Available online at www.sciencedirect.com

ScienceDirect

IFAC Papers Online CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 52-30 (2019) 167–172

Online 3D Mapping and Localization System for Agricultural Robots

Tuan Le * **Jon Glenn Omholt Gjevestad** * **Pål Johan From** *

* *All authors are with Department of Science and Technology, Norwegian University of Life Sciences, Ås 1430, Norway.*
emails: {tuan.dung.le, jon.glenn.gjevestad, pal.johan.from}@nmbu.no

Abstract: For an intelligent agricultural robot to reliably operate on a large-scale farm, it is crucial to accurately estimate its pose. In large outdoor environments, 3D LiDAR is a preferred sensor. Urban and agricultural scenarios are characteristically different, where the latter contains many poorly defined objects such as grass and trees with leaves that will generate noisy sensor signals. While state-of-the-art methods of state estimation using LiDAR, such as LiDAR odometry and mapping (LOAM), work well in urban scenarios, they will fail in the agricultural domain. Hence, we propose a mapping and localization system to cope with challenging agricultural scenarios. Our system maintains a high quality global map for subsequent reuses of relocalization or motion planning. This is beneficial as we avoid the unnecessary repetitively mapping process. Our experimental results show that we achieve comparable or better performance in state estimation, localization, and map quality when compared to LOAM.

Keywords: Agricultural robot, 3D mapping, localization.

1. INTRODUCTION

LiDAR mapping and localization has been widely studied in the literature Magnusson et al. (2007); Kohlbrecher et al. (2011); Zhang and Singh (2014). However, most focus on indoor, urban or city scenarios. The difference in characteristic of urban and agricultural scene is significant. In an urban scene such as a city, sufficient features such as lines, planes, corners from houses, pavements, etc., can be extracted for scan registration. In an agricultural scene, objects such as grass, tall trees, tree leaves can not provide reliable features for detection to the same extent. For example, a tree leaf is unlikely to be observed twice in two consecutive scans. The ground in a farm is more likely to be rugged and not flat as a city street. These challenges prevent directly applying conventional method such as LOAM.

In this work, we propose a complete *online 3D mapping and localization* for our agricultural mobile robotic platform Thorvald II Grimstad and From (2017). The robot is capable of *i)* incrementally building and localizing in a 3D map using 3D point cloud data, *ii)* the global built map can be stored for subsequent reuse. Specifically, an optimization-based approach is used for estimating the robot odometry. We also employ loop-closure detection to ensure the large built 3D map is consistent and usable for later tasks without rebuilding it every time. For relocalization in a pre-built 3D map, we employ a normal distribution transformation (NDT) scan matching method in Stoyanov et al. (2012). Both processes (map building and relocalization) are guaranteed to run online on the robot onboard computer. In summary, we highlight the contributions of this work as follows:

- a complete online 3D LiDAR mapping and localization system for autonomous agricultural robots
- high quality built map for human operator and subsequent reuse
- an evaluation of the proposed system on both simulation and real experiments

We notice that existing methods such as LOAM can store its built map and use it for relocalization purposes. However, the authors of LOAM does not focus on this functionality. Hence, our proposed system fills in the gap for agricultural applications. Even though our method is not suitable for crop field environments, where the appearance of plant gradually changes, it is still applicable for other agricultural tasks such as product transportation between fields and storage, between polytunnels. Therefore, we argue that our proposed system is still useful.

The paper is organized as follows: In section II, we review related work. Section III depicts our hardware system overview. Section IV and V discuss the 3D LiDAR map building and localization. Experimental results are presented in Section V and conclusions are discussed in Section VI.

2. RELATED WORK

Several works on mapping and localization in agricultural domain have been focused on crop field environment. Early work by Khanna et al. (2015) proposed a simple mapping solution by using a stereo camera for generating 3D point-cloud but using a commercial software. Albani et al. (2017) proposed a decentralized multi-UAV system for crop field mapping and weed detection. However, the system was only tested in simulation without any validation from

2405-8963
Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2019.12.516

real field. Popović et al., Popovi et al. (2017) proposed a Gaussian Process model for generating a multiresolution map for biomass monitoring. More recently, Chebrolu et al., Nived et al. (2019) combined aerial images and ground images for localizing in a prebuilt-aerial map of a sugar beet field. The aerial map is continuously updated after each session to maintain a high localization accuracy.

