbb{LP}}) \cdot n_p^{\mathbb{P}} \right\|^2, \qquad (5)$$

where $\mathcal{K}_{\mathcal{L}}$ is the set of laser correspondences between $\mathcal{L}'_{t-1}$ and $\mathcal{L}_t$ obtained by projective data association [1]. $l_p^{\mathbb{P}} \in \mathcal{L}'_{t-1}$ and $l_q^{\mathbb{L}} \in \mathcal{L}_t$ are a pair of corresponding points, and $n_p^{\mathbb{P}}$ is the planar surface normal retrieved at $l_p^{\mathbb{P}}$.

This cost is optimized along with RGBD costs in the same iteration step, where it simply contributes a 3 degree-of-freedom planar motion to the refinement, which is orthogonal to the current gravity orientation.

RGBD costs $E_{\mathcal{D}}$ and $E_{\mathcal{C}}$. We refer to the state-of-the-art dense correspondence error function as [24] to calculate geometric and photometric errors from RGBD images. The geometric error is calculated as:

$$E_{\mathcal{D}}(\mathbf{T}_t^{\mathbb{DW}}) = \sum_{(d_p^{\mathbb{W}}, d_q^{\mathbb{D}}) \in \mathcal{K}_{\mathcal{D}}} \left\| (d_p^{\mathbb{W}} - d_q^{\mathbb{D}} \oplus \mathrm{T}_t^{\mathbb{DW}}) \cdot n_p^{\mathbb{W}} \right\|^2, \qquad (6)$$

where $\mathcal{K}_{\mathcal{D}}$ is the set of corresponding depth pixels $d_p^{\mathbb{W}} \in \mathcal{D}'_{t-1}$ and $d_q^{\mathbb{D}} \in \mathcal{D}_t$. $n_p^{\mathbb{W}}$ is the 3D surface normal retrieved at $d_p^{\mathbb{D}}$. Similarly, the photometric error is calculated as:

$$E_{\mathcal{C}}(\mathbf{T}_t^{\mathbb{DW}}) = \sum_{x \in \mathcal{C}'_{t-1}} \left\| \mathcal{C}'_{t-1}(x) - \mathcal{C}_t(\tau(x, \mathbf{T}_t^{\mathbb{DW}})) \right\|^2, \qquad (7)$$

where $x \in \mathcal{C}'_{t-1}$ iterates over pixels searching for their correspondences $\tau(x, \mathbf{T}_t^{\mathbb{DW}}) \in \mathcal{C}_t$ by reprojection through intrinsics, depth information and $\mathbf{T}_t^{\mathbb{DW}}$.

Iteration strategy. Image pyramids were first brought to use for reconstruction by [24], [25] for faster convergence. In our method, a similar coarse-to-fine strategy is used for adaptively incorporating laser scan information, which is crucial for avoiding poor local optimum and thus ensuring robustness. Since a laser frame contains relatively fewer pixels, establishing a pyramid for these sensors does not significantly increase the efficiency, so we only build image pyramids for RGBD images. Moreover, during our iterative process, the optimization of $E_{\mathcal{L}}$ in the earlier iterations helps robustly locate the approximate sensor poses thanks to its wide field-of-view. Hence, to obtain precise final results, the importance of RGBD is progressively increased for detailed registration relying on local surfaces. In detail, for an increasing pyramid level j, we set the weight of the laser loss term as:

$$w_{l,j} = \sigma_b \cdot \sigma_d^j, \qquad (8)$$

where $\sigma_b = 5.0$ and $\sigma_d = 0.5$ in our implementation for reducing the weight of laser cost over iterations, and 3 levels of pyramids are established for the strategy. Both σ_b and σ_d remain unchanged for all test cases including various types of common room layouts in our experiments. We further evaluate the impact of each term (Equs. 5-7) in Sec. 4.4, and the influence of adjusting such parameters in Sec. 4.5.

3.3 Pose evaluation

In our system, the integration of different types of sensors is capable of presenting additional features for evaluating the correctness of a tracking attempt. Specifically, both RGBD and laser frames contribute to the frame registration process. Therefore, those indicators in the process can effectively measure the quality of surface matching. As another feature component, information from wheel encoders and IMU can be regarded as coarse but reliable measurements for short-term robot motions. In summary, we extract the features listed in Tab. 1 for considering whether a tracking attempt is successful.

Feature	Description	Dim.
F_1	The determinant of the Hessian in $E_{\mathcal{D}}$	2
F_2	The final residual of $E_{\mathcal{D}}$	1
F_3	The percentage of inlier pixels in $E_{\mathcal{D}}$	1
F_4	The determinant of the Hessian in $E_{\mathcal{L}}$	2
F_5	The final residual of $E_{\mathcal{L}}$	1
F_6	The percentage of inlier pixels in $E_{\mathcal{L}}$	1
F_7	The diff. of trans. and rot. between $\widehat{\mathrm{T}}_t^{\mathbb{DW}}$ and $\mathrm{T}_t^{\mathbb{DW}}$	2
F		10

Table 1
Extracted features for pose evaluation, including their dimensions.

We next expand such 10-dimensional vectors into 50-dimensions using a χ^2 kernel map [42], and use an SVM classifier to separate success and failure tracking. The classifier is trained on our simulation dataset, including 9,348 registration attempts from 9 scenes and 35 scans simulated with data from SceneNet [43]. Those registration attempts are performed by registering frames to models integrated through the ground-truth trajectories, and these scans used in the training stage are different from those chosen for the final evaluation.

If an attempt has translational tracking error less than 2mm and rotational tracking error less than 2°, we consider such an attempt as correct. We take 75% of the set for training and the rest for testing, and obtain a classification accuracy of 99.34% on the test set. Further experiments on the pose evaluation are summarized in Sec. 4.6 with our evaluation data, illustrating the effectiveness of judging tracking states using multiple sensors.

