Proceedings of 2016 IEEE
International Conference on Mechatronics and Automation
August 7 - 10, Harbin, China

An Efficient Scan-to-Map Matching Approach for Autonomous Driving

Hao Fu, Lei Ye, Rui Yu and Tao Wu
College of Mechatronic Engineering and Automation
National University of Defense Technology
Changsha, Hunan, P.R.China, 410073
fuhao@nudt.edu.cn

***Abstract*— Accurately localizing the vehicle against a pre-built high precision map is an essential step for the Autonomous Land Vehicle (ALV). This paper proposes an efficient scan-to-map matching approach based on multi-channel lidar. We firstly advocate the usage of both the lidar reflectance map and the height map, as these two maps contain complementary information. Then, borrowing ideas from image optical flow literature, we formulate the scan-to-map matching problem as an optical flow computation problem, and propose an efficient gradient descent approach to solve it. Finally, the proposed approach is integrated into a filtering framework for real-time online localization. Experiments on real-world dataset have demonstrated the validity of our approach. The dataset is also made publicly available for future research.**

***Index Terms*— Autonomous Driving, Scan Matching, Map Matching**

I. Introduction

Recently one has witnessed a rapid progress for autonomous driving. Several companies, including Google and Daimler, have all released their prototype autonomous driving cars. This rapid progress relies on the development from traditional pattern recognition field, and benefits a lot from the utilization of high precision map.

Unlike traditional satellite maps or commercial navigation maps, high precision map contains much more accurate information which could aid the Autonomous Land Vehicle (ALV) in at least three aspects. Firstly, through a sophisticated scan-to-map matching procedure, the ALV could be accurately localized on the map. Secondly, by manually annotating the objects of interest on the map, an accurate prior for the static part of the surrounding scene could be obtained, including traffic light, traffic sign, lane marking, curbs, barriers, etc. Thirdly, as illustrated in [7], the shape of the road obtained from the map could also aid the tracking of other dynamic traffic participants, and help predict their behaviours. To fully utilize those benefits brought by the high-precision map, the ALV firstly needs an efficient scan-to-map matching module.

In this paper, an efficient scan-to-map matching algorithm based on Velodyne 32 or Velodyne 64 sensor is proposed. This kind of sensor, also known as multi-channel lidar, is equipped with multiple laser ray emitters and receivers, which are grouped into several bundles. With the lidar spinning at a high constant rate, these bundles fire sequentially, thus generating a dense 3D point cloud of the environment. As the vehicle moves, the point cloud captured at different positions could then be assembled to generate the high precision map. The map could be represented in different forms, such as the 3D point cloud map, the 2D binary obstacle map or the 2D gray scale map, where the gray value could either represent the reflectance intensity or the height value. This is illustrated in Fig.1. To perform the scan-to-map matching, the current scan should also be transformed to the same modality before it is matched to the map.

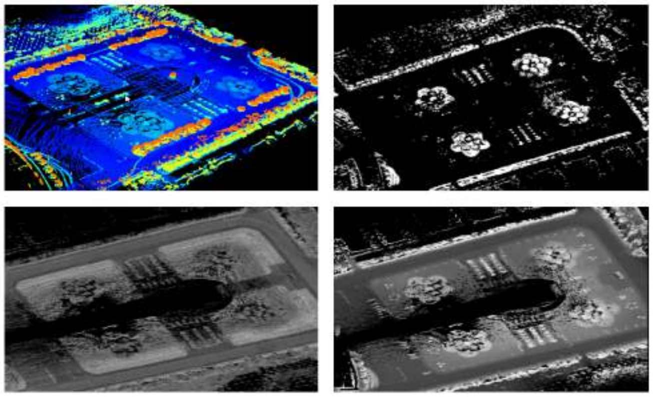

Fig. 1. Different forms of the map. While the 3D point cloud map (top left) is too computational intensive, the 2D binary obstacle map (top right) causes severe information loss. The 2D gray-scale map, either the height map (bottom right) or the reflectance intensity map (bottom left), is a good compromise between computational load and information loss.

