LiDAR-LiDAR Extrinsic Calibration Approaches for Husky UGV on DARPA SubT Challenge Course

Phil Cotter and Mike Schoder

***Abstract*—Extrinsic calibration of two or more robot- or vehicle-mounted sensors is a well-studied problem critical to the accurate navigation, localization, and pose estimation of autonomous systems. Estimating the static transformation between sensors is known as hand-eye calibration. Here, we compare implementations of four established methods on a dataset containing multiple LiDAR scans and odometry estimates from an autonomous test vehicle developed by NASA JPL as part of the DARPA Subterranean Challenge. We find that the methods using closed form solutions and joint optimization of odometry-based trajectory and sensor extrinsics perform best in general. These methods outperform the point-based methods implemented, which estimate extrinsic calibration by aligning point clouds. However, no method tested performed as well as the prior calibration estimate, where sensors were hand-tuned by JPL staff to minimize rotation error. Only the joint nonlinear graph optimization method performed better in translation metrics, but the marginal improvement over the base case was negligible. Despite this, we still assess valid applications of the two top-performing hand-eye calibration methods tested here. Their simplicity and speed enable potential use cases for JPL for online re-calibration or fault-tolerance checking.**

Fig. 1. Husky UGV used by NASA JPL's CoSTAR team for entry in the DARPA Sub-T challenge. Vehicle has three lidar sensors: one "main" sensor on top, and "front" and "rear" sensors which are angled downward to provide maximum coverage. Source: [1]

I. Introduction

COMPETING in the latest DARPA Subterranean ("SubT") challenge, team CoSTAR, headed by NASA JPL in conjunction with several other universities, has developed an autonomous vehicle solution for the purpose of navigating and mapping underground urban environments, such as manmade caves and tunnels. Among other sensors, the Husky unmanned ground vehicle (UGV) is equipped with three 16-channel velodyne LiDAR sensors: one centered on top of the vehicle ("main") and one each oriented to the front and rear, both angled down toward the ground (Fig. 1). The Husky vehicles rely on a continuous stream of LiDAR sensor measurements for localization and mapping, and function as primary inputs to downstream odometry and SLAM systems which perform mission-essential tasks.

A. Problem Description

In order to perform accurate SLAM and navigate correctly, calibration of mounted LiDAR sensors is essential. LiDAR sensors are mounted rigidly to the vehicle, but mounts are imperfect and the harsh conditions to which these vehicles are exposed results in jostling of sensor components which may perturb this alignment. JPL's current method for calibrating LiDAR sensor extrinsics is to use the CAD-based measurements to estimate the rigid translation between sensors, and to perform estimation of the rotation by hand, manually adjusting sensors by visualizing static overlay of intersecting point clouds until the physical rotation closely matches the programmed static rotation. This process is not only time consuming, but is subject to error and must typically be performed by an experienced engineer. Furthermore, this form of calibration cannot be done online if sensors fall out of alignment during execution.

B. Solution Approach

To address these concerns, we conducted a survey of different sensor calibration techniques, and present results of four implementations on a dataset containing LiDAR point clouds and independent odometry estimates for three LiDAR sensors mounted to a single Husky vehicle while the vehicle is in motion over a three minute period. The dataset is captured in the common ROSbag format, and odometry estimates of relative pose for each sensor are given as the result of a LiDAR-only odometry pipeline running on the vehicle. Because LiDAR odometry estimates are computed independently for each sensor and each LiDAR uses its own internal clock, there is also a requirement to synchronize pose estimates before calibrating, and common techniques for this task are addressed here as well.

II. Previous Work

Methods for calibrating dual LiDAR sensors fall into two categories. So-called traditional methods use odometry estimates of each sensor to solve the hand-eye problem, which

refers to the original formulation in which the rigid transform between a wrist-mounted camera and a robot's end-effector pose are estimated. Point-based methods comprise another grouping that is more specific to LiDAR sensors, where overlapping portions of point clouds from separate sensors are matched in order to minimize a cumulative distance between selected points.

A. Decoupled Solutions

The earliest solutions to hand-eye calibration were published in the late 1980s. Work by Tsai and Lenz [2] and Shiu and Ahmad [3] take similar approaches of decoupling the calibration problem into separate rotation and translation problems, using axis-angle representation of relative poses to first minimize rotation error and then solve for translation using linear least squares. A subsequent comparison of approaches by Wang [4] finds that the Tsai-Lenz approach is more accurate and more efficient, and this method has become a de-facto standard for calibration included in several commercial and open-source libraries [5][6]. The work by Park and Martin [7] uses a solution based on Lie theory to further simplify computational requirements for the decoupled solution. Finally, Horaud and Dornaika introduce another version of the decoupled solution using quaternion representation [8].

