

# DLO: Direct LiDAR Odometry for 2.5D Outdoor Environment

Lu Sun, Junqiao Zhao\*, Xudong He, Chen Ye

**Abstract**—For autonomous vehicles, high-precision real-time localization is the guarantee of stable driving. Compared with the visual odometry (VO), the LiDAR odometry (LO) has the advantages of higher accuracy and better stability. However, 2D LO is only suitable for the indoor environment, and 3D LO has less efficiency in general. Both are not suitable for the online localization of an autonomous vehicle in an outdoor driving environment. In this paper, a direct LO method based on the 2.5D grid map is proposed. The fast semi-dense direct method proposed for VO is employed to register two 2.5D maps. Experiments show that this method is superior to both the 3D-NDT and LOAM in the outdoor environment.

## I. INTRODUCTION

In the absence of GNSS, an autonomous vehicle must have the ability to locate itself through the sensors' perception of the environment. The Simultaneous Localization and Mapping (SLAM) algorithm constructs a map by observing environmental landmarks during driving, and correlates and matches the current perception to the map in real time to optimally estimate the ego-pose.

In a SLAM system, one of the most important parts is the front-end, which usually performs the frame-by-frame pose estimation, resulting in an odometry sub-system. Visual odometry (VO) and LiDAR odometry (LO) both have been intensively explored. The VO methods can be classified into indirect and direct method [1]. The indirect methods rely on features extracted from images, such as the ORB feature adopted in ORB-SLAM [2]. The transformation between consequent keyframes is then calculated by the matching of features. The direct methods, such as LSD-SLAM [3], directly estimate transformation between keyframes based on the minimization of the photometric error between two images, thus are faster and more robust than indirect methods in the "texture-less" scene which contains little features. The combination of both merits is also proposed by the SVO [4]. However, as the camera is easily affected by the changing illumination, VO methods still suffer from robust issues, especially in the outdoor environment.

Thanks to the active sensing and the high precision of LiDAR sensor, the LO methods are more robust and accurate than VO in general. The existing LO methods can be classified based on the spatial dimensionality of the measurement involved. 2D LO methods use a two-dimensional occupancy

grid map (OGM) with branch-and-bound [5] or gradient descending scan-match method to estimate the sensor's motion. Practical systems based on these methods are GMapping [6], Hector SLAM [7], and cartographer [8]. However, due to the insufficient representative power of the 2D OGM for outdoor environments, these methods are most suitable for the indoor environment.

The 3D LO methods employ point clouds from the multi-beam LiDAR to estimate the motion of the sensor by matching of point clouds [9] or 3D OGMs [10], or by matching of 3D features extracted from point clouds [11]. Because of the accessible 3D information, the 3D LO method can be applied to both indoor and outdoor environments. Nevertheless, the expensive computation required for point cloud registration or OGM matching in 3D space makes these methods hardly fit the online localization.

The key insight of this paper is that the height variation in the surrounding of a vehicle (represented by a 2.5D grid map or a height map) is discriminative and lightweight, compared to the above two popular representations. Although 2.5D-based localization methods already exist [12], they cannot be used as an odometry system since a pre-built map is mandatory.

This paper presents an LO method based on the 2.5D grid map for autonomous vehicles in the outdoor environment. The main contribution is the usage of the semi-dense direct method originated from VO to realize a fast and accurate registration of 2.5D representations of LiDAR scans.

Firstly, a 2.5D grid map with each cell retaining a height expectation is built. Then, cells with high gradient are selected, and Gauss-Newton method is used to optimize the objective function based on the height difference error (HDE) of two 2.5D grid maps. Finally, the optimal transition matrix of two maps is obtained. The experimental results showed the proposed method can register two frames of HDL-64 LiDAR in centimeter accuracy in 40 ms, comparing to 39s of 3D-NDT and 0.93s of LOAM (without IMU).

## II. RELATED WORK

Existing LO or LiDAR SLAM methods can be divided into 2D, 3D, and 2.5D, according to their map representation. Since the 2D-based methods could not be applied in many open outdoor environments, where merely 2D occupancy information is not sufficient, this section will focus on the 3D and the 2.5D based methods.

The classic ICP algorithm [9], initially proposed by Besl and Mckey, is a point set-based registration method. The registration process is to iteratively find the optimal transition matrix between two point clouds based on the point-wise

This work is supported by the National Natural Science Foundation of China (No. U1764261), the Natural Science Foundation of Shanghai (No. kz170020173571) and the Fundamental Research Funds for the Central Universities (No. 22120170232, No. 22120180095).