For the 3D point cloud map, existing approaches, including Iterative Closest Point (ICP) [2] or Normal Distribution Transform (NDT) [3] could be adopted as the scan-to-map matching module. However, directly comparing two 3D point cloud is a laborious work, and storing the whole 3D point cloud map is very memory consuming, which hinders its usage for real-time application, especially when a rapid localization rate is required, such as driving on the highway or coming to the re-localization. On the contrary, the 2D obstacle map greatly simplifies the map representation, and

978-1-5090-2396-7/16/$31.00 ©2016 IEEE

only stores the obstacle information in binary form. This simplification may greatly accelerate the matching speed, but some key environmental information might get lost.

We believe that the 2D gray-scale map, compared to these two map forms, is a good compromise between computational load and information loss. As the ALV mostly runs on the road which is usually a planar surface, simplifying the 3D point cloud map to 2D gray scale map might not cause severe information loss. The local 2D gray scale map could be directly produced by the multi-channel lidar. To compare the 2D gray scale maps, we didn't notice any principled approaches existing in the scan matching literature. In this paper, we draw inspirations from image optical flow literature. We formulate the scan-to-map matching problem as an optical flow computation problem, and propose an efficient gradient descent approach to solve it. We also compare the performance between our approach and the well-known ICP approach for aligning 2D or 3D point cloud. Experiments on real-world dataset demonstrate the superiority of the proposed approach over its counterpart. The dataset is also made publicly available for future research.

II. Related Work

To compare a lidar scan with the map, existing approaches could be categorized into four types based on how they model the sensor readings: beam sensor model, likelihood field model, map matching model and point cloud model.

For the beam sensor model, it tries to represent the full beam readings by modeling the physical causes of the beam measurement. It is mostly used for inaccurate sensors, such as sonar [16].

For the likelihood field model [15], [10], which is also known as beam end-point model, it does not consider the full beam readings. Instead, it only considers the beam end-point. The distance from the end-point to the nearest obstacle is computed as the proximity measure. To accelerate the computation, the likelihood field of the map is usually pre-computed over the 2-D grid, thus preventing the calculation of the distance from the beam end-point to the nearest obstacle. This likelihood field computation step is quite similar to the distance transform technique [6] used in image processing area.

For the map matching model, a local map is firstly generated from local scans or even one single scan from the multi-channel lidar, and then matched against the pre-built global map. Normalized Cross Correlation (NCC) [12] is usually adopted as the similarity measure.

For the point cloud model, both the current scan and the map are treated as a set of points. These two point clouds are then to be aligned using popular techniques such as ICP. Besides ICP, the point cloud could also be represented as a probabilistic density function, and techniques such as Normal Distribution Transform (NDT) could then be utilized. Features or segments could also be extracted from the point cloud and then used for matching [5].

In this paper, we focus on the map matching model. Compared to other models, we believe that the map matching model is more suitable for the multi-channel lidar data. Firstly, as the lidar produces very accurate range measurements, always in centimeter level, we could safely ignore the beam measurement error as considered in the beam sensor model. Secondly, the multi-channel lidar produces 3D representation of the environment. If we chose to use the likelihood model, the likelihood field representation will also need to be 3D, thus greatly enhancing the computational burden. The 3D point cloud model is also very computational intensive, prohibiting its usage for real-time application.

On the contrary, by adopting the 2D grid map representation, the map matching model achieves a good compromise between rich representation and low computational load. Besides, the multi-channel lidar not only produces accurate range measurements, but also the reflectance intensity. This intensity value contains rich information, and is used as the sole input for the localization module as proposed in [11]. The other three models will all encounter difficulties in utilizing this information, whilst the map matching model is directly applicable by adopting the map form of the 2D gray-scale reflectance intensity map.