3.4 Relocalization and loop detection

There are two aspects in the loop closure optimization that affect the final reconstruction quality of the whole scene, namely the performance of loop detection and loop refinement. While the loop refinement in our algorithm [20] uses a pose graph to perform global deformations and maintain consistency, loop detection provides relations between these submaps for constructing optimization constraints. In common reconstruction approaches [2], [20], loops are detected through the Randomized Ferns method [27], which randomly generates multiple pixel locations, applies 4-channel (i.e., RGBD) thresholds on these pixels to get a code, and uses Hamming distance for quickly measuring the similarity between frames. Since such a method is purely based on visual similarities, misjudgments occasionally happen in scenes with repetitive structures.

Since our multi-sensor pose estimation and evaluation module is more reliable than RGBD based methods, the accumulation of errors is slower and tracking loss is less likely to occur. Hence, we choose to trust the mid-term tracking process and use its result to filter erroneous loops that cannot be filtered through visual appearances. Specifically, we add additional constraints for detecting loops: When the current frame successfully finds its similar frame through the Randomized Ferns method [27], we further estimate the relative global pose between these two frames according to their belonging submaps, where only those within the thresholds (0.05m and 3° for all test scenes) are accepted for the subsequent simultaneous tracking stage. Such a criterion is only examined when the system does not lose track. A representative example comparing incorporating this criterion with the standard approach is visualized in Fig. 6. Detailed comparisons on the quality of loop detection are presented in Sec. 4.7, which demonstrate that such a strategy achieves better pose graph constraints than the previous loop detection criterion.

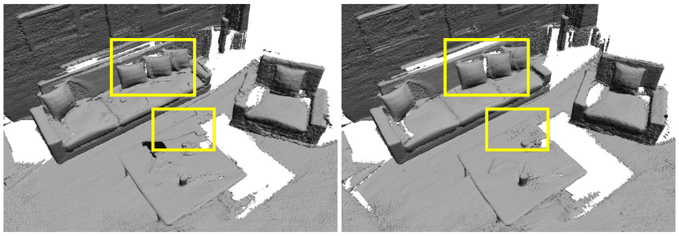


Figure 6. A representative comparative example for adding an additional constraint in the loop detection. Better global consistency (right) can be achieved compared to the original strategy (left) due to the improved correctness of pose graph factors.

4 Experiments and Results

Since there are no publicly available scans for indoor scenes with robots containing all sensors involved above, we assembled our scanning system based on a Turtlebot2 for capturing real-world scenes. Furthermore for quantitative evaluation, we simulate the scanning robot in high-fidelity synthetic scenes from ICL-NUIM [44] and SceneNet [43]. In this section, we first discuss our data acquisition system (Sec. 4.1), then show our reconstruction quality in comparison to several state-of-the-art reconstruction systems (Sec. 4.2), along with speed up testing to evaluate the robustness of methods (Sec. 4.3). We then evaluate the contribution of different types of sensors (Sec. 4.4), and the influence of several parameters (Sec. 4.5). Also, both the SVM classifier (Sec. 4.6) and the additional loop criterion (Sec. 4.7) are further evaluated. Finally, we present our running times (Sec. 4.8) and discuss limitations (Sec. 4.9).

4.1 Data acquisition

Our scanning systems for both real-world and synthetic scenes are shown in Fig. 7. For real-world scenarios, the Turtlebot2 is assembled with an elevated Xtion Pro Live for capturing 480P registered RGBD frames at 30Hz, and a SICK TiM561 for outputting 270° field-of-view planar laser scans at 15Hz. Both inertial and odometry messages are acquired from the Kobuki base at 50Hz. The RGBD camera and the 2D laser scanner are pre-calibrated using a 15×10 chessboard with each square of size 0.05m according to the method proposed by Kassir et al. [45]. For online reconstruction, sensor messages are sent to a workstation through a 5G Wireless LAN for real-time reconstruction and localization. Since our focus is to improve scene reconstruction, the robot is manually driven by sending instructions from a keyboard. 4 scenes were recorded in our scanning dataset, where 2 of them (RL1, RL2) are small living rooms and the rest 2 (RO1, RO2) are cluttered laboratory scenes.

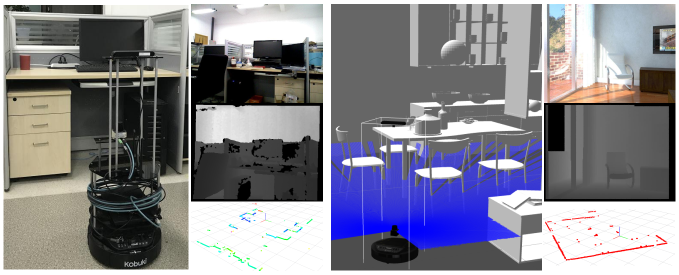


Figure 7. Our scanning system for real-world scenes (left) and synthetic scenes (right), including corresponding sample frames (color, depth and laser).

For synthetic data, we choose to render scenes mainly from the ICL-NUIM living-room [44] and SceneNet [43] due to their high-fidelity production of indoor scenes. For highly realistic simulation of robot motions and sensor noise, we utilize Gazebo, a robot simulation platform for manipulating a virtual Turtlebot2 with the same configuration of assembling and teleoperation as real-world scenarios. Gaussian noise is added to every laser sensor message with standard deviation set to 0.01m, and depth noise is simulated according to the proposed noise model in [44]. 32 scans from 7 scenes are generated in this way, i.e. ICL-NUIM living-room (SA1-7), kitchen-16 (SB1-7), living-room-11 (SC1-6), bath-room-8 (SD1-3), bed-room-2 (SE1-3), office4 (SF1-3) and office-4[1] (SG1-3). Note that the provided office rooms of ICL-NUIM do not come up with a standard mesh model for simulating collision and ray-casting laser scans, hence they are not used in our evaluation. During different scans, we use various maximum linear/angular speeds for manipulating robots (linear speed from 0.091m/s to 0.201m/s, and angular speed from 26.8°/s to 60.6°/s). Compared to the original ICL-NUIM [44] which is designed for RGBD scans, we have more scenes and trajectories for comprehensive multi-sensor evaluation.