All authors are with the Department of Computer Science and Technology, School of Electronics and Information Engineering, Tongji University, Shanghai, 201804, China zhaojunqiao@tongji.edu.cn

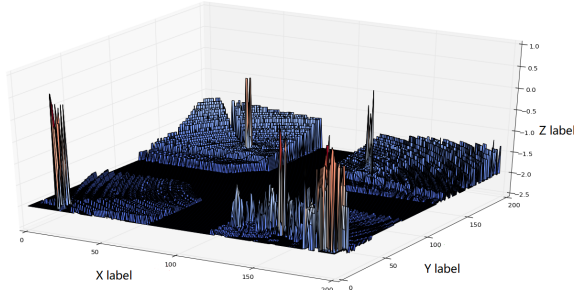


Fig. 1. 2.5D grid map example

distance metric. However, multiple iterations for searching the nearest point pair resulting in a high computational cost. And the result of ICP strongly dependent on the initial value.

The 3D-NDT [10] algorithm partitions the point cloud into 3D voxel-based representation. The multi-variable normal distribution in x, y, and z-axis for points in each cell is calculated. The probability sum of the target point cloud in the distributions of the referencing voxel-based representation subject to certain transition is used as the objective function. Since the function is differentiable, 3D-NDT can directly optimize the function using gradient descending methods such as Gauss-Newton, which is more efficient than the ICP. However, the high-resolution voxel-based representation can be computationally costly and somehow redundant especially in outdoor scenes.

Besides, the feature-based LO method was proposed for high-efficiency [11]. This method extracts planar and corner features for fast matching and achieves good results in various scenes when integrated with an IMU. However, these features can be sparse in road dominated autonomous driving scenes. And without IMU, LOAM tend to drift as shown in Sec. IV.

Similar to our method, the localization method for the outdoor environment based on a 2.5D representation was proposed in [12]. This method employs a multi-layer Gaussian mixture map (GMM) for describing the height variations in a 2D grid map. Each grid cell in the grid map is modeled by a one-dimensional GMM of the height, and all grid maps are stitched in offline to form a prior localization map. Such a representation is shown to be robust to environmental changes. However, this method can only be used for online localization based on the prior map. In addition, the EM algorithm is inefficient.

### III. METHOD

#### A. 2.5D grid map

We simplify the GMM-based 2.5D grid map proposed in [12] by leaving only the expectation of heights in each grid. Such a representation is much more efficient than the GMM-based map and can also indicate the height variation, which we found to be effective in the following matching process. Fig. 1 is an example of the 2.5D grid map. The point cloud data should first be rasterized in a 2D grid map. Let the spherical coordinates of the  $i$ -th LiDAR point  $p_i$

as  $(\gamma_i, \phi_i, \theta_i)$ , where  $\gamma$  is the depth of the point,  $\phi$  is the horizontal angle, and  $\theta$  is the vertical angle. The 3D Cartesian coordinates of  $p_i(x_i, y_i, z_i)$  is:

$$x_i = \gamma_i \cdot \cos\theta_i \cdot \cos\phi_i \quad (1)$$

$$y_i = \gamma_i \cdot \cos\theta_i \cdot \sin\phi_i \quad (2)$$

$$z_i = \gamma_i \cdot \sin\theta_i \quad (3)$$

We project the point cloud onto the 2D grid map. Here we use  $C$  to represent the projection process of the point cloud,  $(x'_i, y'_i)$  is the coordinate of  $p_i$  in the grid map:

$$Cp_i = p'_i(x_i, y_i) \quad (4)$$

in which:

$$x'_i = \text{floor}(x_i/f_x + c_x) \quad (5)$$

$$y'_i = \text{floor}(y_i/f_y + c_y) \quad (6)$$

where  $f_x, f_y$  are the grid map resolution in x and y axes,  $c_x, c_y$  are the center of the grid map, which are half of the numbers of rows and columns. The grid map is shown in Fig. 2(a). For each grid, we keep the height  $z$  of all points in the grid. Then the mean  $\mu$  is used to represent the height expectation for each grid:

$$\mu(p_i) = \frac{1}{n} \sum_{k=1}^n z_k, \quad n \geq 3 \quad (7)$$

where  $n$  is the number of points in  $p'_i$ . Thus a 2.5D grid map projected by the point cloud is obtained. This map retains the advantages of easy storage and building as a 2D grid map, and also represent the height information which is crucial for registration in the outdoor environment. It is for sure that a more precise model of height distribution can be further adopted in this 2.5D grid map, such as using normal distribution or GMM.