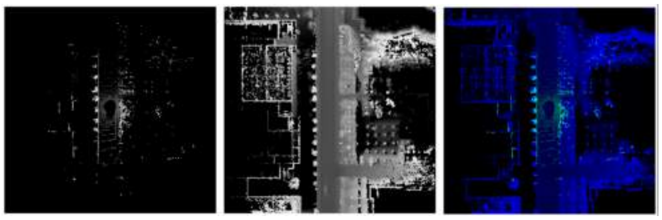

Fig. 2. Illustration of aligning a gray-scale image to the 2D gray-scale map. Given a current scan, it is firstly transformed to a gray-scale image as shown on the left image. The gray value might either represent the reflectance intensity value or the height value. This gray-scale image is then matched against a pre-built map (the middle image). The aligned result is shown in false color in the right image.

III. Scan to map matching algorithm

This paper focuses on utilizing multi-channel lidar data for accurate localization on the map. This problem is mathematically formulated as follows:

Given the current scan T, after some transformation f, the transformed scan $T' = f(T)$ is needed to match against the map M to obtain the correct pose adjustment P. As illustrated in Fig.1, the map may be represented in different forms. The aim of the transformation f is to transform the scan to the same modality as the map. In this paper, we focus on the map form of the 2D gray scale map.

A. Generating the 2D gray scale map

The map is generated in an offline fashion similar to [11], [17], [18]. To generate the map, a pose graph is firstly generated where each node represents a pose. The pose graph contains three kinds of edges which represent the different constraints: the GPS prior constraint over each node, the odometry constraint between both spatially and temporarily neighboring nodes and the scan matching constraint between spatially neighboring nodes. In generating the scan matching constraint, we adopt the LOAM algorithm proposed in [19]. Different from traditional ICP approaches and its variants [1], [14] , which either optimize the point-to-point distance or the point-to-plane distance, the LOAM algorithm firstly separates edge points from the surface points based on a measure of curvature. For the edge points, point-to-line distance is adopted as the distance measure while the point-to-plane distance is utilized for the surface points. Therefore, this approach is more theoretically sound than the ICP algorithm.

The pose graph is then optimized using any pose graph optimization (PGO) algorithms, such as incremental smoothing and mapping (iSAM) [9] or g2o [13]. Based on the optimized poses, the scans are then assembled to generate the map. The map is discretized at a fixed resolution of 20cm. For all points falling on the same cell, four statistical values [11] are calculated to represent that cell: the mean of the height value, the variance of the height value, the mean of the reflectance intensity value and the variance of the reflectance intensity value.

The reasons that we represent each cell using both the reflectance value and the height value is that these two sources contain complementary information. Just as shown in Fig.3, we can clearly observe the lane markings on the ground from the intensity map, while the obstacles such as trees and buildings could be observed from the height map. Only when these two sources of information are combined, could we obtain a more comprehensive understanding of the surrounding scene.

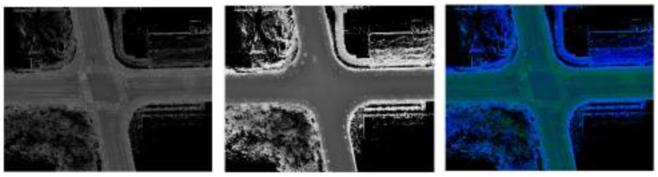

Fig. 3. From left to right: the intensity map, the height map, and the fusion of the two maps. It is clearly seen that the intensity map and the height map contain complementary information.

Note that we model each cell using a single Gaussian distribution instead of multi-modal Gaussian distribution as proposed in [18]. This representation might lose some information contained in the original 3D point cloud. However, we believe that as the vehicle mostly runs on planar surface in city area, this simplification might not cause severe information loss. Besides, in aggregating the 3D points to the 2D map, we purposely omit the overhanging objects by a thresholding operator over the margin between the high points and low points on each cell. This operation is also performed to the current scan in the online localization setting.

This map representation could be easily extended to generate multi-resolution map [18] by setting a larger cell size, such as 40cm or 80cm.