Beside crop fields, a robot might need to travel to other parts of a farm. For example, the robot might need to transport harvested products from crop fields to storage. For this task it also requires a good 3D map since the terrain on a farm is unlikely to be globally flat. Therefore, in this work, we aim to solve a 3D mapping and localization problem using 3D LiDAR for agricultural logistics application.

We focus on geometry approach for LiDAR odometry estimation. The state-of-the-art method, LOAM, is presented in Zhang and Singh (2014, 2017). The method leverages point feature to edge/plane scan-matching for scan registrations. The state estimation is further divided into a cascade system: velocity is estimated with low accuracy but at high frequency and motion estimation runs at low frequency but returns high accuracy estimation. The fused output of the system is a high frequency and high accuracy motion estimation. The result of odometry estimation by LOAM is still by far the best on the KITTI odometry benchmark [1] .

We notice a couple drawbacks that prevent us from directly implementing the original LOAM method. First, LOAM needs to iterate through every point in a given point cloud to compute features for scan matching. This poses a computational bottle neck. Second, an agricultural robot is likely to work in an environment filled with trees, grass, which makes detected features unreliable. For example, an edge feature from a tree leaf is unlikely to be observed twice for matching. Or grass with uneven height on the ground might give inconsistent planar features. And lastly, since LOAM focuses solely on odometry estimation, no loop closure or saving built map functionality is implemented. This prevents an agricultural robot from operating efficiently since it needs to rebuild a map of a large scale environment every time it is turned on. The work in Shan and Englot (2018) is the most similar to ours, however, like the original LOAM, the authors focus on odometry estimation only.

We are inspired by an NDT-based approach for localization in Sakai et al. (2017). However, the authors in Sakai et al. (2017) use a 2D-3D matching while we directly perform a 3D-3D matching. We argue that for agricultural environments, where features are sparse, the use of 2D LiDAR would severely limit the matching process for localization. Hence, we prefer a 3D-3D matching method.

3. SYSTEM OVERVIEW

3.1 Hardware system overview

The robotic system used in this work is an agricultural mobile platform Thorvald II Grimstad and From (2017). The robot is four wheel drive with a modular design.

[1] http://www.cvlibs.net/datasets/kitti/eval_odometry.php

The robot is equipped with a 3D LiDAR Velodyne VLP-16 and a commercial grade IMU Xsens MTi-30. The complete hardware system is shown in Fig.1b.

3.2 Software system overview

The 3D mapping process is divided into three steps. First, incoming LiDAR measurement is preprocessed to separate a set of ground points from non-ground points. The set of non-ground points is further segmented into different clusters, each cluster containing points from one single object. Both set of ground points and object clusters are used for extracting edge and planar features. Second, extracted features are then used to match and estimate pose between consecutive LiDAR scans at scan rate. Pose estimations are further refined at a lower rate by registering those features to a global map. Finally, both pose estimations are fused to give the final pose estimation. Loop closure detection is also executed to guarantee a consistent global map. When the mapping process is done, the final global map is saved for later use.

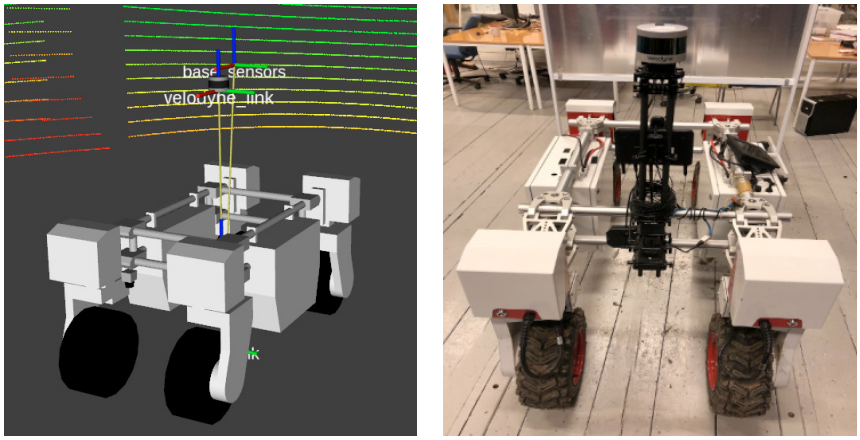
For localization in a pre-built 3D map, we iteratively perform 3D-3D scan matching between LiDAR scan and the 3D map using the NDT representation of the map. Details of mapping and localizing procedure are further discussed in the next section.