4.2 Quality comparison

We compare our system with state-of-the-art publicly available RGBD fusion systems including KinectFusion (KF) [1], Kintinuous (KT) [21], InfiniTAM v2 with the RGBD tracker (ITv2) [25], ElasticFusion (EF) [2], InfiniTAM v3 (ITv3) [20],

[1] office4 is downloaded from *officesSceneNet* repository, while office-4 from *DownloadSceneNet* repository of SceneNet [43].

	KF	KT	ITv2	EF	ITv3	BF	eKF	eKT	eITv2	eEF	eITv3	eBF	CT	Ours
SA	0.489/24	0.286/30	0.433/39	0.099/46	0.467/39	0.081/76	0.270/58	0.203/41	0.234/60	0.098/47	0.397/51	0.074/78	0.062/91	**0.050**/**95**
SB	0.512/23	0.315/26	0.549/33	0.263/15	0.307/33	0.094/29	0.357/44	0.342/28	0.291/51	0.251/16	0.285/72	0.091/30	0.073/95	**0.045**/**99**
SC	0.568/16	0.417/18	0.548/15	0.450/04	0.246/15	0.236/05	0.326/39	0.187/47	0.345/35	0.356/06	0.297/14	0.234/05	0.094/86	**0.047**/**92**
SD	1.379/17	0.323/27	0.604/28	0.140/13	0.834/28	0.284/24	0.216/62	0.119/68	0.094/85	0.128/17	0.147/69	0.284/24	0.053/98	**0.049**/**99**
SE	0.261/48	0.230/43	0.152/56	0.066/28	0.324/56	0.225/08	0.191/63	0.134/62	0.068/94	**0.035**/32	0.317/58	0.221/08	0.065/97	0.040/**98**
SF	0.273/41	0.169/47	0.443/64	0.039/40	0.167/64	0.182/12	0.129/69	0.067/84	0.081/89	**0.034**/41	0.147/68	0.160/14	0.065/97	0.042/**99**
SG	0.807/18	0.631/14	0.365/17	0.416/05	0.351/21	0.074/06	0.617/20	0.307/31	0.633/18	0.421/05	0.275/43	0.074/06	0.079/88	**0.044**/**99**

Table 2
RMSE (in meters) and the coverage of the reconstructed scenes. Frm. - number of total frames, Spd. - speed of the robot. Values are given in the average of frames. Methods starting with ‘e’ are with enhanced pose initialization combining multimodal sensors. Best RMSE and coverage are in bold.

and BundleFusion (BF) [3]. To concentrate on the comparison of model quality, voxel hashing [22] is augmented for earlier systems (KF). We now briefly summarize different trackers. KF only considers the geometric cost $E_{\mathcal{D}}$ from depth sensors for frame registration, while ITv2 in our test has added the photometric cost $E_{\mathcal{C}}$ as a supplement for areas with low geometric hints. We choose to use the latest version of KT, where the loop optimization module is augmented into their back-end for better global consistency. EF addresses the loop-closure through scene deformation with two types of constraints. Since their scene representation is based on surfels, we use these scattered points for quantitative evaluation. Both ITv3 and BF address loop closure and are augmented with their corresponding pose evaluation module, i.e., those recognized as tracking-failure frames will not be integrated into the scene.

To better demonstrate the effectiveness of our tightly coupled registration strategy (Equ. 4), we strengthened the above-mentioned systems with a straight-forward coupling of multiple sensors as our baseline. In detail, their cost functions for optimizing registrations remain unchanged, but the initial predictions of robot states are enhanced by the EKF prediction coupling IMU, wheel encoders, and laser odometry in our method, where the laser odometry is acquired through relative transformations produced by Gmapping [12]. In addition, we also use the final output trajectory from the state-of-the-art Cartographer (CT) for integrating a mesh. We set the parameters of input depth cut-off to 4m and the voxel size to 1cm consistently for all systems and scans, while other parameters are derived from their default configurations.

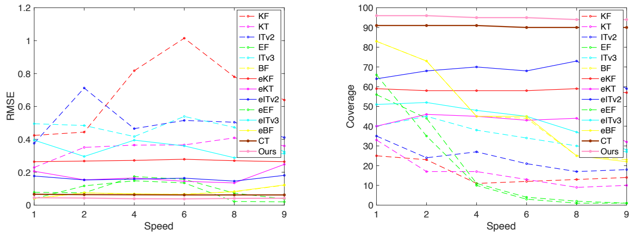
Qualitative evaluation. Qualitative reconstruction results on both real-world scans and simulated scans are shown in Fig. 9 for global consistency and Fig. 10 for detailed views between CT and ours. Despite the difficulty to perform accurate quantitative evaluations on these real-world scans, it can be easily observed from the global appearance of each result that existing RGBD systems, although enhanced with better initial predictions, are still unable to produce robust and consistent reconstruction. Both CT and our algorithm are capable of outputting globally consistent shapes, but ours outperforms CT in terms of preserving surface details since only considering 2D information is not sufficient for accurate registrations, especially when the robot is not idealistically moving on a plane. Compared to hand-held scanning which is more flexible to avoid occlusions, robot scanning in the current form has limited completeness, but it still provides a feasible way to reduce scanning costs and support dense scene understanding.

Quantitative evaluation. Two metrics are calculated for quantitative evaluation by comparing reconstructed samples (mesh models or point clouds) with the given ground truth scene: registration error and overall coverage. The traditional registration error is calculated as the RMSE between sample points to their nearest points on surfaces, while the overall percentage of coverage is used to reflect the overall effectiveness of the tracker. To calculate this metric, we first find out the covered area on the ground truth mesh through the ground truth camera trajectory, and then use a threshold (0.10m) to search its surroundings on their output scene to check whether this area has been successfully reconstructed. The low coverage rate may be due to poor reconstruction accuracy, false-positive tracking attempts, or frames being rejected by their pose evaluation module. We list the results of our overall quantitative performance in Tab. 2 averaged over the total reconstructed points.