B. Scan-to-map Matching

Given the current transformed scan T (we dropped the superscript for readability) and the map M, we are trying to minimize the energy function:

$$E(\mathbf{p})\doteq\sum_{\mathbf{x}}\mathtt{dist}(M(\mathbf{W}(\mathbf{x};\mathbf{p})),T(\mathbf{x})) \tag{1}$$

where $\mathbf{p}$ is the parameter (lidar's global pose) to be estimated, and $\mathbf{x}$ loops through all the map cells whose gray-scale values are nonzero in the current scan. $\mathbf{W}$ is the transformation to be estimated. $\mathtt{dist}(\cdot,\cdot)$ represents the distance, i.e., the inverse similarity measure between the two cells. As the cell is represented using both the mean and variance of the height and the intensity value, therefore, the distance could be defined as:

$$\mathtt{dist}(M,T)\doteq\frac{(\mu_i(M)-\mu_i(T))^2}{(\sigma_i(M)+\sigma_i(T))^2}+\frac{(\mu_z(M)-\mu_z(T))^2}{(\sigma_z(M)+\sigma_z(T))^2} \tag{2}$$

where $(\mu_i,\sigma_i,\mu_z,\sigma_z)$ are the mean and variance of the intensity and the height value.

This distance measure encourages the matched cells to have similar intensity and height value. Those cells with larger variance will tend to produce a smaller distance measure, and thus contribute less to the final energy score [11]. Although the above distance measure is more theoretically sound, the variance term in the denominator will make the computation of the partial derivative of the whole energy term much more complicated. If we omit the variance term and simply use the difference between the mean value, the energy term will become easily tractable. In practice, we found this omission seldom deteriorates the system performance. This is mostly due to the fact that we ignore the overhanging objects, and each cell simply represents a stick on the ground. This makes the mean value proportional to the variance value of each cell.

We assume the transformation $\mathbf{W}$ to be a translation after a rotation:

$$\mathbf{W}(\mathbf{x};\mathbf{p})=\begin{pmatrix} cos(\phi) & -sin(\phi) & d_x \\ sin(\phi) & cos(\phi) & d_y \end{pmatrix}\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \tag{3}$$

Thus we have $\mathbf{p}=[\phi,d_x,d_y]$. For readability, we define $T(\mathbf{x})\doteq[\mu_i(T),\mu_z(T)]^\mathsf{T}$ and $M(\mathbf{x})\doteq[\mu_i(M),\mu_z(M)]^\mathsf{T}$. Thus equation (1) could be written as:

$$E(\mathbf{p}) = \sum_{\mathbf{x}} \|M(\mathbf{W}(\mathbf{x};\mathbf{p})) - T(\mathbf{x})\|_2 \tag{4}$$

where $\|\cdot\|_2$ represents the L2 norm. The partial derivative of $\mathbf{W}$ with respect to $\mathbf{p}$ is:

$$\frac{\partial \mathbf{W}}{\partial \mathbf{p}} \mid_{\mathbf{p}=\mathbf{p}'} = \begin{pmatrix} -sin(\phi') * x - cos(\phi') * y & 1 & 0 \\ cos(\phi') * x - sin(\phi') * y & 0 & 1 \end{pmatrix} \tag{5}$$

Given an initial $\mathbf{p}$, we are trying to find $\Delta\mathbf{p}$ to minimize (4). Writing equation (4) in incremental form:

$$minimize \sum_{\mathbf{x}} \|M(\mathbf{W}(\mathbf{x};\mathbf{p}+\Delta\mathbf{p})) - T(\mathbf{x})\|_2 \tag{6}$$

$$M(\mathbf{W}(\mathbf{x};\mathbf{p}+\Delta\mathbf{p})) \approx M(\mathbf{W}(\mathbf{x};\mathbf{p})) + \frac{\partial M}{\partial \mathbf{x}} \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \Delta\mathbf{p} \tag{7}$$

Thus, equation (6) can be re-written as:

$$\arg\min_{\Delta\mathbf{p}} \sum_{\mathbf{x}} \|M(\mathbf{W}(\mathbf{x};\mathbf{p})) + \nabla M \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \Delta\mathbf{p} - T(\mathbf{x})\|_2 \tag{8}$$

The solution to this equation is:

$$\Delta\mathbf{p} = H^{-1} \sum_{\mathbf{x}} \left[\nabla M \frac{\partial \mathbf{W}}{\partial \mathbf{p}}\right]^{\mathsf{T}} [T(\mathbf{x}) - M(\mathbf{W}(\mathbf{x};\mathbf{p}))] \tag{9}$$

where H is the Hessian matrix.