4. 3D LIDAR MAPPING

4.1 Data Preprocessing

The original LOAM method by Zhang and Singh (2014, 2017) works well in indoor environments. The authors also confirmed that feature matching is less reliable in outdoor environments due to worse feature extraction in Zhang and Singh (2017). We adopt the approach in Shan and Englot (2018); Bogoslavskyi and Stachniss (2017) to preprocess the raw point cloud data before extracting its features. In particular, the ground points are first removed from the point cloud and the remaining points are segmented into clusters where each cluster contains points of one object. The whole segmentation process is based on the projected range image from the raw 3D point cloud for fast performance. We notice that Shan and Englot (2018) employs the ground removal strategy from Himmelsbach et al. (2010) and require a heuristic predefined number of ground scans to perform ground detection. We find that the ground removal method by Bogoslavskyi and Stachniss (2017) is more robust and implement this approach.

Let $\mathcal{P}_k$, $k \in Z^+$ be set of point cloud at measurement k. After preprocessing, a set of ground points G_k and non-ground points Q_k ($G_k, Q_k \subset \mathcal{P}_k$) are obtained for feature extraction. Notice that, G_k, Q_k also contain labels for their points, i.e, ground label for ground points and unique label for each cluster and its points. We also eliminate clusters containing less than forty points. The idea of separating and labelling points is to further improve the feature matching process by matching only points with corresponding labels. For example, ground points are *never* used to match with edge features, which most likely come from non-ground points.



(a) Simulated robot (b) Real robot

Fig. 1. Hardware system overview.

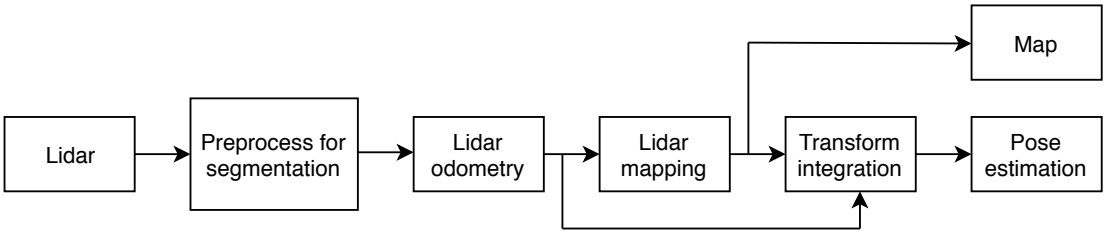


Fig. 2. Block diagram of map building.

4.2 Odometry Estimation and Mapping

The LiDAR odometry estimation process is executed in the following order.

First, we extract features from the currently received LiDAR scan $\mathcal{P}_k$. Following Zhang and Singh (2017), we also use a threshold to identify edge and planar features. However, to avoid iterating through 3D points, we perform this process using the projected range image as in Shan and Englot (2018); Bogoslavskyi and Stachniss (2017). Let S be the set of all points p_i on the same row of the range image of $\mathcal{P}_k$. The roughness c of p_i is evaluated in Eq.1, where $\|\cdot\|$ is the Euclidean distance and $|\cdot|$ is the number of points:

$$c = \frac{\left\| \sum_{j \in S, j \neq i} (r_j - r_i) \right\|}{|S| \cdot \|r_i\|} \tag{1}$$

The point p_i is classified as edge feature if its roughness c score is greater than a threshold, or else it is considered as planar feature. Let $\mathcal{E}_k, \mathcal{H}_k$ be the sets of all extracted edge and planar features, respectively. Zhang and Singh (2017) performs several condition checks to reject outliers feature points for scan matching. In contrast, by leveraging the label associated with each point, we still can ensure reliable scan matching result between scan $\mathcal{P}_k$ and $\mathcal{P}_{k-1}$ as follows. For each type of extracted features, we select a small subset of edge features $E_k, E_k \subset \mathcal{E}_k$ with maximum c score a small subset of planar features $H_k, H_k \subset \mathcal{H}_k$ with minimum c score. Then for finding correspondences, we only match points from E_k with points of the same label from $\mathcal{E}_{k-1}$ and similarly for H_k and $\mathcal{H}_{k-1}$.