As shown in Tab. 2, our system has shown better performance on most tested trajectories, outperforming both state-of-the-art RGBD and LiDAR systems in scene quality, even when they are enhanced with straightforwardly performed multi-sensor fusion strategies (those methods starting with ‘e’). Typically in simulation environments, scenes generated via estimated poses from Cartographer have shown comparable results, but coarse laser information is still not sufficient for high-accuracy localization and especially vulnerable to non-ideal motions (such as vibrations), which was mentioned before and visualized in Fig. 10. Admittedly, providing an initial guess for RGBD registration is able to improve their performance in most cases, but their results still lack quality, as a good initial guess cannot guarantee that the follow-up iterations will not fall into local optimum. Although EF and BF occasionally provide comparable reconstruction accuracy, they discard a substantial amount of frames (or unstable surfels in EF) as these methods are not sure if such frames are precisely located, resulting in an insufficient coverage rate.

4.3 Scanning speed evaluation

In addition, we analyze how the speed of robot motion may affect reconstruction results in different systems. 2 trajectories (SA-1 and SA-2) are chosen to simulate accelerated cases, where intermediate camera and odometry frames of these trajectories are removed to match the frame rates, and inertial sensor readings are also updated. Such simulation is beneficial in maintaining the same trajectory to make the results more interpretable, although our current implementation has a limitation that we do not simulate the motion blur of color frames. The results of running on such accelerated simulation sequences are shown in Fig. 8, averaged over total reconstructed points.



(a) RMSE (b) Coverage

Figure 8. Reconstruction quality (RMSE and coverage) when scanning speed is increased on the same trajectory.

In Fig. 8, it is obvious that previous RGBD systems are vulnerable to fast movement. Especially for confidence based fusion such as EF and BF, they remove surfels or frames which are difficult to align to maintain overall precision. As a result, their overall coverage is dramatically declined. It also reveals that augmenting initial hints in such conditions largely benefits the final quality. In summary, our method performs consistently well, even with 9 times speed up.

4.4 Effectiveness of cost terms in pose estimation

For better evaluating the strategy of combining multiple sensors in our proposed tracking module (Equ. 4) through jointly minimizing their costs (Equs. 5-7), we examine the performance of possible different combinations. In detail, we randomly choose 1000 adjacent pairs of frames on each scene (7000 in total), and use five combinations as listed in Tab. 3 to demonstrate their effectiveness, with their translational/rotational error recorded for quantitative comparison. Since synthetic scenes from SceneNet [43] (SB-SG) do not contain color information, we split the set into two parts and do not consider $E_{\mathcal{C}}$ on these scenes. According to our experiment, the combination of RGBD costs ($E_{\mathcal{D}}$ and possible $E_{\mathcal{C}}$) and the laser cost ($E_{\mathcal{L}}$) achieves the best performance on the test set, which demonstrates the effectiveness of the proposed multi-sensor pose estimation scheme.

	SA		Others	
	Tran-Err.	Rot-Err.	Tran-Err.	Rot-Err.
$E_{\mathcal{D}}$	0.0027	0.13	0.0024	0.12
$E_{\mathcal{C}}$	0.0056	0.27	-	-
$E_{\mathcal{D}}$+$E_{\mathcal{C}}$	0.0017	0.09	-	-
$E_{\mathcal{L}}$	0.0018	0.10	0.0021	0.10
E	0.0014	0.07	0.0018	0.08

Table 3
The average translational/rotational errors of using different combinations of terms, in meters and degrees, respectively.

4.5 Evaluation of parameters

Our parameter settings are fixed for all test cases, including both real-world scenarios and simulated scans from diverse scenes. We further perform an experiment of adjusting the weight w_l for balancing between laser and RGBD costs used in Equ. 4. Specifically, we choose several candidates for σ_d (in different columns) and σ_b (in different rows), and rerun our system for testing. As the results of the mean and standard deviation of RMSE reveal in Tab. 4, our system is still able to maintain high-quality reconstruction even when the weight is changing within a certain range, where the parameters we use are considered as a good choice suitable for most situations.

	0.25	0.33	0.5	0.75
2.5	0.0487(0.0053)	0.0476(0.0052)	0.0476(0.0057)	0.0475(0.0044)
5.0	0.0484(0.0053)	0.0474(0.0058)	**0.0471**(0.0060)	0.0475(0.0063)
7.5	0.0477(0.0054)	0.0478(0.0052)	0.0476(0.0062)	0.0481(0.0061)
10.0	0.0476(0.0054)	0.0484(0.0055)	0.0482(0.0056)	0.0489(0.0055)

Table 4
Performance in terms of RMSE of different parameter configurations, where the corresponding standard deviation is shown in the bracket.

4.6 Pose evaluation quality

We further evaluate the performance of our classifier for pose evaluation (Sec. 3.3) when running on these simulated scans, and compare it with the previous system [20]. Specifically, we additionally attempt their classifier on our system, and use the deviation of the estimated transformations compared to the ground-truth as the criterion for statistics. As a consequence, both precision and recall of our classifier have significantly outperformed the one in [20], leading to a better judgement of these estimated poses in robotic scanning scenarios (see Tab. 5).

		InfiniTAM v3		Ours	
Scene	Attempts	Precision	Recall	Precision	Recall
SA	29428	0.8452	0.7783	0.9770	0.9905
SB	40133	0.8680	0.7894	0.9844	0.9865
SC	40736	0.8346	0.7625	0.9841	0.9883
SD	3641	0.8761	0.7333	0.9783	0.9885
SE	7568	0.8528	0.7440	0.9745	0.9906
SF	11894	0.8518	0.7289	0.9725	0.9905
SG	24591	0.8643	0.7581	0.9762	0.9857
Ave.	157991	0.8526	0.7672	0.9801	0.9881

Table 5
Statistics of pose evaluation classification precision and recall, averaged over all tracking attempts for all scenes.

4.7 Relocalization quality

We further compare the effectiveness of pose graph construction between InfiniTAM v3 [20] and ours, where Randomized Ferns [27] are used in both methods to detect loops, but our algorithm uses an additional constraint (Sec. 4.7) to filter them. Specifically for each tested trajectory, we traverse its final pose graph and compare all ultimate edge constraints to the ground truth relative transformations. Their average translation/rotation errors are summarized and listed in Tab. 6.