Besides, the inverse of this Hessian matrix also reflects the matching uncertainty [10], which enables this approach to be seamlessly integrated into a Bayesian filtering framework.

After $\Delta\mathbf{p}$ is calculated, it is added to $\mathbf{p}$, and the above computation repeats. The iteration will stop only if a predefined max iteration number is reached or the computed $\Delta\mathbf{p}$ is smaller than a threshold. In practice, we found that as the iteration goes on, $\mathbf{p}$ might be fluctuated around the minimum. To ease the fluctuation, a weight decaying factor λ_t is multiplied to $\Delta\mathbf{p}$ before it is added to $\mathbf{p}$.

C. Integrating into a localization system

Just like all iterative optimization approaches, the time consumption for the scan-to-map matching algorithm depends on the iteration number, which is always not a constant. This iteration number highly depends on the initial estimate. The more accurate the initial estimate is, the less iteration number is required, and the more quickly the algorithm will converge.

To obtain an accurate initial position estimate, we can either use the wheel odometry or the GPS position estimate. However, the vehicle wheel might be slippery and the GPS might be very unreliable due to high building blockage. Therefore, we propose the following matching strategy. Instead of directly matching each scan to the map, the current scan is firstly matched to the previous scan using any off-the-shelf scan-to-scan matching algorithms, such as ICP or LOAM. In this step, the initial transformation estimate is obtained from the previous time step, and we only search possible transformations in the close vicinity of this estimate. In practice, we find this step of scan registration seldom produces errors.

The computed transform is then added to the previous map position, and generates the initial map position estimate for the current scan. As this initial position estimate is very accurate, the following scan to map matching step is greatly accelerated, and usually converges in a few iterations.

For online localization, the above described scan-to-map matching approach is integrated into a filtering framework. Either Extended Kalman Filter (EKF) or Partical Filter (PF) could be adopted. For EKF, we apply similar techniques as in [17]. For PF, our approach performs as a local search module, which makes the proposal distribution data dependent. It has been shown in previous work [8] that with much fewer particles, a better proposal distribution could obtain comparable or even better performance than the original PF.

IV. Experiments

We conduct experiments in real-traffic scenario. Our autonomous car is equipped with a Velodyne 64 lidar and an integrated Novatal GPS/IMU device. It drives in the suburb area located in the north part of Changsha city. The route is about 12 kilometers long and contains many loops. The map is built using incremental smoothing and mapping (iSAM) [9] algorithm which fuses the scan-matching constraint, the odometry constraint and the GPS constraint. The generated map is shown in Fig.4(a). The red and green dots shown in the image represent the GPS trajectory and the iSAM-optimized trajectory. If we directly use the GPS trajectory, the generated map will look like the one shown in Fig.4(b), while the iSAM-optimized trajectory will generate the map shown in Fig.4(c), which is much clearer than Fig.4(b).

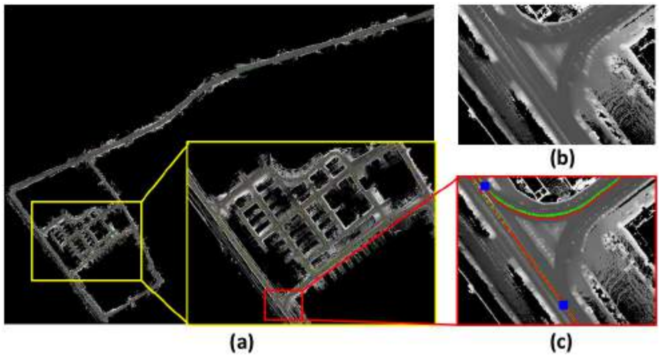

Fig. 4. Fig.(a) shows the 2D gray scale height map generated by offline iSAM algorithm [9]. Fig.(c) shows an enlarged map, where the red dots and green dots represent the raw integrated GPS/IMU trajectory and the iSAM-optimized trajectory. The blue dots represent positions which we uniformly sampled from the whole trajectory to form our 100-scan dataset. Fig.(b) is the map generated by using the raw integrated GPS/IMU trajectory. It is much more blurred than Fig.(c).