Second, after finding the correspondences of the feature points, the distance between a point in the k^{th} scan and its correspondence is used to estimate the LiDAR motion, denoted as

$$\begin{aligned} \mathbf{x}_k &= [\mathbf{R}, \mathbf{T}] \\ \mathbf{T}_k &= [t_x, t_y, t_z]^T \\ \mathbf{R}_k &= [roll, pitch, yaw]^T \end{aligned} \tag{2}$$

where $\mathbf{T}_k$ and $\mathbf{R}_k$ is translational and rotational part, respectively. Stacking all the equations describe the geometric relationship between an edge points p_i and its corresponding edge line, we have:

$$f_{\mathcal{E}}(\mathbf{x}_{k,i}) = d_{\mathcal{E}}, i \in \mathcal{E}_k \tag{3}$$

Similarly, we can obtain another set of equations for planar points and their corresponding planar patches:

$$f_{\mathcal{H}}(\mathbf{x}_{k,i}) = d_{\mathcal{H}}, i \in \mathcal{H}_k \tag{4}$$

The detail derivation of $d_{\mathcal{E}}, d_{\mathcal{H}}$ is exactly as in Zhang and Singh (2017) and omitted here for brevity. While in Zhang and Singh (2017), the authors combine $f_{\mathcal{E}}(\mathbf{x}_{k,i}), f_{\mathcal{H}}(\mathbf{x}_{k,i})$ into one system of non linear equations and apply the Lavenberg-Marquardt (LM) method to solve it, we follow the approach in Shan and Englot (2018) to obtain the motion estimation in a more efficient way. We first solve Eq.4 using the same LM method. Notice that Eq.4 estimates transformation between planar patches, the estimation of roll, pitch angles and translation in z direction is more accurately estimated than other components. We then use the three components as constraints to solve Eq.3. Again, for edge lines in Eq.3, translation in x, y and yaw angles are estimated more robustly and we selectively choose these components. Finally, we fuse these six components together to achieve the final 6-DOF pose estimation.

Let $\mathcal{G}_{k-1}$ be the set of point clouds in the global map accumulated up to the LiDAR $(k-1)^{th}$ scan . We implement the similar method in Zhang and Singh (2017)

to match the points in $\mathcal{E}_k, \mathcal{H}_k$ to $\mathcal{G}_{k-1}$ to further refine the pose estimation. Readers are referred to Zhang and Singh (2017) for the details. We notice the difference here is that we explicitly aim for a consistent and reusable large-scale map, not just accurate odometry estimation. Hence, we implement a pose-graph optimization with loop closure detection in Dellaert and Kaess (2017); Kaess et al. (2012) to obtain the fine map. Specifically, the pose obtained in the odometry estimation step is considered a node in the graph. A loop is detected by matching between $\mathcal{E}_k, \mathcal{H}_k$ and $\mathcal{E}_{k-1}, \mathcal{H}_{k-1}$. If a match is found, it is added as a new constraint to the graph. The graph is efficiently updated using iSAM2 library in Kaess et al. (2012).

5. LOCALIZATION IN A PRIOR 3D MAP

Given a 3D LiDAR map built in the previous section, the robot can estimate its pose as the sensor ego-motion. We adopt an NDT-based scan matching for localization similar to Sakai et al. (2017). In comparison to the ICP method, 3D NDT scan matching is faster and at least as accurate as the state-of-the-art ICP method Stoyanov et al. (2012). Instead of performing heavy computation scan matching by iterating through every point, the robot only needs to compare between the much smaller estimated Gaussian components, which represent the map and the received LiDAR scans. In addition, the robot might experience abrupt changes on uneven terrain, which in turn causes a large displacement between consecutive scans.

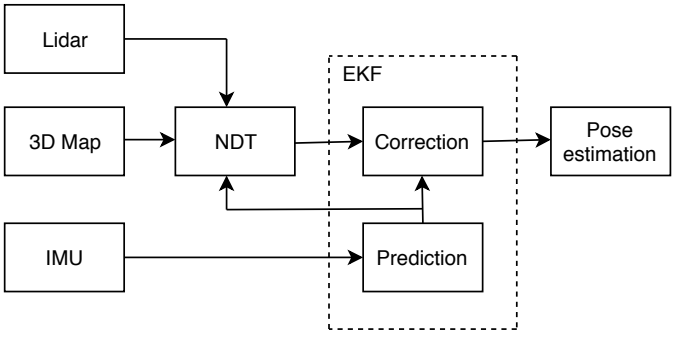


Fig. 3. Block diagram of localization system.