In Tab. 6, it is shown that the severity of erroneous edges can be diminished through our global position constraints. Such errors have positive correlations with final reconstruction errors, when tested with the same tracker for a fair comparison.

	InfiniTAM v3		Ours	
	Tran-Err.	Rot-Err.	Tran-Err.	Rot-Err.
SA	0.423	2.275	0.120	2.090
SB	0.628	4.097	0.069	1.539
SC	0.391	3.684	0.094	1.283
SD	0.071	1.266	0.064	1.418
SE	0.247	2.098	0.081	1.177
SF	1.214	9.379	0.103	1.724
SG	0.171	3.181	0.072	1.685

Table 6
The average translational/rotational errors of factors between submaps, in meters and degrees, respectively.

4.8 Memory usage and running times

We deployed our system on a desktop PC with an i7-6850K CPU (3.6GHz, 6 cores), 32GB RAM, and a single GeForce Titan Xp GPU (12GB, 3840 cores). In summary, we group the performance by different scenes and list average performance in Tab. 7. Each newly allocated submap takes a fixed 132MB for 3D and 8MB for 2D voxel blocks. The average running time is gradually increased and remains stable after a certain amount of submaps are created as discussed in [20]. Overall, statistics show that the running time meets the real-time requirement (less than 66ms for 15Hz synchronized messages).

Seq.	Frm.	Sub.	Mem.	Time	Seq.	Frm.	Sub.	Mem.	Time
RL1	1041	10	2.1	28.5	RL2	984	12	2.5	23.2
RO1	4307	40	7.0	33.0	RO2	1385	13	2.6	32.1
SA	1070	7	1.7	26.9	SB	1347	8	1.8	28.7
SC	1980	15	3.0	31.6	SD	341	5	1.4	24.1
SE	590	6	1.5	25.1	SF	957	6	1.5	25.3
SG	791	6	1.5	23.9					

Table 7
The average memory usage and computational time for each test scene. All timings are given in milliseconds and memory footprints in GB.

4.9 Limitations

Our system has two main limitations: (1) The *post* synchronization method used for associating RGBD and laser frames are based on their timestamps, which may be slightly different under an approximate time policy. We have not considered possible motion happened during this time interval when jointly registering multiple frames. Theoretically, the maximum time offset within such a batch of frames may reach 1/60 seconds (0.01 seconds on average as reported through our experiments). Hence, the motion during this time interval will influence the accuracy of pose estimation due to a slight difference of actual pose when these frames are generated. A general solution for the problem is to use a hardware synchronization device that simultaneously triggers the shutter/scan of each sensor. (2) The laser cost $E_{\mathcal{L}}$ is based on the Manhattan assumption [39] as most 2D LiDAR SLAM algorithms do: Idealistically, the laser scanner only captures points right on the horizontal plane at its height, but these points are actually distributed in a narrow slice due to both possible structural vibration during robot motion and uneven ground it drives on. For a major category of surfaces (walls, cabinets, and poles) that are orthogonal to the gravity direction within such a slice, strategies as illustrated in Fig. 4 can be used for constructing reasonable costs. Our method works robustly even if the scene contains some non-orthogonal points, as an overall cost function considering all the points is optimized. In order to test the influence of non-orthogonal surfaces, we take scenes with such surfaces and evaluate their impact. For each scene, we identify those non-orthogonal points on surfaces, count their proportion among all laser points in the scene, and test the performance of our pose estimation scheme both with and without these points taken into account in the cost function. As shown in Tab. 8, these scenes contain about 10% of non-orthogonal points, which is representative for real-world scenes, and these points have little effect on the performance of pose estimation. In practice, their proportion or influence can be further reduced through changing the height of the laser scanner or stabilizing the robot motion, respectively.

	E (with non-ortho)		E (without non-ortho)		
	Tran-Err.	Rot-Err.	Tran-Err.	Rot-Err.	non-orthogonal (%)
SA	0.0014	0.07	0.0013	0.07	12.53
SC	0.0021	0.08	0.0020	0.07	12.40
SE	0.0019	0.09	0.0018	0.09	8.57
SF	0.0017	0.09	0.0017	0.09	5.51

Table 8
The average translational/rotational errors with/without non-orthogonal points, in meters and degrees, respectively. Scenes without non-orthogonal surfaces (SB, SD, and SG) are not listed.

5 Conclusions

In this paper, we present a real-time system considering measurements from heterogeneous sensors for robot-based high-fidelity dense reconstruction. Planar laser frames are fused into a 2D TSDF volume for pose estimation in combination with the original RGBD scheme, and a classifier that considers information from multiple sensors, as well as the registration progress, is proposed for pose evaluation. In the future, our system can be combined with a smart motion planning system for automatically gathering real-world scenes, mass producing high-fidelity 3D scenes for understanding and manipulation tasks.

Acknowledgments

We would like to thank the reviewers for their constructive comments. This work was supported by the National Key Technology R&D Program (Project Number 2017YFB1002604), the Joint NSFC-DFG Research Program (Project Number 61761136018), and the Natural Science Foundation of China (Project Number 61521002).

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 *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, 2011, pp. 127–136.

[2] 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*, 2015, pp. 1:1–1:9.

[3] 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*, vol. 36, no. 3, pp. 24:1–24:18, 2017.

[4] Y.-P. Cao, L. Kobbelt, and S.-M. Hu, "Real-time high-accuracy three-dimensional reconstruction with consumer RGB-D cameras," *ACM Transactions on Graphics*, vol. 37, no. 5, pp. 171:1–171:16, 2018.

[5] B.-S. Hua, Q.-H. Pham, D. T. Nguyen, M.-K. Tran, L.-F. Yu, and S.-K. Yeung, "SceneNN: A scene meshes dataset with aNNotations," in *International Conference on 3D Vision (3DV)*, 2016, pp. 92–101.