From the whole iSAM-optimized trajectory, we uniformly sampled 100 positions which consist our dataset. These positions are shown as the blue dots in Fig.4(a) and (c).

A. *Experiments on the 100-scan dataset*

We consider the iSAM-optimized positions for these 100 scans as their ground-truth positions. To testify the scan-to-map matching algorithm, we added different levels of noise to these positions. The scan-to-map matching algorithm will then start from these noise corrupted positions and to see whether they would converge to the ground-truth positions.

For the baseline approach, we perform Iterative Closest Point (ICP) [2] on the 2D obstacle map. In generating the obstacle map, following [4], we firstly segment each scan into three classes according to different height intervals: ground, obstacle and hanging objects. Those obstacle points are then projected onto the 2D grid cell to generate the local obstacle map. These local obstacle maps are then assembled based on their iSAM-optimized positions to generate the global obstacle map. In the matching step, each occupied cell in the obstacle map is represented as a 2D point. All the points from the local obstacle map and the global obstacle map form the local point cloud and the global point cloud, which are then to be matched using the ICP algorithm. As the ICP algorithm is performed on the 2D point cloud, we term this approach as 2D-ICP.

It is obvious that the 2D obstacle map loses the height information which might be essential for the matching procedure. To incorporate the height information, we propose to combine the obstacle map with the height map. For those cells considered as obstacles, its height value is quantized into discrete bins (We use 16 bins in this paper.), and we represent that cell using the same number of points as the occupied bins. In this way, two obstacle cells with different height values could be distinguished. We term this approach as 2.5D ICP. It differs from 3D ICP in that the number of variables to be estimated is three (x,y,azimuth) instead of six (x,y,z,azimuth,pitch,roll). It differs from 2D ICP in that each obstacle cell is represented by a number of duplicate points instead of only one single point. The reason that we didn't consider 3D ICP is that the matching speed is always too slow, usually longer than 1 second for one match, and storing the whole 3D point cloud map is very memory consuming, prohibiting its usage for real-time online localization. The concept of both 2D ICP and 2.5D ICP are illustrated in Fig.5.

We perform experiments on the 100-scan dataset. For each scan with its iSAM-optimized ground-truth position, we added zero-mean Gaussian noise with variance level up to 6 meters or 6 degrees. The noise is added to both x, y and azimuth. We repeat the experiments for 20 times, resulting in a total number of 2000 trials. For each trial, we perform 2D ICP, 2.5D ICP and our proposed approach. These approaches start from the same noise corrupted position. The corrected position is then compared with the ground truth position to obtain the error for each method.

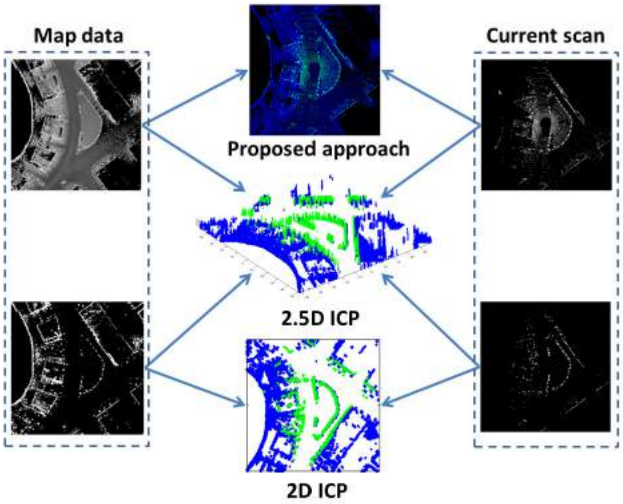

Fig. 5. Illustration of 2D ICP, 2.5D ICP and our proposed approach. Our proposed approach performs on the gray-scale map, whilst 2D ICP performs on the 2D binary obstacle map. To incorporate the height information, 2.5D ICP represents each occupied cell with a number of duplicate points, where the duplicate number is proportional to the height value.

