Proceeding of the IEEE
International Conference on Robotics and Biomimetics
Dali, China, December 2019

2D Range Flow-based Odometry fusing LiDAR and IMU

Guo Ziwei
Robotics Institute
Beihang University
Beijing, China
Sy1707621@buaa.edu.cn

Liu Rong
Robotics Institute
Beihang University
Beijing, China
rliu@buaa.edu.cn

***Abstract* - This paper describes a 2D odometry method fusing LiDAR and IMU data to estimate ground vehicle motion and performing better in long corridor scenarios and moving objecting scenarios. The method builds on range flow constraint for every scanned point and rejects moving objects by fusing IMU data. Besides, our method applies the graphical model to solve the optimization problem that adds IMU constraint to the odometry system. Experiment results demonstrate that our approach is able to perform as well as other state-of-art approaches and better than these approaches in long corridor scenarios and moving objecting scenarios. Besides we test our approach on an embedded system which makes it suitable for those low-cost robotic applications that require planar odometry.**

Index Terms – Lidar Odometry, mobile robots.

I. INTRODUCTION

Odometry is a fundamental building block for robot localization and navigation and real-time methods for odometry have made significant progress in recent years. Mobile robots and autonomous vehicle typically use different techniques for pose estimation that are major based on integrated GPS/IMU systems, wheel encoders, laser scanners or visual odometry (VO). Light detection and ranging (LiDAR) are important for obstacle detection and localization while VO methods are flexible and good at loop-detection, but their sensitivity to illumination and light changes makes it hard to be used as solo navigation sensor

Our approach focuses on ground vehicle odometry using a 2D LiDAR built upon [1]. LiDAR-based methods have the advantages in independent of the vehicle type of locomotion and precise, as supported by experimental validation. Thus, it is suitable for the robot moving in planar equipped a 2D LiDAR. However, LiDAR-based approaches cannot handle the low geometry texture scenarios like moving in a long corridor and dynamic environment. In such situations, an IMU can help a lot due to its independent of the external environment. In order to solve the problems, our approach applies IMU to estimate ego-motion where moving objects and low geometry texture affect motion estimation.

The main contribution of the paper is an approach fusing 2D LiDAR data and IMU data to handle low geometry texture and dynamic environment and exploit graphical model optimization a range flow-based localization system for autonomous ground vehicles equipped with LiDAR and IMU sensors. Furthermore, we test our approach both on a dataset [2] and a ground vehicle in real-world experiments. All results are obtained in real-time and validate the superior performance of the proposed algorithm. The paper is organized as follows: we present in detail the range flow constraint equation in section III and demonstrate how to fuse IMU measurement and how odometry works in section IV. In section V we give experiment both on dataset and real robot. In section VI, we conclude the work in this paper.

II. RELATED WORKS

The most popular approaches used in LiDAR are the iterative closest point (ICP) algorithm [3], [4] and its variants. The ICP algorithm aims to find the transformation between a point cloud and some reference surface (or another point cloud), by minimizing the squared errors between the corresponding entities. And in the field of mobile robot, it has been extensively employed to match 2D laser scans called scan match. For registering two-point clouds recorded at unknown relative positions, the ICP algorithm iteratively performs two steps: data association and transformation estimation given the data association. The point-to-point distance and the point-to-plane distance are two commonly used metrics. Furthermore, a very successful approach was proposed by Censi [5] using a point-to-line metric rather than point-to-point original metric of ICP and this method is commonly used in Robot Operation System (ROS) as 2D LiDAR odometry. Gonzalez&Gutierrez [6] formulated the "velocity constraint equation" an adaptation of the optical flow constraint for scans. However, this method was only tested with simple simulated scenarios and provided modest results. The Generalized-ICP [7] combines the Iterative Closest Point and point-to-plane ICP algorithm into a single probabilistic framework and uses the framework to model locally planar surface structure from both scans instead.

More recently, other approaches combine LiDAR with other sensors to improve the accuracy of the estimation and to compensate for individual sensor deficiencies such as [8], [9]. [8] fuses LiDAR odometry with stereo vision and IMU with extended Kalmen filter and [9] uses monocular estimate robot pose without scale and applies the ICP approach to estimate 1-DoF scale. But approaches above do not focus on the situations that robot in sparse geometry texture and the dynamic environment. Our proposed method applies MEMS IMU which is cheap and often equipped by ground vehicles to work in such scenarios.