[6] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "ScanNet: Richly-annotated 3D reconstructions of indoor scenes," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 5828–5839.

[7] Y. Zhang, W. Xu, Y. Tong, and K. Zhou, "Online structure analysis for real-time indoor scene reconstruction," *ACM Transactions on Graphics*, vol. 34, no. 5, pp. 159:1–159:13, 2015.

[8] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 77–85.

[9] S. Larsson and J. Kjellander, "Path planning for laser scanning with an industrial robot," *Robotics and Autonomous Systems*, vol. 56, no. 7, pp. 615–624, 2008.

[10] L. Liu, X. Xia, H. Sun, Q. Shen, J. Xu, B. Chen, H. Huang, and K. Xu, "Object-aware guidance for autonomous scene reconstruction," *ACM Transactions on Graphics*, vol. 37, no. 4, pp. 104:1–104:12, 2018.

[11] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1309–1332, 2016.

[12] G. Grisetti, C. Stachniss, and W. Burgard, "Improved techniques for grid mapping with rao-blackwellized particle filters," *IEEE Transactions on Robotics*, vol. 23, no. 1, pp. 34–46, 2007.

[13] W. Hess, D. Kohler, H. Rapp, and D. Andor, "Real-time loop closure in 2D LiDAR SLAM," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 1271–1278.

[14] H. Deilamsalehy, T. C. Havens, and J. Manela, "Heterogeneous multisensor fusion for mobile platform three-dimensional pose estimation," *Journal of Dynamic Systems, Measurement, and Control*, vol. 139, no. 7, p. 071002, 2017.

[15] J. Zhang and S. Singh, "Visual-LiDAR odometry and mapping: Low-drift, robust, and fast," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 2174–2181.

[16] Y. Balazadegan Sarvrood, S. Hosseinyalamdary, and Y. Gao, "Visual-LiDAR odometry aided by reduced IMU," *ISPRS International Journal of Geo-Information*, vol. 5, no. 1, p. 3, 2016.

[17] C. Wen, L. Qin, Q. Zhu, C. Wang, and J. Li, "Three-dimensional indoor mobile mapping with fusion of two-dimensional laser scanner and RGB-D camera data," *IEEE Geoscience and Remote Sensing Letters*, vol. 11, no. 4, pp. 843–847, 2014.

[18] X. He, D. N. Aloi, and J. Li, "Probabilistic multi-sensor fusion based indoor positioning system on a mobile device," *Sensors*, vol. 15, no. 12, pp. 31 464–31 481, 2015.

[19] S. J. Julier and J. K. Uhlmann, "New extension of the Kalman filter to nonlinear systems," in *Signal Processing, Sensor Fusion, and Target Recognition VI*, vol. 3068, 1997, pp. 182–194.

[20] O. Kähler, V. A. Prisacariu, and D. W. Murray, "Real-time large-scale dense 3D reconstruction with loop closure," in *European Conference on Computer Vision (ECCV)*, 2016, pp. 500–516.

[21] T. Whelan, M. Kaess, M. Fallon, H. Johannsson, J. Leonard, and J. McDonald, "Kintinuous: Spatially extended KinectFusion," in *Robotics: Science and System Workshop on RGB-D: Advanced Reasoning with Depth Cameras*, 2012.

[22] M. Nießner, M. Zollhöfer, S. Izadi, and M. Stamminger, "Real-time 3D reconstruction at scale using voxel hashing," *ACM Transactions on Graphics*, vol. 32, no. 6, pp. 169:1–169:11, 2013.

[23] C. Kerl, J. Sturm, and D. Cremers, "Robust odometry estimation for RGB-D cameras," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2013, pp. 3748–3754.

[24] T. Whelan, M. Kaess, H. Johannsson, M. Fallon, J. J. Leonard, and J. Mcdonald, "Real-time large-scale dense RGB-D SLAM with volumetric fusion," *International Journal of Robotics Research (IJRR)*, vol. 34, no. 4–5, pp. 598–626, 2015.

[25] 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 (TVCG)*, vol. 21, no. 11, pp. 1241–1250, 2015.

[26] D. Galvez-Lopez and J. D. Tardos, "Real-time loop detection with bags of binary words," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2011, pp. 51–58.

[27] B. Glocker, J. Shotton, A. Criminisi, and S. Izadi, "Real-time RGB-D camera relocalization via randomized ferns for keyframe encoding," *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, vol. 21, no. 5, pp. 571–583, 2015.

[28] G. Grisetti, R. Kummerle, C. Stachniss, and W. Burgard, "A tutorial on graph-based SLAM," *IEEE Intelligent Transportation Systems Magazine*, vol. 2, no. 4, pp. 31–43, 2010.

[29] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "G2O: A general framework for graph optimization," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2011, pp. 3607–3613.

[30] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM: a versatile and accurate monocular SLAM system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[31] V. Kubelka, L. Oswald, F. Pomerleau, F. Colas, T. Svoboda, and M. Reinstein, "Robust data fusion of multimodal sensory information for mobile robots," *Journal of Field Robotics*, vol. 32, no. 4, pp. 447–473, 2015.

[32] M. Bloesch, S. Omari, M. Hutter, and R. Siegwart, "Robust visual inertial odometry using a direct EKF-based approach," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 298–304.

[33] J. C. Chow, D. D. Lichti, J. D. Hol, G. Bellusci, and H. Luinge, "IMU and multiple RGB-D camera fusion for assisting indoor stop-and-go 3D terrestrial laser scanning," *Robotics*, vol. 3, no. 3, pp. 247–280, 2014.

[34] T. Qin, P. Li, and S. Shen, "VINS-Mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, 2018.

[35] E. Mendes, P. Koch, and S. Lacroix, "ICP-based pose-graph SLAM," in *IEEE International Symposium on Safety, Security, and Rescue Robotics*, 2016, pp. 195–200.

[36] M. Chen, S. Yang, X. Yi, and D. Wu, "Real-time 3D mapping using a 2D laser scanner and IMU-aided visual SLAM," in *IEEE International Conference on Real-Time Computing and Robotics*, 2017, pp. 297–302.