Let $\mathbf{x}_t = [\mathbf{p}_t, \mathbf{q}_t, \mathbf{v}_t, \mathbf{b}_t^\omega]^T$ be the state vector at time t that we need to estimate, where $\mathbf{p}_t$ is the position, $\mathbf{q}_t$ is the rotation vector in quaternion representation, $\mathbf{v}_t$ is the velocity and $\mathbf{b}_t^\omega$ is a constant bias for raw gyroscope measurements $\hat{\boldsymbol{\omega}}_t$ from an IMU, that is rigidly attached to the LiDAR sensor frame. Since the robot normally runs at low speed, we can assume a constant translational velocity for the motion model. Employing a standard Extended Kalman filter, the prediction step is defined as follows:

$$\begin{aligned} \mathbf{x}_t &= [\mathbf{p}_{t-1} + \mathbf{v}_{t-1} \cdot \delta t, \mathbf{q}_{t-1} \cdot \delta\mathbf{q}_t, \mathbf{v}_{t-1}, \mathbf{b}_{t-1}^\omega]^T \\ \delta\mathbf{q}_t &= [\frac{\delta t}{2}\omega_t^x, \frac{\delta t}{2}\omega_t^y, \frac{\delta t}{2}\omega_t^z, 1] \\ \boldsymbol{\omega}_t &= \hat{\boldsymbol{\omega}}_t - \mathbf{b}_{t-1}^\omega \end{aligned} \tag{5}$$

where δt is a time step, $\delta\mathbf{q}_t$ is the rotation during δt with the bias-compensated angular velocity $\boldsymbol{\omega}_t$. The predicted pose $\mathbf{x}_t, \mathbf{q}_t$ are used as initial guess for the NDT process to match the observed point cloud to the global map. The correction step then uses the NDT estimation result to correct the final state estimation.

6. EXPERIMENTS

We validate our proposed system on both simulated and real datasets. Here, we provide quantitative evaluations on: position drift while mapping, relocalization on previously built 3D LiDAR map and map quality comparison.

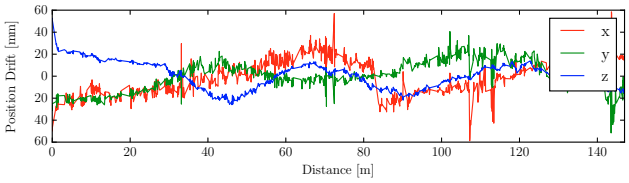


Fig. 4. Position drift when mapping in simulation.

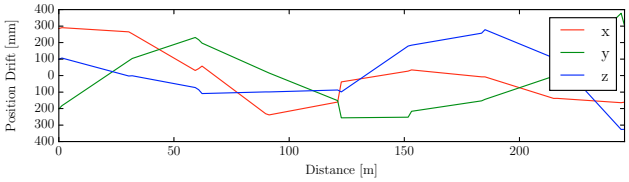


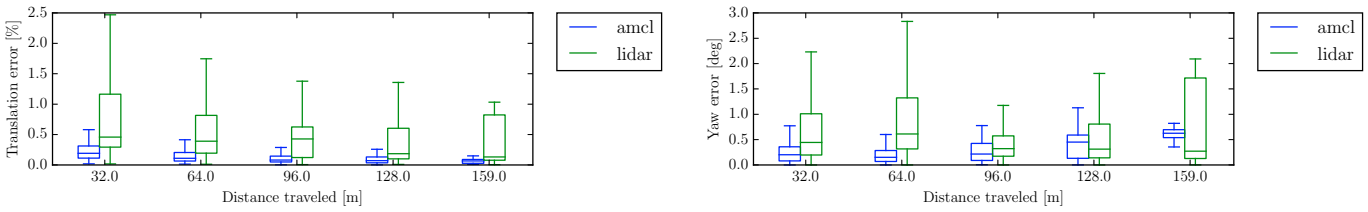
Fig. 5. Position drift when mapping in real scene.