978-1-7281-6321-5/19//$31.00 ©2019 IEEE

III. 2D RANGE FLOW CONSTRAINT IN RIGID ENVIRONMENT

2D range flow constraint was firstly introduced by Gonzalez&Gutierrez [6] and subsequently generalized and named as the "range flow constraint equation" in [10]. It estimates the 2D velocity of a LiDAR from the motion that it observes. Assuming that the environment is static and rigid, let $v\left(v_x, v_y, \omega\right)$ be the velocity of a 2D LiDAR moving on a surface and $R\left(t,\theta\right)$ be a range measured for a generic point P at its polar coordinate $\left(r,\theta\right)$ (in Fig.1) and time t. The velocity of point p v_{p} is.

$$v_{\mathrm{p}} = -\omega r \begin{bmatrix} -sin\theta \\ cos\theta \end{bmatrix} - \begin{bmatrix} v_x \\ v_y \end{bmatrix}. \tag{1}$$

On the other hand, the velocity of P can be calculated by derivate:

$$\begin{aligned} v_{\mathrm{p}} &= \frac{\Delta_p}{\Delta_t} = \begin{bmatrix} \dot{r}\cos\theta - r\sin\theta\dot{\theta} \\ \dot{r}\sin\theta + r\cos\theta\dot{\theta} \end{bmatrix} \\ &= \begin{bmatrix} \cos\theta & -r\sin\theta \\ \sin\theta & r\cos\theta \end{bmatrix} \begin{bmatrix} \dot{r} \\ \dot{\theta} \end{bmatrix} \end{aligned}. \tag{2}$$

With Δ_t is the time gap between consecutive scans. We can get an equation from (1) and (2):

$$\begin{bmatrix} \cos\theta & -r\sin\theta \\ \sin\theta & r\cos\theta \end{bmatrix} \begin{bmatrix} \dot{r} \\ \dot{\theta} \end{bmatrix} + \omega r \begin{bmatrix} -sin\theta \\ cos\theta \end{bmatrix} + \begin{bmatrix} v_x \\ v_y \end{bmatrix} = 0. \tag{3}$$

Assuming the differentiability of $R\left(t,\theta\right)$, the range of any point at the second scan can be expressed as the Taylor expansion.

$$\begin{aligned} \mathrm{R}(t+\Delta_t, \theta+\Delta_\theta) &= R\left(t,\theta\right) + \frac{\partial R}{\partial t}\left(t,\theta\right)\Delta_t \\ &+ \frac{\partial R}{\partial \theta}\left(t,\theta\right)\Delta_\theta + O(\Delta_t{}^2, \Delta_\theta{}^2) \end{aligned}. \tag{4}$$

With

$$\begin{aligned} R_t &= \frac{\partial R}{\partial t}\left(t,\theta\right) \\ R_\theta &= \frac{\partial R}{\partial \theta}\left(t,\theta\right) \end{aligned}.$$

Dividing (4) by Δt and neglecting the second/higher-order terms:

$$\dot{r} = \frac{\Delta_R}{\Delta_t} \simeq R_t + R_\theta \dot{\theta}. \tag{5}$$

Equation (5) is the range flow constraint equation introduced by Gonzalez&Gutierrez [11]. It states a constraint over possible velocities that a point from the scene can reach once the local structure of the environment and the temporal rate of change $\frac{\Delta_R}{\Delta_t}$ have been estimated for a given range point $R\left(t,\theta\right)$.

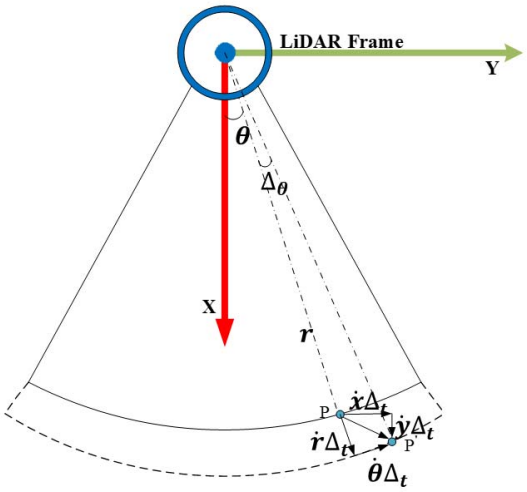

Fig. 1 LiDAR frame.