[37] M. Bosse, R. Zlot, and P. Flick, "Zebedee: Design of a spring-mounted 3D range sensor with application to mobile mapping," *IEEE Transactions on Robotics*, vol. 28, no. 5, pp. 1104–1119, 2012.

[38] "ROS message filter: Approximate time synchronization strategy," http://wiki.ros.org/message_filters/ApproximateTime.

[39] Y. Furukawa, B. Curless, S. M. Seitz, and R. Szeliski, "Manhattan-world stereo," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009, pp. 1422–1429.

[40] R. Smith, M. Self, and P. Cheeseman, "Estimating uncertain spatial relationships in robotics," in *Autonomous Robot Vehicles*, 1990, pp. 167–193.

[41] S. Huang and G. Dissanayake, "Convergence and consistency analysis for extended kalman filter based SLAM," *IEEE Transactions on Robotics*, vol. 23, no. 5, pp. 1036–1049, 2007.

[42] A. Vedaldi and A. Zisserman, "Efficient additive kernels via explicit feature maps," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 34, no. 3, pp. 480–492, 2012.

[43] A. Handa, V. Patraucean, V. Badrinarayanan, S. Stent, and R. Cipolla, "Understanding real world indoor scenes with synthetic data," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 4077–4085.

[44] A. Handa, T. Whelan, J. McDonald, and A. J. Davison, "A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 1524–1531.

[45] A. Kassir and T. Peynot, "Reliable automatic camera-laser calibration," in *Australasian Conference on Robotics and Automation*, 2010, pp. 137:1–137:10.

Sheng Yang received his B.S. degree in computer science from Wuhan University in 2014. He is currently a Ph.D. candidate in computer science in Tsinghua University. His research interests include computer graphics, geometric modeling and processing.

Beichen Li Beichen Li received his B.S. degree in computer science and technology from Tsinghua University in 2018. He then joined the Computational Fabrication Group in Massachusetts Institute of Technology as a Ph.D. candidate. His major research interests include computer graphics, 3D geometry and computational design.

Minghua Liu is an undergraduate student at Tsinghua University. His research interests include computer graphics and machine learning.

Yu-Kun Lai Yu-Kun Lai received his bachelor's and Ph.D. degrees in computer science from Tsinghua University in 2003 and 2008, respectively. He is currently a Reader in the School of Computer Science & Informatics, Cardiff University, UK. His research interests include computer graphics, geometry processing, image processing and computer vision. He is on the editorial board of The Visual Computer.

Leif Kobbelt is a distinguished professor at RWTH Aachen University in Germany and head of the Computer Graphics Group. His research interests cover many areas in Computer Graphics and Geometry Processing with a focus on the generation, optimization and interactive modification of complex 3D models. He received his masters degree in 1992 and Ph.D. in 1994 from the University of Karlsruhe, Germany and published a substantial number of papers in international top-conferences and journals. For his research he was awarded with the Eurographics Outstanding Technical Contribution Award 2004, two Gunter Enderle Awards (1999 and 2012), ERC Advanced Grant 2013, and the Gottfried Wilhelm Leibniz Prize in 2014. He is a member of the Academia Europaea and the North Rhine-Westphalian Academy of Sciences.

Shi-Min Hu is currently a professor in the department of Computer Science and Technology, Tsinghua University, Beijing. He received the PhD degree from Zhejiang University in 1996. His research interests include digital geometry processing, video processing, rendering, computer animation, and computer-aided geometric design. He has published more than 100 papers in journals and refereed conferences. He is Editor-in-Chief of Computational Visual Media (Springer), and on editorial board of several journals, including Computer Aided Design (Elsevier) and Computers & Graphics (Elsevier). He is the corresponding author of the paper.

