2013 IEEE/RSJ International Conference on
Intelligent Robots and Systems (IROS)
November 3-7, 2013. Tokyo, Japan

# Pairwise LIDAR Calibration Using Multi-Type 3D Geometric Features in Natural Scene

Mengwen He*, Huijing Zhao*, Franck Davoine+, Jinshi Cui*, Hongbin Zha*

***Abstract*— It has become a well-known technology that 3D measurement of a large environment could be achieved by using a number of 2D LIDARs on a mobile platform. In such a system, calibration is essential for making collaborative use of different LIDAR data, while existing methods usually require modifications to the environments, such as putting calibration targets, or rely on special facilities, which is labor intensive and put many restrictions to potential applications. This research aims at developing a calibration method for multiple 2D LIDAR sensing systems, which could be conducted in a general outdoor environment using the features of a nature scene. Special focus is cast on solving the noisy sensing in a complex environment and the occlusions caused by largely different sensor viewpoints. A multi-type geometric feature based calibration algorithm is proposed, which extracts the features such as points, lines, planes and quadrics from the 3D points of each LIDAR sensing. Transformation parameters from each sensor to the frame on a moving platform is estimated by matching the multi-type features. Experiments are conducted using the data sets of an intelligent vehicle platform (POSS-V) through a driving in the campus of Peking University. Results of calibrating two LIDAR sensors with largely different viewpoints are presented, and the accuracy and robustness concerning noisy feature extractions are examined intensively.**

## I. Introduction

Light Detection and Ranging (LIDAR) sensor has become more and more popular in robotics applications recently. As it has high accuracy in measuring both depth and bearing values, and it is not affected by illumination conditions, LIDAR sensor has been widely applied in many research topics, such as 3D modeling [1], [2], object detection [3], [4], [5], scene understanding [6], [7] and SLAM [8], [9], [10].

One way using a LIDAR sensor is to mount it on a moving platform with a localization module, such as a GPS/IMU navigation unit, to acquire a 3D sensing of the robot's working environment. As a 3D LIDAR, such as a Velodyne sensor, is much expensive, 2D LIDARs are generally used for such purposes [11]. As a 2D LIDAR scans only on a plane, multi-LIDAR sensing systems have been developed to simultaneously sense on different directions, so as to acquire a more complete knowledge to the whole 3D environment [12]. In such systems, calibration is essential to find the geometric transformations among different LIDAR sensor frames as well as the robot's body frame. A multi-LIDAR sensing system (called POSS-V system as shown in Fig. 1) has been developed in the authors' previous research [13]. Five 2D LIDAR sensors (L1-L5) are mounted on a vehicle platform for 3D sensing, and a GPS/IMU navigation unit is used for localization. This research focuses on the issue of pairwise LIDAR calibration, where the POSS-V system is used to study the algorithm.

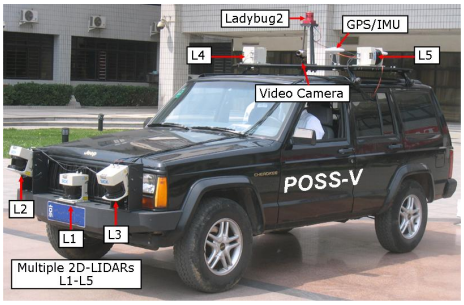


Fig. 1. The POSS-V platform that are used to study the calibration of a multiple 2D LIDAR sensing system.

Although many LIDAR sensing systems have been developed in literature, calibration (mainly refers to extrinsic calibration) is often poorly documented, which finds the transformations from each sensor to the system's body frame. Among the documented works, a multi-LIDAR sensing system is calibrated by recovering the rigid transformation among multiple LIDARs [14], [15] or from LIDARs to a reference frame, such as a camera [16], [17], [18] or the ego-vehicles odometry center [12]. This procedure usually requires some modification of the scene by introducing landmarks that are visible (or detectable) by both sensors, such as specifically placed reflective targets [14], poles [19], checker boards [20], which introduce laborious field work, or rely on special instruments or facilities [15]. Furthermore, some methods make use of visible laser [21] or require that the LIDAR sensor can measure reflectivity [14], which put strong restriction to the adaptiveness of the methods on a broad range of systems. Calibration of 3D LIDAR systems has also been studied [15], [22], and the authors stressed that their methods have the potentials being extended to 2D LIDARs calibration in natural scene. As their methods put some assumptions to the scene objects, such as contiguous surfaces, more elaborations are required in transforming the algorithm to a more general natural scene. More importantly, in a multiple 2D LIDAR sensing system, the scanning planes of different 2D LIDARs could be greatly different,