IV. LIDAR ODOMETRY

A. Velocity Estimation

We formulate odometry as a nonlinear optimization problem. First, we obtain the following equation according to (3) and (5):

$$\begin{bmatrix} \cos\theta & -r\sin\theta \\ \sin\theta & r\cos\theta \end{bmatrix} \begin{bmatrix} \dot{r} \\ \dot{\theta} \end{bmatrix} + \omega r \begin{bmatrix} -\sin\theta \\ \cos\theta \end{bmatrix} + \begin{bmatrix} v_x \\ v_y \end{bmatrix} = 0. \tag{6}$$

Equation (6) has two degrees of freedom. Each point in scan provides two additional constraints and only one additional unknown (its angular velocity) so a minimum of three scanned points will suffice to solve for the six unknowns $\left(\dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3, v_x, v_y, \omega\right)$.

But in practice, (6) cannot equal to 0 due to the linear approximation, measurement error or moving objects in the environment. Therefore, we define the residual $\rho\left(\xi\right)$ as the evaluation of the range flow constraint (7) for a given velocity v.

$$\begin{aligned} \rho\left(\xi\right) &= \begin{bmatrix} \cos\theta & -r\sin\theta \\ \sin\theta & r\cos\theta \end{bmatrix} \begin{bmatrix} \dot{r} \\ \dot{\theta} \end{bmatrix} \\ &+ \omega r \begin{bmatrix} -\sin\theta \\ \cos\theta \end{bmatrix} + \begin{bmatrix} v_x \\ v_y \end{bmatrix} \end{aligned}. \tag{7}$$

To obtain an accurate estimate, we use the Cauchy M-estimator to minimize all residuals of all the points of a scan.

The optimization problem is solved with Iteratively Reweighted Least Squares (IRLS).

B. Reject Moving Objects

As mentioned before, the rigidity hypothesis can render (3) inaccurate. When an object is moving between two scans, the velocity of the points that hit on the moving object causes the error of robot velocity estimation. To handle this, we get angular rate and integrate acceleration from IMU between consecutive scan with respect to an inertial frame to obtain predicted velocity v_{guess}. First, we calculate every point's velocity and compare with v_{guess}, if the deviation is beyond a threshold, we treat it as an outlier and drop it. Ideally, the velocity of the points hit on the static is close to the robot velocity and the velocity of other points that hit on the moving object is the sum or minus of the two velocities. To calculate v_{guess} from IMU measurements, namely $\omega_g^b(t)$ as angular rate expressed in the body frame and $a^w(t)$ linear acceleration expressed in world frame, both are affected by additive white noise $\mathbf{n}$ and a slowly varying sensor bias $\mathbf{b}$ and true values denoted by $\widetilde{\omega}_g^b(t)$, $\widetilde{a^b}(t)$.

$$\widetilde{\omega}_g^b(t) = \omega_g^b(t) + b_g(t) + n_g(t). \tag{8}$$

$$\widetilde{a^b}(t) = q_{b_t w}\left(a^w(t) + g^w\right) + b_a(t) + n_a(t). \tag{9}$$

Where $q_{b_t w}$ denotes robot frame to world frame rotation and g^w denotes gravity in world frame.

The goal now is to infer the linear velocity of the robot (the IMU frame) so we integrate equation (8), (9) between timestamp i and timestamp j:

$$v_j^w = v_i^w + \int_{t\in[i,j]}\left(q_{wb_t}a^{b_t} - g^w\right)\delta t. \tag{10}$$

For discrete-time implementation, different numerical integration methods such as Euler, mid-point, RK4 integration can be applied. Here we use mid-point between k measurement and k+1 measurement:

$$v_{k+1}^w = v_k^w + \frac{1}{2}\Delta_t \left[q_{wb_k}\left(a^{b_k} - b_k^a\right) - g^w + q_{wb_{k+1}}\left(a^{b_{k+1}} - b_k^a\right) - g^w\right]. \tag{11}$$

So, the v_{guess} of the k+1 scan is $v_{guess}(v_{k+1}^w, \omega_g^b)$. Every time we estimate robot velocity v from (7) after rejecting the outlier, update v_{guess} by v.