B. Simultaneous Solutions

While the decoupled formulations referenced above are simple and have closed-form linear solutions, the sequential rotation-then-translation approach means that solutions are not necessarily optimal overall. Several approaches to solve for rotation and translation simultaneously have been proposed. Daniilidis [9] uses a dual-quaternion parameterization based on screw-theory to simultaneously estimate rotation and translation using singular value decomposition in closed-form. Andreff et al. [10] make use of the Kronecker product to transform the simultaneous hand-eye calibration problem into a linear system, enabling a joint closed-form solution. Furer et al. [11] combine the approach of Daniilidis and other iterative methods to produce a robust end-to-end method.

C. Iterative Solutions

The simultaneous closed-form approaches involve complex derivations, and more recent literature shows they lack robustness to measurement noise. Iterative approaches, where the difference between left and right sides of the hand-eye formulation are sequentially minimized, are shown to be more robust to noise. In addition to their decoupled solution, Horaud and Dornaika [8] introduce a non-linear optimization formulation to solve for the full rigid transformation jointly along with trajectory estimates using the Levenberg-Marquardt method, and provide results showing the improved accuracy and robustness as compared to their own decoupled approach. Schmidt et al. [12] extend the work of Daniilidis by using screw-theory and the dual-quaternion approach to perform pose outlier rejection, and then implement a similar non-linear optimization approach. Furer et al. use the dual-quaternion approach of Daniilidis coupled with RANSAC-based and screw-theory-based pose filtering to remove outliers in order to generate an initial calibration estimate, and then implement a Levenberg-Marquardt non-linear optimization method representing continuous time trajectories as B-splines as described in [13] as a refinement step. Finally, Koide et al. [14] introduce a formulation using pose-graph optimization to minimize sensor reprojection error.

D. Point-Cloud Based

Point cloud registration techniques can also be utilized to compute an estimate of the extrinsic sensor calibration between sensor pairs [15]. The goal is to perform pairwise registration, finding the transform that best aligns two point clouds given some degree of overlap between these point clouds. We implement two popular solutions for point cloud registration here: iterative closest point (ICP) and normal distributions transform (NDT). Many extensions and variants on these approaches also exist in literature and open-source which could also be investigated including TEASER++, IDC, pIC, and others. The NDT implementation benefits from having no reliance on upstream LiDAR odometry to compute a transformation, making it robust to noise from odometry estimation.

E. Temporal Sensor Synchronization

The ROSbags provided for each of the three sensors contain data that is not perfectly synchronized in time. This misalignment occurs because each sensor typically operates on its own internal clock. A standard method to resolve these delays is to use correlation analysis. We use a synchronization package from ETHZ's Hand-Eye Calibration Library, which establishes correlation based on angular velocity norms of sensor odometry estimates and re-samples poses to align at the same frequency. This time-synchronized data enables the implementations described below.

III. Implementation

A. Evaluation Framework

For the implementations compared in this paper, we evaluate independently the relative root mean squared (RMSE) rotation and translation error of the rigid transformation between the vehicle's main and front LiDAR sensors, for sequential poses. We use the dataset provided by JPL, which after time-alignment contains 2576 corresponding pose-pairs in each of the two sensor frames.

B. Methods Evaluated

1) Tsai-Lenz Method: We implement the Tsai-Lenz method in MATLAB using odometry estimates provided by existing onboard UGV system. Our implementation borrows from Zoran Lazarevic's open source implementation [16]. We run multiple tests using different numbers of relative pose combinations to evaluate the error dependence. We alternatively consider every possible combination of pose pairs (3M pairs),

every pair of sequential poses (2575 pairs), and every fifth (515 pairs), 25th (102 pairs), and 125th (20 poses) sequential pose pairs.