Fig.6 shows the registration error under different noise level. It is observed that under the noise level of 2 meter and 2 degree, all these three approaches could reduce the error to be around 0.5 meter and 0.5 degree. However, as the noise level increases, both 2D ICP and 2.5D ICP performs much worse than the proposed approach. At the noise level of 5 meter and 5 degree, 2D ICP and 2.5D ICP will produce errors around 3 meter and 2 degree, while the proposed approach could reduce it to around 1.5 meter and 1 degree. Note that the noise level of 5 meter and 5 degree does not necessarily mean that the noise is exactly 5 meter and 5 degree. They represent the variance value of the zero-mean Gaussian distribution. It is possible that during some trials, the noise could be as large as 15 meters and 15 degree.

B. *Real-vehicle experiments*

The proposed scan-to-map matching algorithm is also incorporated into an Extended Kalman Filtering (EKF) framework for real-time online localization. In the EKF framework, the proposed scan-to-map matching algorithm performs as the observation model. The motion model is built using the 2D-ICP matching of two successive scans.

The iSAM optimized positions obtained offline are considered as the ground-truth. We compare our localization accuracy with the output from a high-end Novatel INS system. The results are shown in Fig.7. While the Root Mean Square (RMS) lateral and longitudinal error for the INS system are 84.5cm and 79.7cm, our localization system reduce it to 14.6cm and 15.5cm respectively. Note that this error is smaller than the size of one grid cell, which is 20cm throughout all the experiments.

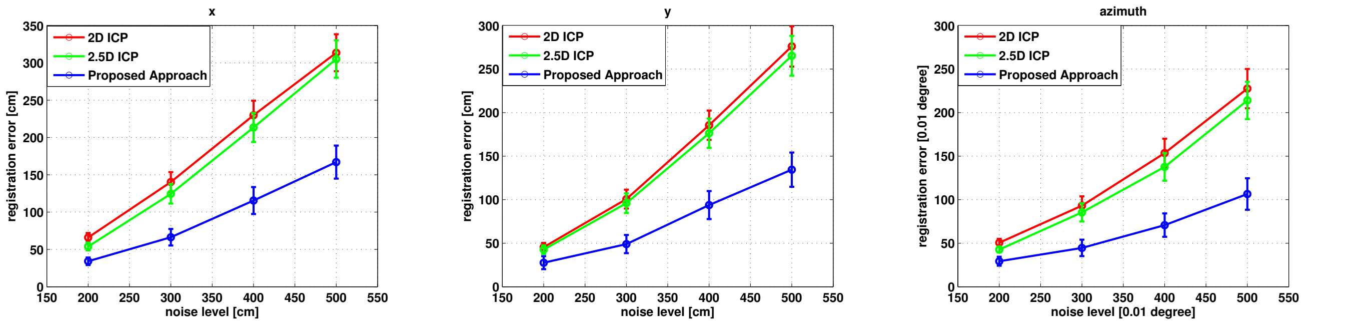

Fig. 6. Registration error in x, y and azimuth for 2D ICP, 2.5D ICP and the proposed approach.

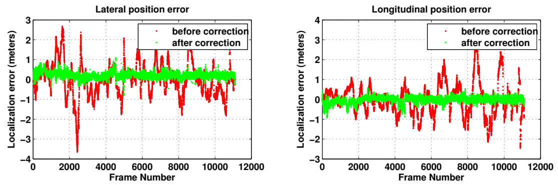

Fig. 7. The iSAM-optimized positions are deemed as the ground-truth. The localization error of a high-end Novatel INS system is shown as the red dots, which may go up to 3 meters. The green dots represent the localization error produced by our proposed approach.

V. Concluding Remarks

In this paper, we proposed a novel approach for scan-to-map matching. Different from existing approaches which either rely on the reflectance intensity map or the height map, our approach naturally utilizes both modalities in a principled way. We cast the scan-to-map matching problem as an optical flow computation problem, and propose an efficient gradient descent approach to solve it. Experiments on real-world dataset demonstrates the validity of our approach. This dataset will also be made public available.