C. Local Optimization

Now we have two odometry message streams from IMU and scan match and we want to use both information. The solutions to fuse those sensor messages are divided into two main categories: filter based and smoothing. The filter category itself can be subdivided into Extended Kalman Filter (EKF), information filter and particle filters. Recently there has been a lot of researches on solving SLAM through smoothing, where the solution is computed using sparse optimization techniques. By exploring the sparsity of the SLAM problem, it is possible to considerably reduce the computation time needed to converge to a solution. Also, the graphical models used to model the optimization problem provide a powerful layer of abstraction that helps to better understand the problem and to design powerful solutions, like the iSAM2 algorithm [12]. The objective of the graph-based optimization is to find a configuration for the variables that best matches the measurements encoded in the factors. If the observations are affected by Gaussian noise, this is the same as to compute a configuration that maximizes the likelihood of the observations. Generically, each factor f_i encodes an error function that computes the difference between the expected observation and the acquired observation. For measurements in Euclidean spaces, this function has the form:

$$e_{f_y} = o_i - h(\mathcal{X}_i). \tag{12}$$

Where we use o_i to denote the measurement encoded in f_i and $h(\mathcal{X}_i)$ is the observation model of the IMU. Finally, we model our system as Fig. 2.

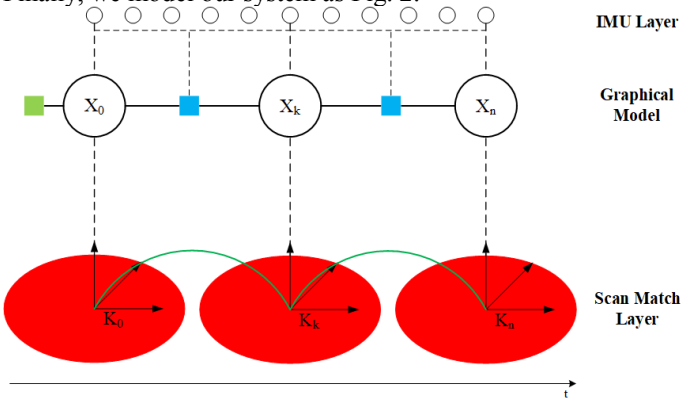

Fig. 2 graph model.
Small circles represent IMU measurements, the green square represents prior information,
the blue squares represent odometry constraints from IMU integration
and the big circles denote robot poses
at consecutive time steps (x1, x2, . . .) from LiDAR odometry.

V. EXPERIMENTS

This section is composed of three different experiments. The first experiment addresses the evaluation of the proposed LiDAR odometry and its comparison with other approaches in MIT dataset [2]. The second and the third are to evaluate the robustness in low geometry texture scenarios where the LiDAR moving in a corridor and the dynamic environment. For comparison, we choose two state-of-the-art scan match algorithms: Point-to-Line ICP (PL-ICP) [5] and RF2O [1]. In

both cases, we use the implementations that their authors published online.

For real experiments, a SLAMTEC rplidar A2 2D LiDAR at a frequency of 10 Hz and a MEMS IMU at a frequency of 200 Hz mounted on a ground vehicle made by ourselves. We use an evaluation tool [13] to get absolute pose error [14] for quantitative evaluation on each experiment.

A. Comparison in MIT dataset experiment

The dataset provides the LiDAR data from a Hokuyo UTM-30LX Laser in 40 Hz, IMU data from 3DM-GX2 in 100Hz and wheel odometry data. The environment is a hall in the Ray and Maria Stata Center and the odometry estimated by different algorithms on a PC with i5-3470 3.20 GHz CPU and 8GB RAM. The result of each algorithm is plotted in Fig.3 and we also plot the wheel odometry to compare. The error results of each algorithm are in Table Ⅰ (using LI-odom represent our approach temporarily). Table Ⅰ illustrates that our algorithm performs better than wheel odom and RF2O while it has a similar performance with PL-ICP in usual indoor scenarios.