*M. He, H. Zhao, J. Cui and H. Zhao are with the State Key Lab of Machine Perception (MOE), School of EECS, Peking University, Beijing, P.R. China alexanderhmw at pku.edu.cn

+F. Davoine is with CNRS, LIAMA Sino French Laboratory, Beijing, China

This work is partially supported by the Hi-Tech Research and Development Program of China [2012AA011801], the NSFC-ANR Grants [61161130528] , and the NSFC Grants [91120010].

which yields much occlusions between the data of different sensors, and makes it a big challenge in generating data correspondences.

The research introduced in this paper aims at developing a method of calibrating a multiple 2D LIDAR sensing system, which could be conducted in a general outdoor environment, without any modification of the environment nor requiring any special facilities. Below we briefly denote LIDAR for a 2D LIDAR sensor. The scanning planes of LIDAR sensors could be largely different, with only small area of overlapped sensing. Consequently the algorithm needs to make use of the features from a natural scene, and it needs to have both accuracy and robustness to sensing noise and occlusions. A multi-type geometric feature based calibration algorithm is developed in this research, which extracts the features such as points, lines, planes and quadrics from the 3D points of each LIDAR sensing in a natural scene. Transformation parameters from each LIDAR sensor to the robot's body frame is estimated by matching the multi-type features. Experiments are conducted using the data sets of POSS-V through a driving in the campus of Peking University. Results of calibrating the LIDAR sensors $L3$ and $L4$ with the vehicle frame are presented, which have a special focus on examining accuracy and robustness concerning noisy feature extractions.

This article is structured as follows: the definition of multi-type geometric features as well as their extraction method are presented in section 2; a calibration algorithm using multi-type geometric features is described in section 3; experimental results and discussions are given in section 4, followed by conclusion and future works in section 5.

## II. Multi-type geometric features extraction in 3D points

### A. Formulation of 3D point generation

A typical LIDARs sensing platform mainly concerns three frames, i.e. LIDAR sensor frame, vehicle frame, and a world frame, among which two geometric transformations are concerned. They are a transformation $M_V^i$ that registers the vehicle frame at time $i$ to a world coordinate system, which is estimated through a localization procedure using such as a GPS/IMU navigation unit or SLAM, and varies with the vehicle's traveling; and a transformation $M_{L_j}$ that registers the sensor frame of LIDAR $L_j$ to the vehicle, which is regarded as time invariant in a rigid sensor platform. Here both $M_V^i$ and $M_{L_j}$ are in homogeneous notations. Thus, given a 2D point $\mathbf{q}^i$ measured by LIDAR $L_j$ at time $i$, a 3D coordinate $\mathbf{p}^i$ in the world coordinate system can be estimated by sequentially aligning $M_{L_j}$ and $M_V^i$ as below:

$$\tilde{\mathbf{p}}^i = M_V^i \cdot M_{L_j} \cdot \tilde{\mathbf{q}}^i \tag{1}$$

$$\left(\tilde{\mathbf{p}}^i = \left[\mathbf{p}^T, 1\right]^T, \tilde{\mathbf{q}}^i = \left[\mathbf{q}^T, 0, 1\right]^T\right)$$