#### B. registration

Since the 2.5D grid map is analogous to "texture-less" gray-scale image as shown in Fig. 2, we adopt the direct method originated from VO to register two 2.5D grid maps [13]. In the direct method, The optimization is no longer the reprojection error used in indirect feature-based matching, but the photometric error, which is the difference between the gray values of the same positions in two frames.

The direct method is based on the illumination invariant assumption: the gray value of the same spatial point is constant in sequent images. For cameras, this assumption does not always hold for all scenarios. However, in our case, this assumption is interpreted into the height invariant assumption, which is true in most of the time since the average height of a grid in the 2.5D grid map does not change, no matter how the autonomous vehicle moves. The compensation of pitch angle is crucial in practice to settle height disturbance caused by vehicle maneuvers.

Since most of the grids represent the ground with the height gradient be 0, only the grids with the height gradient greater than a certain threshold are used to calculate the

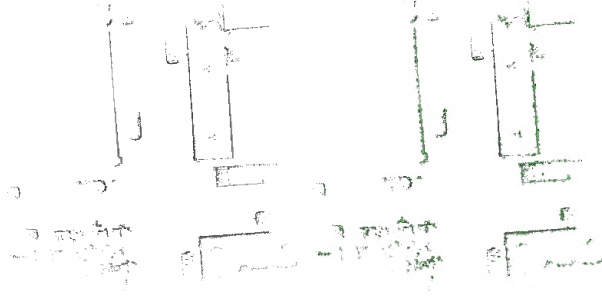


Fig. 2. Grid maps show: (a) LiDAR points projected into the grid. (b) Grids with high gradient colored in green

height difference error (HDE). Therefore, the semi-dense direct method is suitable for 2.5D grid map registration for its efficiency. Fig. 2(b) shows the grids with high gradient extracted, which are colored in green.

We use the Gauss-Newton method to solve the transition matrix  $T$  of two 2.5D grid maps, the HDE between two grid maps is minimized. The objective function  $J$  is

$$\arg \min_{R,t} J = \sum_{i=1}^N (\mu_1(Cp_i) - \mu_2(C(Rp_i + t)))^2 \quad (8)$$

where  $\mu_1(p)$  and  $\mu_2(p)$  represent the average height in grid  $p$ ,  $R$  is the rotation matrix and  $t$  is the translation vector. Because transition matrix can't be added directly, we can't access to the partial derivative of  $J$  with respect to  $T$ . As a result, Lie algebra  $\xi$  is used to express the pose:

$$T = \begin{bmatrix} R & t \\ 0^T & 1 \end{bmatrix} = \exp(\xi^\wedge) \quad (9)$$

Then the objective function becomes:

$$\arg \min_{\xi} J = \sum_{i=1}^N (\mu_1(Cp_i) - \mu_2(C\exp(\xi^\wedge)p_i))^2 \quad (10)$$

And the HDE  $e_i$  is modeled as:

$$e_i = \mu_1(Cp_i) - \mu_2(C\exp(\xi^\wedge)p_i) \quad (11)$$

To employ the Gauss-Newton method, we need to calculate the Jacobian matrix  $J_i$ , which is the derivation of the independent variable  $\xi$  by HDE  $e_i$ :

$$J_i = \frac{\delta e_i}{\delta \xi} \quad (12)$$

suppose:

$$q = \exp(\xi^\wedge)p \quad (13)$$

$$g = Cq \quad (14)$$

Because derivation of the  $\xi$  by  $e_i$  cannot be calculated directly, according to the chain rule, the  $J_i$  can be divided into three parts:

$$J_i = -\frac{\delta \mu_2}{\delta g} \frac{\delta g}{\delta q} \frac{\delta q}{\delta \xi} \quad (15)$$

From left to right:  $\frac{\delta \mu_2}{\delta g}$  is the partial derivative of grid, which is the grid gradient,  $\frac{\delta g}{\delta q}$  is the partial derivative of points in



Fig. 3. The TiEV autonomous vehicle platform

3D space by grid, and  $\frac{\delta q}{\delta \xi}$  is the partial derivative of Lie algebra by points in 3D space.

The grid gradient can be calculated as:

$$[(\mu_2(u+1, v) - \mu_2(u, v), \mu_2(u, v+1) - \mu_2(u, v))] \quad (16)$$

We use bilinear interpolation to calculate floating-point coordinates of  $[u, v]$ .