2) ETHZ Hand-Eye: ETHZ's Autonomous Systems Lab maintains an open source library based upon the work of Furrer et al. [11], which includes modules to perform trajectory time alignment and interpolation, and estimate hand-eye calibration transforms using a combination of existing solutions. Time alignment is performed by resampling poses using linear interpolation and then correlating angular velocity norms of each pose to compute the delay offset. The approach here further performs outlier filtering by discarding the top 1% of angular velocity norms and applying a low-pass kernel in order to be more robust to outliers. We also use this time alignment method to pre-process data for the Tsai-Lenz and ICP methods tested here.

Two separate modules exist to perform hand-eye calibration. First is an implementation of the dual-quaternion approach described by Daniilidis [9], with the addition of outlier removal, which is done by first applying pose filtering followed by RANSAC. Pose filtering relies on computing the dot product of screw-axis of each pose, and removing pose pairs with a value above a given threshold, which indicates that the poses are close to parallel and provide insufficient information. The calibration estimate from this approach is then used as an initial estimate in a non-linear batch optimization process which jointly refines trajectory and extrinsic calibration estimates to provide a more accurate and robust solution. We further experiment by using different approaches as initial estimates into this non-linear optimization, including the provided base-case estimate and the best estimate from the Tsai-Lenz approach.

3) LiDAR-Align: In the first of two point-cloud based approaches, we implemented ETHZ's LiDAR-Align method, which uses an ICP-based method [17]. The implementation requires a specified initial transformation estimate. ICP then iteratively optimizes the transform required to align two point clouds by minimizing the sum of the squared distance between corresponding points in the two clouds. Because it is point-based and does not take the surface of the reference cloud into consideration, matching corresponding points is error-prone and computationally expensive. ETHZ's LiDAR-Align implementation also requires highly non-planar motions in order to generate a high-quality transform. Because of the relatively planar nature of this data set, we expected this implementation to perform poorly relative to other methods.

4) Normal Distributions Transform: We also tested a Normal Distributions Transform (NDT) registration method using an implementation from the Point Cloud Library (PCL) [18]. NDT, rather than matching point-to-point correspondences, subdivides the point cloud's space into voxels and then transforms the reference cloud into a piecewise-smooth set of normal distributions represented in each voxel [15]. This approach is less computationally expensive and error prone than the ICP implementation described above. The implementation used a manual sampling of time steps wherein the Husky was stationary and then selected the transformation that resulted in the lowest Euclidean distance fitness score between the transformed clouds. We chose multiple points in time wherein the robot was translationally and rotationally static, and compared the transforms generated from NDT at these discrete points in time. Point cloud alignment could be qualitatively compared using PCL's visualization functionality. The transform resulting in the lowest Euclidean distance-based fitness score was selected as the representative transformation from NDT to compare with other approaches.

IV. Results

A. Measuring Performance

In comparing the performance of these implementations, we use both a quantitative and qualitative approach. Quantitatively, we calculate the RMSE for rotation and translation separately for relative poses at successive timesteps using a given extrinsic calibration. This has the effect of comparing the reprojection error induced by each calibration transformation. Qualitatively, we use the OctoMap package to visualize the resultant map from each transform, observing if and when misalignment occurs [19].

B. Comparison of Implementations

1) Tsai-Lenz Method: Results from our implementation of the Tsai-Lenz method converge closely toward the base case static transformation, but do not perform better, even with every possible combination of pose pairs considered (Table I). Several possible explanations exist: the base case may be very close to optimal, or measurement noise may be corrupting the linear-least squares fit. It is also worth noting the RMSE dependence on the number of pose pairs uses for the solution. At 515 pose pairs (every 25th relative pose) the increase in error is still very low, and deviation is not noticeable until using 100 or fewer poses, which can be seen in visualization of transformed trajectories in Fig. 2. This suggests that applying some method of pose filtering for outlier removal and decreasing the problem size may be worthwhile, particularly for online applications of this algorithm, which scales in On^3 time (for our implementation). Table I provides experimental runtimes, measured on an Intel i7 quad-core 2.7GHz CPU with 16GB RAM. These improvements are addressed (with the addition of some implementation complexity) in the next section.