We first validate the proposed system using the simulation built on *gazebo* [2] for our project [3]. The simulated scene consists of two polytunnels and food processing storage. The simulated Thorvald robot is configured to physically match the real one. It is equipped with a simulated Velodyne VLP-16 [4] and a 2D LiDAR Hokuyo. Currently, 2D LiDAR with *gmapping* SLAM [5] is used for building map. The de-facto AMCL [6] is used for localization in a pre-built 2D map. Hence, we directly compare the localization results from two different sensor modalities and show that we can achieve comparable or better results. The ground truth is taken from *gazebo*.

In the simulation test, the robot is first manually driven around the scene while both *gmapping* and the proposed 3D LiDAR mapping are running to build the 2D and 3D map of the scene, respectively. The built maps are then saved for localization test. The 2D maps are omitted due to space constraint. After building maps, the robot is again driven manually through the scene using the previously built maps for localization. Both *AMCL* and the proposed localization method are running to estimate the robot pose. Both estimation results are recorded and analyzed following Zhang and Scaramuzza (2018). Position drift when mapping with LiDAR is shown in Fig.4. The relative errors in translation and rotation (yaw) are shown in Fig.6a,6b, respectively. The relative errors also show a consistently low median translational error of the proposed method (less than 0.5%). For relative rotation (yaw) errors, the proposed method shows a smaller median error in comparison with *amcl* for long trajectories.

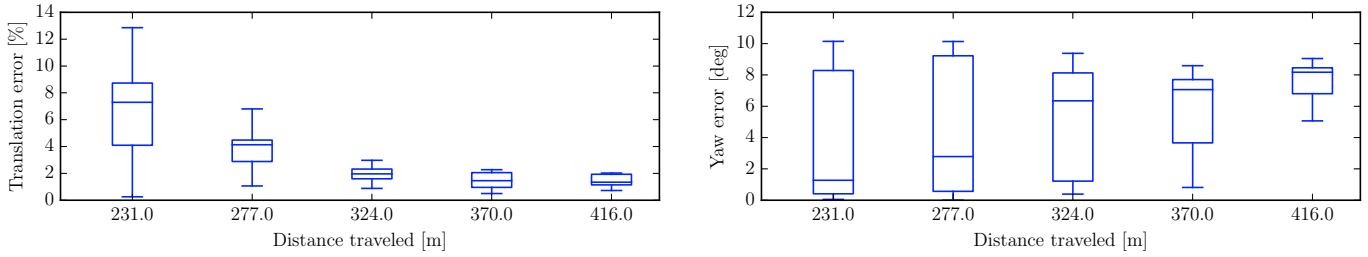
We conduct another test with a real dataset. The robot (Fig.1b) is driven manually around our campus. Starting

[2] http://gazebosim.org/
[3] https://rasberryproject.com/
[4] https://github.com/LCAS/velodyne_simulator
[5] http://wiki.ros.org/gmapping
[6] http://wiki.ros.org/amcl



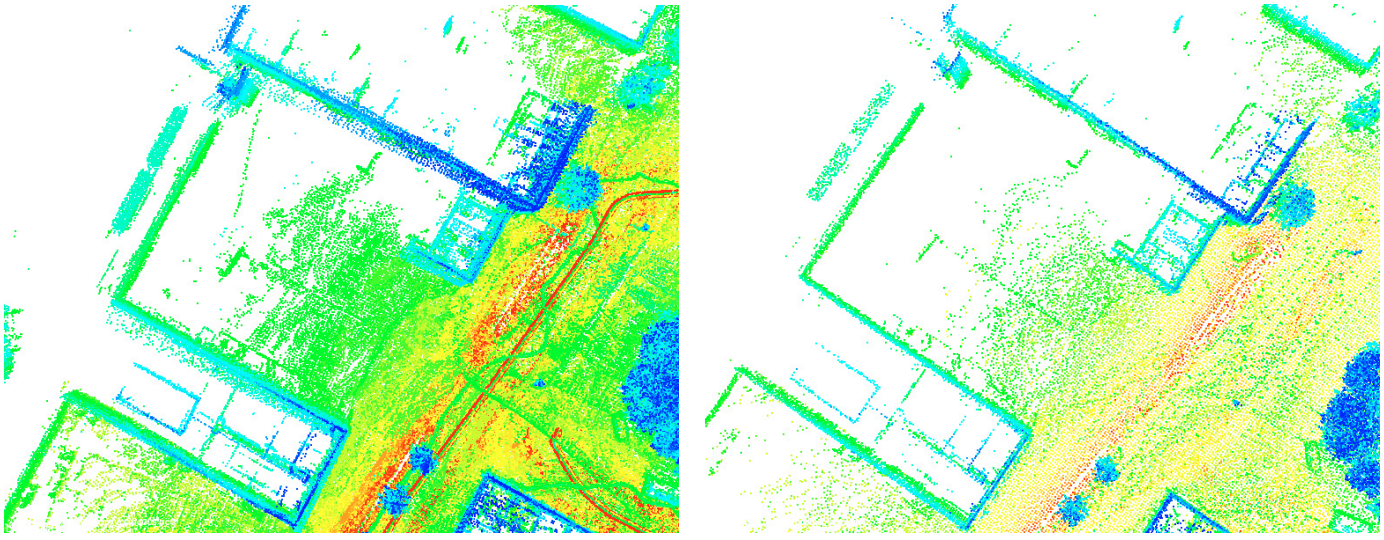
(a) Translation error (b) Rotation error