The left two partial derivatives can be calculated by:

$$\begin{aligned} \frac{\delta g}{\delta q} \frac{\delta q}{\delta \xi} &= \begin{bmatrix} f_x & 0 & -f_x x \\ 0 & f_y & -f_y y \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & -1 & y \\ 0 & 1 & 0 & 1 & 0 & -x \\ 0 & 0 & 1 & -y & x & 0 \end{bmatrix} \\ &= \begin{bmatrix} f_x & 0 & f_x x & f_x x y & -f_x - f_x x^2 & f_x y \\ 0 & f_y & -f_y y & f_y - f_y y & -f_y x y & -f_y y \end{bmatrix} \end{aligned} \quad (17)$$

According to the equation (16) and equation (17), Jacobian matrix  $J_i$  can be calculated, and Gauss-Newton method is used to optimize the objective function. The Gauss-Newton method is finally solved as:

$$\left( \sum_{i=1}^N J_i^T J_i \right) \delta \xi^* = - \sum_{i=1}^N J_i^T e_i \quad (18)$$

We can then obtain the autonomous vehicle's position in every frame through continuous registration between frames which is termed as the Direct LiDAR Odometry (DLO).

#### IV. EXPERIMENTS

We used the KITTI dataset and datasets captured by the TiEV<sup>1</sup> autonomous driving platform, shown in (Fig. 3) to verify the proposed algorithm. TiEV equips sensors including HDL-64, VPL-16, IBEO, SICK, vision, RTKGPS + IMU. In this experiment, we used HDL-64 LiDAR only to collect data at Jiading campus of Tongji University to test our method. In these experiments, the size of 2.5D grid map is 400 by 400, and each grid is 10cm by 10cm. The parameters in the algorithm were set as:

$$f_x = 0.1$$

$$f_y = 0.1$$

<sup>1</sup>cs1.tongji.edu.cn/tiev

$$c_x = 200$$

$$c_y = 200$$

The computing platform we use is a Core i5 4200U processor and 8GB memory. The operating system is Ubuntu 14.04.

#### A. KITTI dataset

KITTI [14] is a popular autonomous vehicle dataset in which the odometry data contains raw data from the camera, LiDAR, and poses of each frame. We used the point cloud data to test the proposed method and compare it with the ground truth. We also used open implementations of 3D-NDT<sup>2</sup> and LOAM<sup>3</sup> for comparison. Because DLO does not use the information of the IMU, we ran the LOAM program without the IMU data.

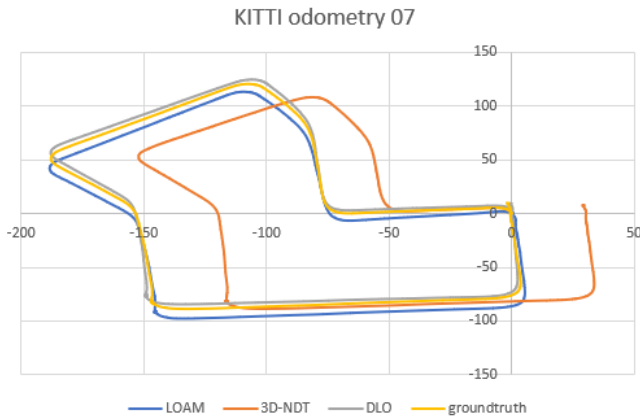


Fig. 4. Comparison between trajectories of DLO, 3D-NDT, LOAM (without IMU) and the ground truth using KITTI odometry dataset sequence 07

Fig. 4 shows the trajectories of the sequence 07 of the KITTI odometry dataset produced by DLO, 3D-NDT, LOAM and the ground truth. This sequence is composed of 1100 frames and the full length is over 700m. The result of DLO best conformed to the ground truth, with the absolute position error constantly below 3 meters as shown by Fig. 5, which depicts deviations (error in meters) from the trajectory point to the ground truth point at each frame. The error of 3D-NDT was considerably large, and LOAM is a little away from the groundtruth. However, LOAM still performed worse than DLO in this case. The corresponding error percentage is shown in Fig. 6, which depicts the ratio between position error and the trajectory length, starting from 50 meters.