TABLE I
Error results for Tsai-Lenz calibration approach using different numbers of pose pairs

No. Rel. Poses	Trans RMSE [m]	Rot RMSE [deg]	Runtime [ms]
Base	0.010169	0.399845	–
3,314,025	0.010206	0.399978	107.6
2,575	0.010758	0.400718	1.038
515	0.010202	0.400245	0.7913
102	0.010731	0.401441	0.7528
20	0.049473	0.435163	0.7392

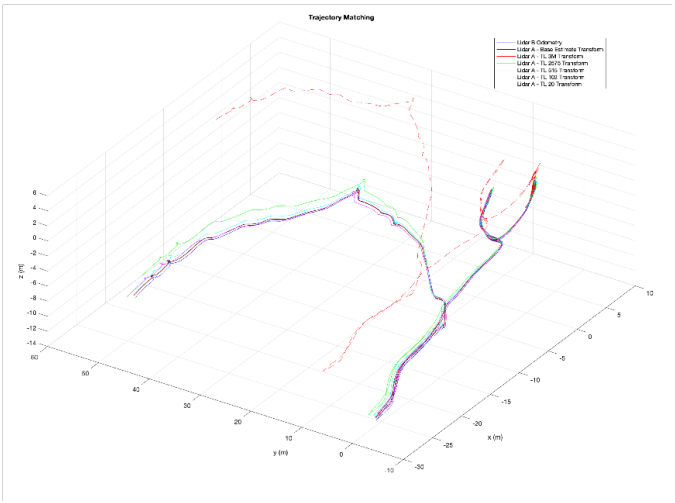

Fig. 2. Comparison of transformed "main" sensor trajectory to the "front" trajectory after applying estimated transformation computed from Tsai-Lenz approach using different numbers of pose-pairs. Solution does not begin to deviate substantially from the best case until we use 100 or fewer pairs, at which point translation deviation from the base case exceeds 5%.

2) ETHZ Hand-Eye Library: Using the ETHZ package, we first implemented the dual-quaternion approach, including outlier removal via RANSAC and pose filtering as previously described. Results are given in Table II, and we see that the approach does not perform as well as either the base case or the Tsai-Lenz approach. This is possibly caused by overly aggressive pose filtering; indeed a large number of pose pairs are excluded by the screw-axis dot product threshold. We use the results of the dual-quaternion approach, along with the other two existing estimates, as initial guesses for the joint non-linear optimization using Levenberg-Marquardt, which is implemented in a separate module. We can see that the optimization converges to nearly identical solutions from all guesses, and that the solution is extremely close to the base case solution. This supports the hypothesis that the base case is near optimal. The optimization implementation is also simple and easy to use, and may be useful as a standalone module for calibration refinement, or as a final step in a pipeline which takes an initial guess from any one of a variety of methods.

TABLE II
COMPARISON OF TRAJECTORY RMSE USING ETHZ HAND-EYE PACKAGE

	INITIAL GUESS		OPTIMIZATION	
INIT. GUESS	TRANS	ROT	TRANS	ROT
BASE	0.010169	0.399845	0.010163	0.399924
TSAI-LENZ	0.010206	0.399978	0.010173	0.399925
DQ + RANSAC	0.014576	0.399878	0.010174	0.399925

3) LiDAR-Align Library: The resulting extrinsic calibration solution produced by the LiDAR-Align approach produced the poorest results of all four implementations. Both the translational and rotational RMSE were substantially higher than all of the closed-form and jointly-optimized approaches, as seen in Table III. ETHZ's documentation warns that "accurate results require highly non-planar motions." Because this data set is generally planar, it does not provide sufficient information in the z-direction, leading to poor alignment in this direction which partially explains this increased error. An alternative ICP implementation could better address the planar nature of this dataset and provide a better performing calibration. Fig. 5 provides a qualitative comparison using OctoMap of the LiDAR-Align calibration versus the ETHZ Batch Optimization calibration. The visible misalignment underscores this implementation's poorer performance on this dataset.

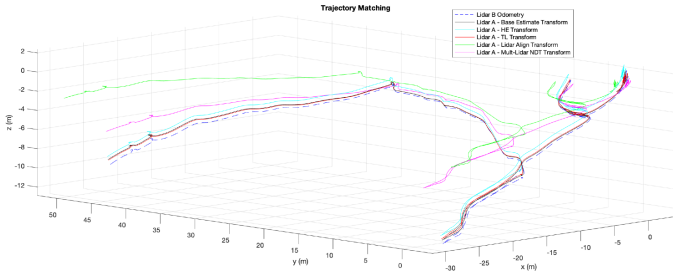

Fig. 3. Comparison of transformed trajectories for different calibration approaches. LiDAR A (the vehicle "main" sensor) is transformed into the coordinate frame of LiDAR B ("front") using the estimated rigid transformation.