Fig. 6. Quantitative localization comparison between *amcl* and proposed method in simulation. The ground truth is taken from *gazebo*.



(a) Translation error (b) Rotation error

Fig. 7. Quantitative localization comparison of proposed method with a real dataset.



(a) Built map by LOAM (b) Built map by our method

Fig. 8. Qualitative comparison of built maps. Color indicates intensities for viewing. Notice the difference of the building walls. Our method produces sharper map than LOAM's. Best viewed in color.

in front of our lab, which is served as a storage, the robot moves to our mockup polytunnel and back. The total trajectory is 500 meters. The ground truth in this test is obtained via a RTK-GNSS Septentrio AsteRx4 system. Notice that, we solely use the RTK-GNSS for ground truth comparison. To further challenge the proposed localization method, only *half* of the dataset is used for mapping. For localization in a built 3D map, the robot uses the whole dataset, in which half of the dataset contains LiDAR scans from the opposite moving direction when mapping. This mimics a scenario where we want to perform a fast mapping process and the robot can reliably use the built map for localization. We also achieve a low drift in position as shown in Fig.5. The relative pose error is shown in Fig.7. The robot achieves a small median translation error ($< 2\%$) for the whole trajectory. However, we encounter accumulating drift in rotation estimation, which is contributed by our assumption of constant bias of angular velocity and the inconsistency of the EKF.

Finally, we compare the quality of the built map between our proposed method and the original LOAM. We follow Droeschel et al. (2014) to calculate the *mean map entropy* (MME) from the mapped points $\mathcal{P} = \{p_1, ..., p_n\}$. The mean map entropy $H(\mathcal{P})$ is used as the *crispness/sharpness* metric of the map. A map with lower entropy has higher quality. The mean map entropy is defined in Eq.6.

$$h(p_k) = \frac{1}{2} \ln |2\pi e \Sigma(p_k)|$$
$$H(\mathcal{P}) = \frac{1}{n} \sum_{k=1}^{n} h(p_k) \quad (6)$$

where $h(p_k)$ is the entropy of the mapped point p_k, $\Sigma(p_k)$ is the sample covariance of the mapped point p_k in a local radius $r = 0.3m$ around p_k, and $\mathrm{H}(\mathcal{P})$ is averaged over all n mapped points.

The 3D maps built by LOAM and our method have the values of entropy: -0.19 and -0.22, respectively. The maps size is 27.7 MB and 8.2 MB by LOAM and ours, respectively. Our method produces a sharper map with less memory consumption for storage.

We illustrate the differences in quality of the built maps by two methods in Fig.8. LOAM retains more points in its map but the quality of the map is lower than ours.

The video of experiments is available online:
https://youtu.be/05sTYF8AKaY

7. CONCLUSIONS

In this work, we propose a complete online 3D mapping and localization system for intelligent agricultural robots. Existing methods, such as the state-of-the-art LOAM, primarily focus on odometry estimation in urban scenarios. We provide an additional localization method to make use of an accurate 3D built map, which is vital for an agricultural robot to work on a large scale farm without remapping before operating. The proposed system is tested using simulated and real datasets.