Tab. I shows the running time of three methods on the dataset. It shows that DLO was much more efficient than 3D-NDT and LOAM as expected. 3D-NDT used all point cloud information for scan matching and optimized the object function up to convergence, which resulted in huge computational time. LOAM is known to be able to yield good results based on the VLP-16 lidar. However, when dealing with the denser HDL-64 data, it is necessary to decrease the

<sup>2</sup><http://pointclouds.org/>

<sup>3</sup>[https://github.com/laboshin/loam\\_velodyne](https://github.com/laboshin/loam_velodyne)

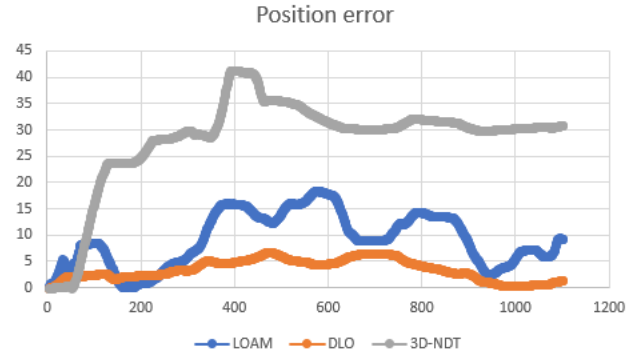


Fig. 5. Comparison between position errors of DLO, 3D-NDT, LOAM (without IMU) using KITTI odometry dataset sequence 07

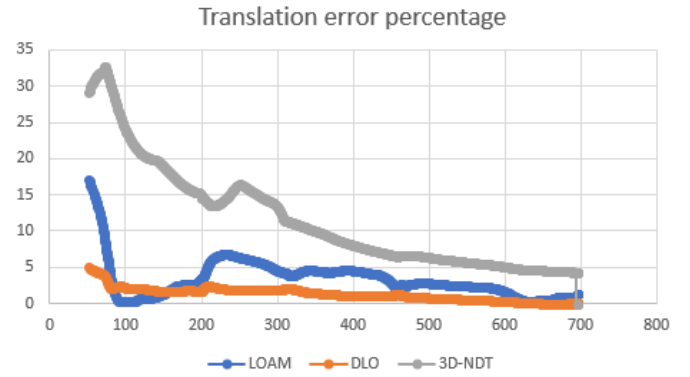


Fig. 6. Comparison between Translation error percentages of DLO, 3D-NDT, LOAM (without IMU) using KITTI odometry dataset sequence 07

number of edge and planar points and to down-sample the point cloud. The frame rate after careful tuning is at most 1hz (including both matching and mapping, the matching was conducted at 3hz), which is not comparable to DLO.

TABLE I  
RUNNING TIMES OF DLO, 3D-NDT, AND LOAM

	DLO	3D-NDT	LOAM
total time	44s	43153s	1025s
time/frame	0.04s	39.23s	0.93s

We further compared results generated based on different sizes of the grid in Fig. 7. We concluded that the size of 10cm by 10cm is a good choice considering both the accuracy and the time-consumption.

#### B. The dataset of Tongji university

We also tested DLO using dataset captured by TiEV as shown in Fig. 8. This is a closed outdoor route with a total length of about 1.3km, including turns with vagarious angles of about 60°, 90°, 120°, and a 270° turn in a roundabout. This dataset contains more than 2000 frames. During the experiment, we found the 3D-NDT took days to produce a result, so we compared DLO only with LOAM using this dataset.