4) Multi-LiDAR NDT Library: The NDT implementation produced an extrinsic calibration that outperformed the LiDAR-Align approach, but still under-performed the closed-form and jointly-optimized solutions. Resulting representative fitness scores for pairwise clouds can be seen in Table IV. Because this implementation uses a pair of selected point clouds, the result is reliant on only a single sample in time. It is notable that despite this sampling approach, a relatively good transformation could be computed (see the reasonable RMSE values in Table III). NDT was also observed to be highly reliant on the initial estimate passed to the solver so that the optimization would arrive at a sensible transformation and not converge to an undesirable local optimum. The package is easy to implement and provides convenient visualization ability which can be used to qualitatively confirm the viability of the solution. A more robust implementation of this NDT approach would optimize over the entire time series, thus making it less reliant on a single pairwise computation. This more robust NDT approach could serve as a complement to the closed-form and jointly-optimized approaches discussed earlier given that it relies on an independent source of data.

TABLE III
RMSE ERROR COMPARISONS FOR BEST VERSION OF EACH APPROACH IMPLEMENTED

METHOD	TRANS RMSE	ROT RMSE
BASE	0.010169	0.399845
TSAI-LENZ	0.010206	0.399978
ETHZ DQ + RANSAC	0.014576	0.399878
ETHZ BATCH OPTIM.	0.010163	0.399924
ETHZ LIDAR-ALIGN	0.042729	0.488196
MULTI-LIDAR NDT	0.019603	0.443844

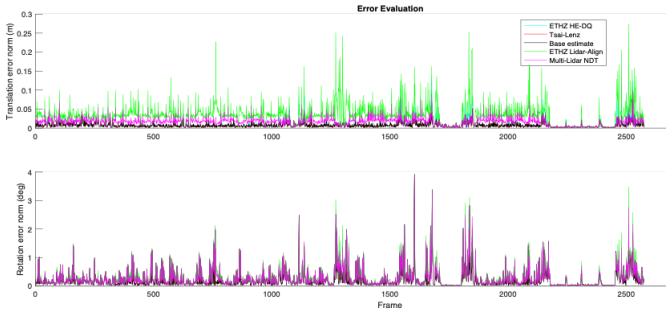

Fig. 4. Rotation and translation RMSE comparison of sequential relative poses for different approaches implemented.

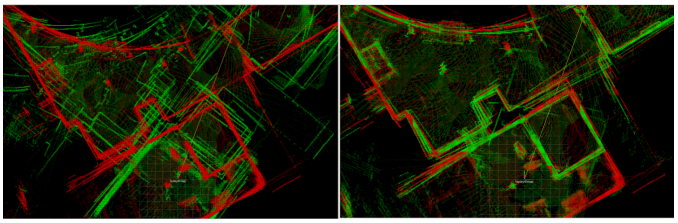

Fig. 5. Comparison of point cloud maps overlaid from both LiDAR sensors using the ETHZ LiDAR-Align implementation (left) and ETHZ Batch Optimization (right) methods to estimate extrinsic sensor calibration. Visualization is performed using the OctoMap package [19]

TABLE IV
EUCLIDEAN DISTANCE FITNESS SCORE FOR NDT SOLUTIONS

BAG TIME	FITNESS SCORE
1582327002	0.40985
1582327080	1.35433
1582327131	1.06633

V. CONCLUSION

This project investigated existing methods for LiDAR-LiDAR sensor calibration on a Husky UGV navigating the Urban circuit of DARPA's SubT challenge. We investigated closed form solutions, joint-optimization approaches, and point-cloud based methods. Ultimately, none of the approaches implemented improved significantly over the manually-calibrated estimated generated by JPL. However, the approaches implemented here could be utilized for a more consistent, automated, online approach to extrinsic sensor calibration. Implementing one of these solutions online to validate current calibration or perform an expedient re-calibration if required, would enable an autonomous robot to produce more reliable downstream odometry and SLAM computations given the robot's extreme operating environment. The ability to perform online LiDAR-LiDAR calibration could enhance the robustness of the Husky UGV entrant in future iterations of the SubT challenge.

ACKNOWLEDGMENT

The authors would like to thank Prof Luca Carlone, the Fall 20 VNAV staff, and the SubT team at NASA JPL for their support and guidance.