We notice, that by applying segmentation on input point clouds, we achieve more robust and better point cloud registration. Hence, future work involves further exploitation of point cloud segmentation to deal with dynamic environment. In addition, we plan to improve the localization system to further reduce drifts in rotation estimation caused by the inconsistency of the EKF.

8. ACKNOWLEDGMENT

We would like to thank Tixiao Shan and Igor Bogoslavskyi for fruitful discussions and code base.

REFERENCES

Albani, D., Nardi, D., and Trianni, V. (2017). Field coverage and weed mapping by uav swarms. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 4319–4325. IEEE.

Bogoslavskyi, I. and Stachniss, C. (2017). Efficient online segmentation for sparse 3d laser scans. *PFG – Journal of Photogrammetry, Remote Sensing and Geoinformation Science*, 1–12.

Dellaert, F. and Kaess, M. (2017). Factor graphs for robot perception. *Foundations and Trends in Robotics, FNT*, 6(1-2), 1–139. http://dx.doi.org/10.1561/2300000043.

Droeschel, D., Stückler, J., and Behnke, S. (2014). Local multi-resolution representation for 6d motion estimation and mapping with a continuously rotating 3d laser scanner. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, 5221–5226. IEEE.

Grimstad, L. and From, P.J. (2017). Thorvald ii - a modular and re-configurable agricultural robot. In *IFAC 2017 World Congress*.

Himmelsbach, M., Hundelshausen, F.V., and Wuensche, H.J. (2010). Fast segmentation of 3d point clouds for ground vehicles. In *Intelligent Vehicles Symposium (IV), 2010 IEEE*, 560–565. IEEE.

Kaess, M., Johannsson, H., Roberts, R., Ila, V., Leonard, J.J., and Dellaert, F. (2012). isam2: Incremental smoothing and mapping using the bayes tree. *The International Journal of Robotics Research*, 31(2), 216–235.

Khanna, R., Möller, M., Pfeifer, J., Liebisch, F., Walter, A., and Siegwart, R. (2015). Beyond point clouds-3d mapping and field parameter measurements using uavs. In *2015 IEEE 20th conference on emerging technologies & factory automation (ETFA)*, 1–4. IEEE.

Kohlbrecher, S., Von Stryk, O., Meyer, J., and Klingauf, U. (2011). A flexible and scalable slam system with full 3d motion estimation. In *2011 IEEE International Symposium on Safety, Security, and Rescue Robotics*, 155–160. IEEE.

Magnusson, M., Lilienthal, A., and Duckett, T. (2007). Scan registration for autonomous mining vehicles using 3d-ndt. *Journal of Field Robotics*, 24(10), 803–827.

Nived, C., Philipp, L., Thomas, L., and Behnke, S. (2019). Robot localization based on aerial images for precision agriculture tasks in crop fields. In *Robotics and Automation (ICRA), 2019 IEEE International Conference on*. IEEE.

Popovi, M., Vidal-Calleja, T., Hitz, G., Sa, I., Siegwart, R., and Nieto, J. (2017). Multiresolution mapping and informative path planning for uav-based terrain monitoring. In *Intelligent Robots and Systems (IROS), 2017 IEEE International Conference on*. Vancouver.

Sakai, T., Koide, K., Miura, J., and Oishi, S. (2017). Large-scale 3d outdoor mapping and on-line localization using 3d-2d matching. In *System Integration (SII), 2017 IEEE/SICE International Symposium on*, 829–834. IEEE.

Shan, T. and Englot, B. (2018). Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 4758–4765. IEEE.

Stoyanov, T., Magnusson, M., Andreasson, H., and Lilienthal, A.J. (2012). Fast and accurate scan registration through minimization of the distance between compact 3d ndt representations. *The International Journal of Robotics Research*, 31(12), 1377–1393. doi: 10.1177/0278364912460895.

Zhang, J. and Singh, S. (2014). Loam: Lidar odometry and mapping in real-time. In *Robotics: Science and Systems*, volume 2, 9.

Zhang, J. and Singh, S. (2017). Low-drift and real-time lidar odometry and mapping. *Autonomous Robots*, 41(2), 401–416.

Zhang, Z. and Scaramuzza, D. (2018). A tutorial on quantitative trajectory evaluation for visual(-inertial) odometry. In *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*.