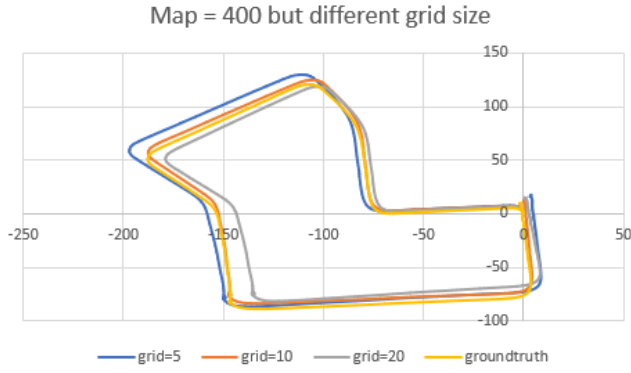


Fig. 7. The different size of the grid based on DLO

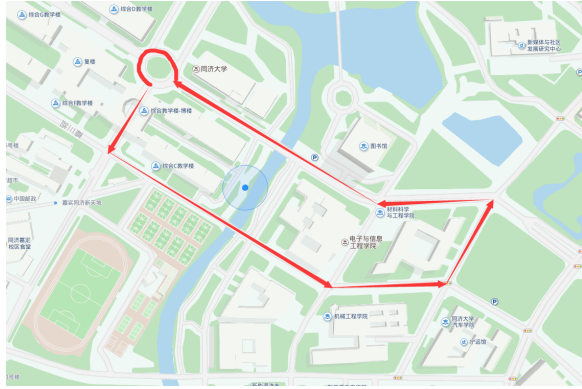


Fig. 8. The experimental route in the Jiading dataset

In Fig. 9, LOAM resulted in an inaccurate trajectory either in straight paths or at corners. The trajectory deviates from the starting point at around 200 meters at the end. The proposed DLO performed much better. However, it is also not completely closed. The distance between the starting point and the ending point is about 32 m, far less than the LOAM result without IMU. We found the error occurred mostly at corners which can be improved if IMU data is integrated. Moreover, the extra cost for estimating a 6-DoF pose in 3D-NDT and LOAM is restricted since all the used datasets comprise relative flat area. Therefore, it is appropriate to estimate the 3 DoF pose instead of the 6 DoF pose.

## V. CONCLUSIONS

In this paper, a DLO method based on the 2.5D grid map is proposed. This method is highly efficient and suitable for the outdoor environment. The point cloud of a frame is firstly projected onto a 2.5D grid map and then registered using the semi-dense direct method. Experiments with the KITTI dataset and the dataset we captured demonstrated that DLO outperforms 3D-NDT, both in accuracy and efficiency. Compared with the LOAM, which is the state-of-the-art in the KITTI benchmark list, DLO does not rely on extracted features from the point cloud. Therefore, the registration accuracy will be significantly higher in structureless scenes. The shortcoming of this method is the assumption of a flat world, which however makes sense for autonomous driving

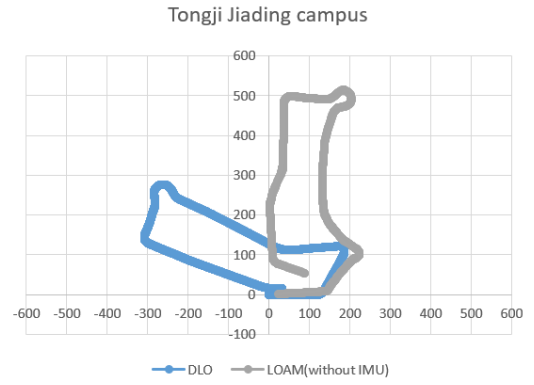


Fig. 9. Comparison of the trajectories of DLO and LOAM (without IMU) using Jiading dataset

tasks. Another possible improvement is to adopt the height distribution instead of the height expectation which will be explored in our future work.

## REFERENCES

- [1] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2017.
- [2] 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.
- [3] J. Engel, T. Schöps, and D. Cremers, "Lsd-slam: Large-scale direct monocular slam," in *European Conference on Computer Vision*. Springer, 2014, pp. 834–849.
- [4] C. Forster, M. Pizzoli, and D. Scaramuzza, "Svo: Fast semi-direct monocular visual odometry," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, 2014, pp. 15–22.
- [5] E. L. Lawler and D. E. Wood, "Branch-and-bound methods: A survey," *Operations Research*, vol. 14, no. 4, pp. 699–719, 1966.
- [6] 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.
- [7] S. Kohlbrecher, J. Meyer, O. von Stryk, and U. Klingauf, "A flexible and scalable slam system with full 3d motion estimation," in *Proc. IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR)*. IEEE, November 2011.
- [8] W. Hess, D. Kohler, H. Rapp, and D. Andor, "Real-time loop closure in 2d lidar slam," in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*. IEEE, 2016, pp. 1271–1278.
- [9] P. J. Besl, N. D. McKay *et al.*, "A method for registration of 3-d shapes," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 14, no. 2, pp. 239–256, 1992.
- [10] M. Magnusson, "The three-dimensional normal-distributions transform: an efficient representation for registration, surface analysis, and loop detection," Ph.D. dissertation, Örebro universitet, 2009.
- [11] J. Zhang and S. Singh, "Loam: Lidar odometry and mapping in real-time," in *Robotics: Science and Systems*, vol. 2, 2014.
- [12] R. W. Wolcott and R. M. Eustice, "Fast lidar localization using multiresolution gaussian mixture maps," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE, 2015, pp. 2814–2821.
- [13] X. Gao, T. Zhang, Y. Liu, and Q. Yan, *14 Lectures on Visual SLAM: From Theory to Practice*. Publishing House of Electronics Industry, 2017.
- [14] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.