REFERENCES

[1] K. Ebadi, Y. Chang, M. Palieri, A. Stephens, A. Hatteland, E. Heiden, A. Thakur, N. Funabiki, B. Morrell, S. Wood, L. Carlone, and A.-a. Agha-mohammadi. LAMP: Large-Scale Autonomous Mapping and Positioning for Exploration of Perceptually-Degraded Subterranean Environments. [Online]. Available: http://arxiv.org/abs/2003.01744

[2] R. Tsai and R. Lenz, "A new technique for fully autonomous and efficient 3D robotics hand/eye calibration," vol. 5, no. 3, pp. 345–358. [Online]. Available: http://ieeexplore.ieee.org/document/34770/

[3] Y. Shiu and S. Ahmad, "Calibration of wrist-mounted robotic sensors by solving homogeneous transform equations of the form AX=XB," vol. 5, no. 1, pp. 16–29, Feb./1989. [Online]. Available: http://ieeexplore.ieee.org/document/88014/

[4] C.-C. Wang, "Extrinsic calibration of a vision sensor mounted on a robot," vol. 8, no. 2, pp. 161–175. [Online]. Available: http://ieeexplore.ieee.org/document/134271/

[5] "OpenCV HandEyeCalibrationMethod." [Online]. Available: https://docs.opencv.org/3.4/d9/d0c/group__calib3d.html

[6] Jena-Yves Bouguet, "Camera Calibration Toolbox for Matlab," CalTech. [Online]. Available: http://www.vision.caltech.edu/bouguetj/calib_doc/

[7] F. C. Park and B. J. Martin, "Robot sensor calibration: Solving AX=XB on the Euclidean group," vol. 10, no. 5, pp. 717–721.

[8] R. Horaud and F. Dornaika, "Hand-Eye Calibration," vol. 14, no. 3, pp. 195–210. [Online]. Available: http://journals.sagepub.com/doi/10.1177/027836499501400301

[9] K. Daniilidis, "Hand-Eye Calibration Using Dual Quaternions," vol. 18, no. 3, pp. 286–298. [Online]. Available: http://journals.sagepub.com/doi/10.1177/02783649922066213

[10] N. Andreff, R. Horaud, and B. Espiau, "On-line hand-eye calibration," pp. 430–436.

[11] F. Furrer, M. Fehr, T. Novkovic, H. Sommer, I. Gilitschenski, and R. Siegwart, "Evaluation of combined time-offset estimation and hand-eye calibration on robotic datasets," in *Field and Service Robotics*, M. Hutter and R. Siegwart, Eds. Springer International Publishing, pp. 145–159.

[12] J. Schmidt, F. Vogt, and H. Niemann, "Robust Hand–Eye Calibration of an Endoscopic Surgery Robot Using Dual Quaternions," in *Pattern Recognition*, ser. Lecture Notes in Computer Science, B. Michaelis and G. Krell, Eds. Springer Berlin Heidelberg, vol. 2781, pp. 548–556. [Online]. Available: http://link.springer.com/10.1007/978-3-540-45243-0-70

[13] H. Sommer, J. R. Forbes, R. Siegwart, and P. Furgale, "Continuous-Time Estimation of Attitude Using B-Splines on Lie Groups," vol. 39, no. 2, pp. 242–261. [Online]. Available: http://arc.aiaa.org/doi/10.2514/1.G001149

[14] K. Koide and E. Menegatti, "General Hand–Eye Calibration Based on Reprojection Error Minimization," vol. 4, no. 2, pp. 1021–1028. [Online]. Available: https://ieeexplore.ieee.org/document/8616862/

[15] M. Mangusson, "The three-dimensional normal-distributions transform — an efficient representation for registration, surface analysis, and loop detection," Ph.D. dissertation, 2009.

[16] Zoran Lazarevic, "Hand-Eye Calibration." [Online]. Available: http://lazax.com/www.cs.columbia.edu/ laza/html/Stewart/matlab/handEye.m

[17] Z. Taylor and A. Millane, "ETHZ: Lidar-Align." [Online]. Available: https://github.com/ethz-asl/lidar_align

[18] Multiple Contributors, "Point Cloud Library: Normal Distributions Transform." [Online]. Available: https://github.com/PointCloudLibrary/pcl/tree/master/doc/tutorials/content

[19] Hornung, et al, "Octomap: An efficient probabilistic 3d mapping framework based on octrees," vol. 34, pp. 189–206. [Online]. Available: https://link.springer.com/article/10.10072Fs10514-012-9321-0