Acknowledgment

The authors would like to thank the National Scientific Foundation of China for the funding of Grant 61503400 and 91220301.

References

[1] K S Arun, T.S. Huang, and S.D. Blostein. Least-Squares Fitting of Two 3-D Point Sets. *IEEE Trans on Pattern Analysis and Machine Intelligence*, pages 698–700, 1987.

[2] P.J. Besl and H.D. McKay. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.

[3] Peter Biber and W Straßer. The normal distributions transform: A new approach to laser scan matching. In *International Conference on Intelligent Robots and Systems*, pages 2743 – 2748, 2003.

[4] Tongong Chen, Bin Dai, Liu Daxue, and Ruili Wang. Gaussian-Process-Based Real-Time Ground Segmentation for Autonomous Land Vehicles. *Journal of Intelligent & Robotic Systems*, 76(3-4):563–582, 2013.

[5] B. Douillard, A. Quadros, P. Morton, J.P. Underwood, M. De Deuge, S. Hugosson, M. Hallstrom, and T. Bailey. Scan segments matching for pairwise 3D alignment. In *IEEE International Conference on Robotics and Automation*, pages 3033–3040, 2012.

[6] Ricardo Fabbri, Luciano Da F. Costa, Julio C. Torelli, and Odemir M. Bruno. 2D Euclidean distance transform algorithms. *ACM Computing Surveys*, 40(1):1–44, 2008.

[7] Tobias Gindele, Sebastian Brechtel, Joachim Schröder, and Rüdiger Dillmann. Bayesian Occupancy Grid Filter for Dynamic Environments Using Prior Map Knowledge. In *Intelligent Vehicles Symposium (IV)*, pages 669–676, 2009.

[8] Giorgio Grisetti, Cyrill Stachniss, and Wolfram Burgard. Improved techniques for grid mapping with Rao-Blackwellized particle filters. *IEEE Transactions on Robotics*, 23(1):34–46, 2007.

[9] Michael Kaess, Student Member, and Ananth Ranganathan. iSAM : Incremental Smoothing and Mapping. *IEEE Trans. on Robotics*, 24(6):1365–1378, 2008.

[10] Stefan Kohlbrecher, Oskar Von Stryk, Johannes Meyer, and Uwe Klingauf. A Flexible and Scalable SLAM System with Full 3D Motion Estimation. In *IEEE International Symposium on Safety, Security and Rescue Robotics*, pages 155–160, 2011.

[11] Jesse Levinson and Sebastian Thrun. Robust vehicle localization in urban environments using probabilistic maps. In *IEEE International Conference on Robotics and Automation*, pages 4372–4378, 2010.

[12] J.P. Lewis. Fast Normalized Cross-Correlation. *Vision Interface*, pages 120–123, 1995.

[13] K Rainer, Giorgio Grisetti, Hauke Strasdat, Kurt Konolige, and Wolfram Burgard. g2o: A General Framework for Graph Optimization. In *ICRA*, pages 3607–3613, 2011.

[14] Aleksandr V Segal, Dirk Haehnel, and Sebastian Thrun. Generalized-ICP. In *Proc. of Robotics: Science and Systems*, volume 2, pages 1–8, 2009.

[15] S. Thrun. A Probabilistic On-Line Mapping Algorithm for Teams of Mobile Robots. *The International Journal of Robotics Research*, 20:335–363, 2001.

[16] Sebastian Thrun. Learning Occupancy Grid Maps With Forward Sensor Models. *Autonomous Robots*, 15(2):111–127, 2003.

[17] Ryan W Wolcott and Ryan M Eustice. Visual localization within LIDAR maps for automated urban driving. In *IROS*, pages 176–183, 2014.

[18] Ryan W Wolcott and Ryan M Eustice. Fast LIDAR localization using multiresolution Gaussian mixture maps. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2814–2821, 2015.

[19] Ji Zhang and Sanjiv Singh. LOAM : Lidar Odometry and Mapping in Real-time. In *Robotics: Science and Systems*, pages 1–8, 2014.