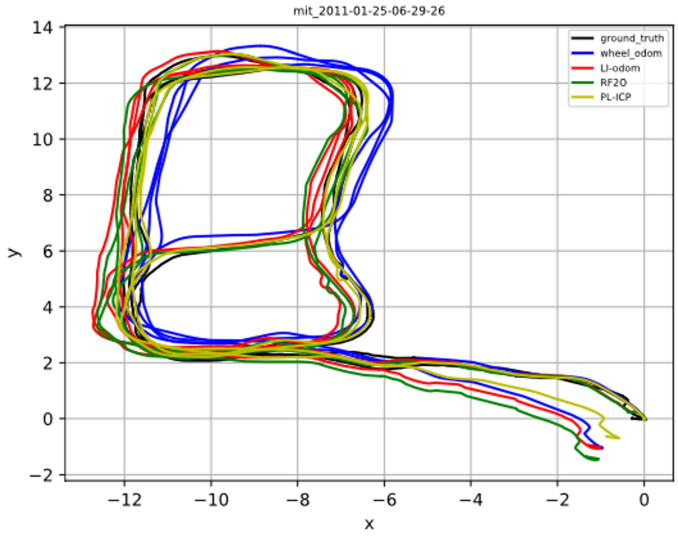

Fig. 3 Trajectory of MIT dataset,
the black line represents ground truth,
the blue line represents wheel odom,
the red line represents our method,
the green line represents RF2O and the yellow one is PL-ICP.

TABLE I

QUANTITATIVE ERRORS IN MIT DATASET

evaluation	Algorithms			
	PL-ICP	***RF2O***	***LI-odom***	***Wheel odom***
APE-RMSE(m)	0.145	0.357	0.209	0.340

B. Real Experiment

In real experiments, we collect all data by our robot then test our algorithm and the compared algorithms on ODROID XU4, an embedded PC containing a hybrid processing unit with a quad-core ARM A7 at 1.5 GHz and a quad-core ARM A15 at 2.0 GHz

The second experiment aims to evaluate the precise in a dynamic environment, we collected the data in a corridor where there are two doors (see Fig.3). During the experiment, the robot stays still while people pass by and two doors open and close. Table Ⅱ shows the quantitative result to evaluate the performance in dynamic environment.

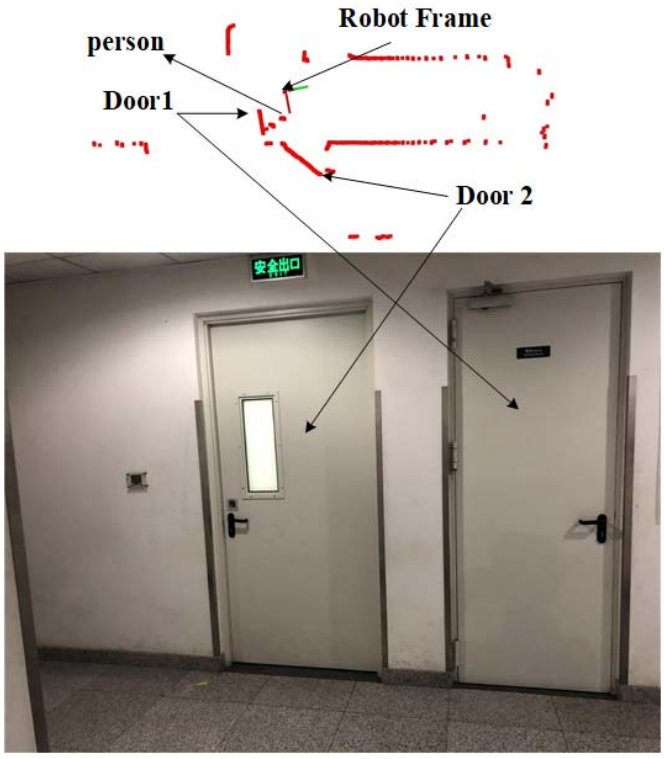

Fig. 4 Dynamic environment

TABLE II

QUANTITATIVE ERRORS IN MIT DATASET

evaluation	Algorithms		
	PL-ICP	***RF2O***	***LI-odom***
APE-RMSE(m)	0.0106	0.0375	0.0099

The third experiment aims to evaluate the robustness and precise in a long corridor with sparse geometry texture (see Fig.4), the robot collects data back and forth. Due to the lack of ground truth, we set the robot back to the starting line to evaluate the precise. The result is shown in Fig.5, the PL-ICP and RF2O cannot work at all because ICP solver will provide an arbitrary solution when the lidar moves parallel to the wall but our approach can still work in such scenario.