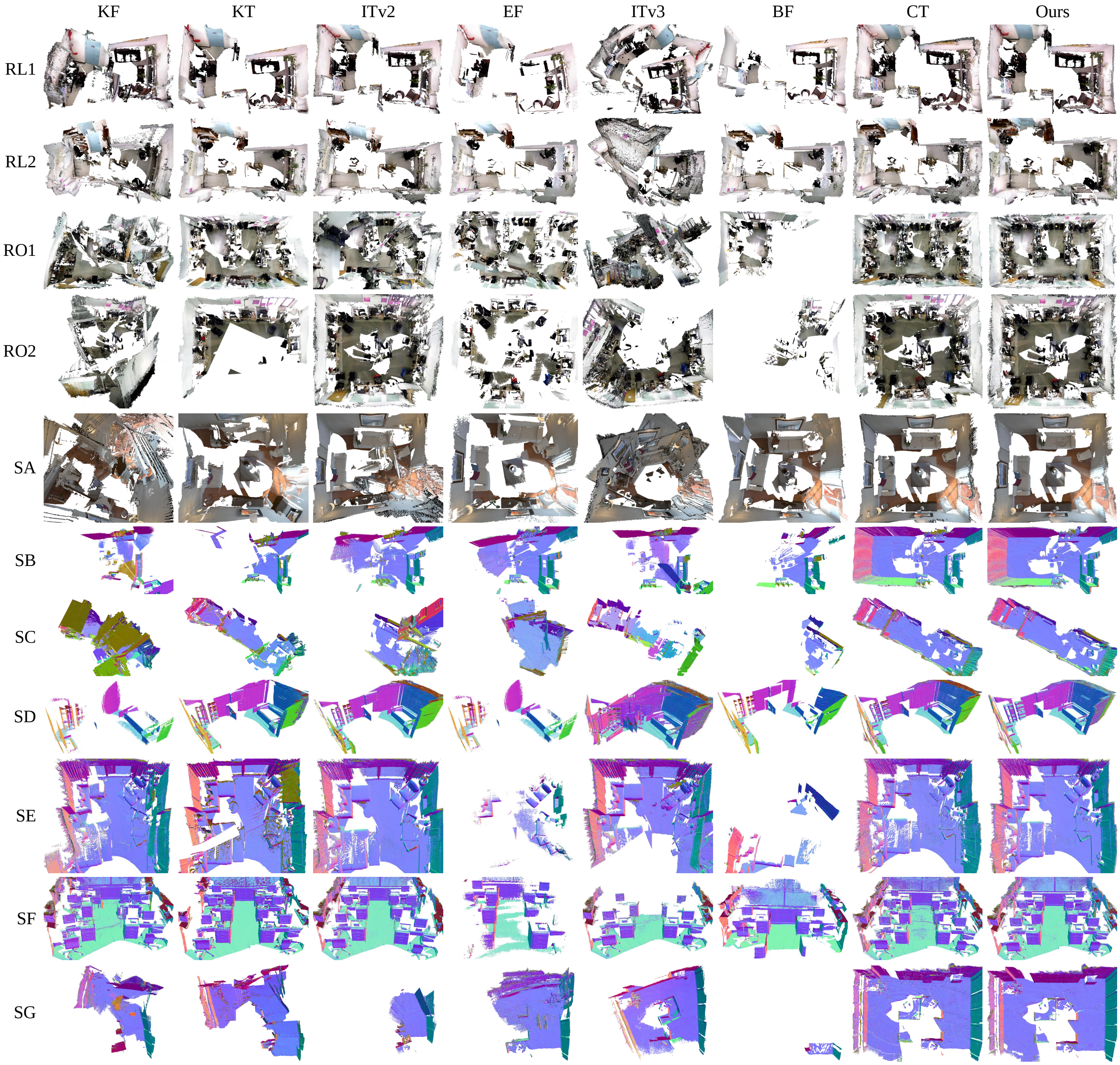


Figure 9. Typical examples of reconstruction results in global top-down views. Color-coding is used for synthetic scenes to visualize surface normals. Both CT and ours can maintain the global consistency of reconstruction. Please refer to our supplementary document for close-up views of these results.

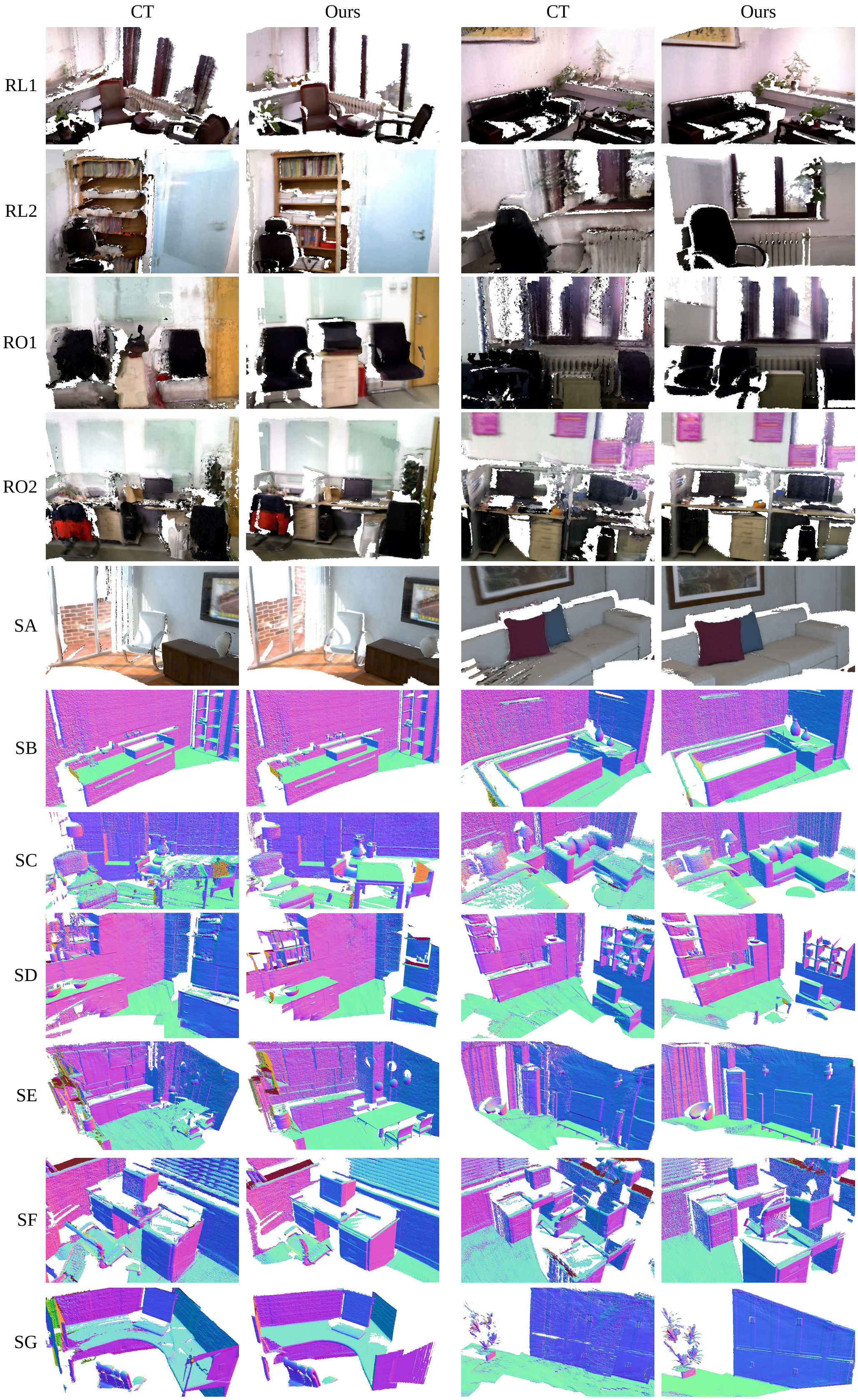


Figure 10. Typical examples of reconstruction results in local views. Color-coding is used for synthetic scenes to visualize surface normals. CT is based on 2D laser scans and thus not sufficient for precisely estimating 3D poses. Their reconstruction results are worse than ours in these local views. Please refer to our supplementary document and video for a more detailed/dynamic comparison of these results.