Fig. 5 Long corridor experiment,
the dash lines demonstrate the direction of the robot

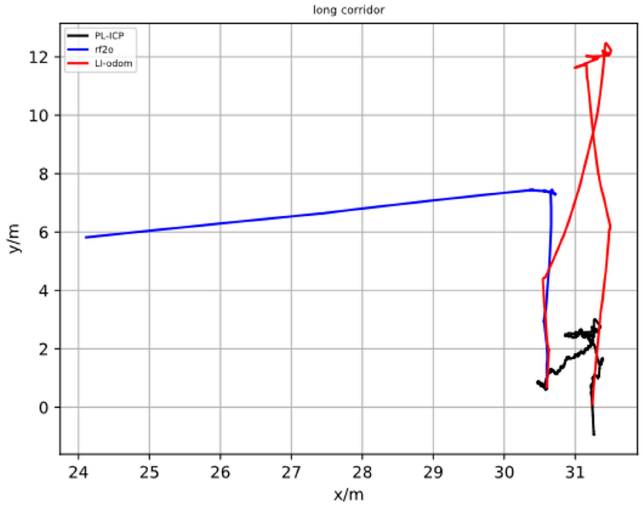

Fig. 6 Trajectory in a long corridor

VI. CONCLUSION

We present a new approach to estimate ground vehicle odometry fusing 2D LiDAR and IMU messages and test it both in MIT dataset and real-world comparing with other LiDAR -based odometry algorithms. The results demonstrate that our approach performs well even if in dynamic or sparse geometry texture environment.

REFERENCES

[1] M. Jaimez, J. G. Monroy, and J. Gonzalez-Jimenez, "Planar odometry from a radial laser scanner. A range flow-based approach," in 2016 IEEE International Conference on Robotics and Automation (ICRA), 2016, pp. 4479-4485.

[2] M. Fallon, H. Johannsson, M. Kaess, and J. J. Leonard, "The MIT Stata Center dataset " Int. J. Rob. Res., vol. 32, no. 14, pp. 1695-1699, 2013.

[3] P. J. Besl and N. D. McKay, "A method for registration of 3-D shapes," Transactions on Pattern Analysis and Machine Intelligence, vol. 14, no. 2, pp. 239-256, 1992.

[4] Y. Chen and G. Medioni, "Object modeling by registration of multiple range images," in Proceedings. 1991 IEEE International Conference on Robotics and Automation, 1991, pp. 2724-2729 vol.3.

[5] A. Censi, "An ICP variant using a point-to-line metric," in 2008 IEEE International Conference on Robotics and Automation, 2008, pp. 19-25.

[6] J. Gonzalez and R. Gutierrez, "Mobile robot motion estimation from a range scan sequence," in Proceedings of International Conference on Robotics and Automation, 1997, vol. 2, pp. 1034-1039 vol.2.

[7] A. Segal, D. Hähnel, and S. Thrun, Generalized-ICP. 2009.

[8] H. Deilamsalehy and T. C. Havens, "Sensor fused three-dimensional localization using IMU, camera and LiDAR," in 2016 IEEE SENSORS, 2016, pp. 1-3.

[9] K. Huang and C. Stachniss, Joint Ego-motion Estimation Using a Laser Scanner and a Monocular Camera Through Relative Orientation Estimation and 1-DoF ICP. 2018, pp. 671-677.

[10] H. Spies, B. Jähne, and J. L. Barron, "Range Flow Estimation," Computer Vision and Image Understanding, vol. 85, no. 3, pp. 209-231, 2002/03/01/ 2002.

[11] J. Gonzalez and R. Gutierrez, "Direct motion estimation from a range scan sequence," Journal of Robotic Systems, vol. 16, no. 2, pp. 73-80, 1999.

[12] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, and F. Dellaert, "iSAM2: Incremental smoothing and mapping using the Bayes tree," The International Journal of Robotics Research, vol. 31, no. 2, pp. 216-235, 2012/02/01 2011.

[13] M. Grupp. (2017). evo: Python package for the evaluation of odometry and SLAM. Available: https://github.com/MichaelGrupp/evo.

[14] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2012, pp. 573-580.