In real measurements, both $M_V^i$ and $M_{L_j}$ could be erroneous. As this research focus on sensor calibration, which aims at improving the accuracy of $M_{L_j}$, we bound localization errors by restricting on experimental sites to such as open area for a good GPS localization, or a flat ground for a LIDAR-based 2D SLAM, etc. and the localization error we used is derived by the localization method itself. Given an initial estimation $\hat{M}_{L_j}$, which contains calibration error $M_{e_j}$ with respect to its true value $M_{L_j}$, a 3D coordinate $\hat{\mathbf{p}}^i$ is estimated subsequently on the erroneous calibration parameters as below:

$$\begin{aligned} \hat{\tilde{\mathbf{p}}}^i &= M_V^i \cdot \hat{M}_{L_j} \cdot \tilde{\mathbf{q}}^i \\ &= M_V^i \cdot (M_{e_j} \cdot M_{L_j}) \cdot \tilde{\mathbf{q}}^i \end{aligned} \tag{2}$$

With Eq. 1 and Eq. 2, a transformation $M_A^i$ from a calibration error free 3D point $\mathbf{p}^i$ to an erroneous one $\hat{\mathbf{p}}^i$ is obtained as below:

$$\begin{aligned} \hat{\tilde{\mathbf{p}}}^i &= M_A^i \cdot \tilde{\mathbf{p}}^i \\ &= (M_V^i \cdot M_{e_j} \cdot {M_V^i}^{-1}) \cdot \tilde{\mathbf{p}}^i \end{aligned} \tag{3}$$

In this research, calibration of a multi-LIDAR sensing system is formulated as a problem of registering the 3D points of different LIDAR sensors $L_j$ to reduce the calibration errors $M_{e_j}$ in $\hat{M}_{L_j}$. As LIDAR sensing is a sparse sampling of 3D points on the surfaces of surrounding objects, it is difficult to generate a direct point-to-point matching, especially when the viewing directions and scanning planes are largely different. However the inherent geometric features implied in the 3D point clouds could be more reliable in generating correspondences between different data sets, and more robust to sparse sampling and occlusions.

### B. Multi-Type Geometric Feature Definition

Given a set of 3D points $P = \{\mathbf{p}^i\}$ of a simple geometric shape, which could be a point, a line, a plane or a quadric, a geometric formula $f^\phi(\mathbf{p}) = 0$ could be found by fitting on the 3D points, where $\phi$ denotes the type of a geometric shape. Formulations to the most commonly used geometric shapes are listed in Table.I.

In addition, confidence measures $\left(\mu, \sigma^2\right)$ are defined as below, which are important indexes evaluating fitting reliability as well as reflecting inherent properties of the object surface, such as smoothness, material etc. Higher $\mu$ means lower fitting reliability, and higher $\sigma^2$ means higher noise. In order to make sure the mean is nonnegative, especially for quadric features, we use a second-order error covariance model.

$$\begin{cases} \mu &= E\left(f^\phi(\mathbf{p}^i)^2\right) \\ \sigma^2 &= D\left(f^\phi(\mathbf{p}^i)^2\right) \end{cases} \tag{4}$$

$$(E : \text{expectation}, D : \text{deviation})$$

A geometric feature is thus defined as below:

$$G = \left\{P, \phi, f^\phi, (\mu, \sigma^2)\right\} \tag{5}$$

As defined in Eq. 3, if a calibration error exists, i.e. $M_{e_j} \neq I$, a deviation of $\hat{\mathbf{p}}^i$ from its true 3D coordinates $\mathbf{p}^i$ exists

TABLE I
DESCRIPTION OF GEOMETRIC SHAPE

| Geometric shape | Surface equation | Notation on coefficients |
|---|---|---|
| $\phi_0$: point | $f^{\phi_0}(\mathbf{p}) = (\mathbf{p}-\mathbf{o})^T \cdot (\mathbf{p}-\mathbf{o}) = 0$ | $\mathbf{o}$: the point position |
| $\phi_1$: line | $f^{\phi_1}(\mathbf{p}) = (\mathbf{p}-\mathbf{o})^T \cdot (\mathbf{D}_1 \cdot \mathbf{D}_1^T + \mathbf{D}_2 \cdot \mathbf{D}_2^T) \cdot (\mathbf{p}-\mathbf{o}) = 0$ | $\mathbf{o}$: a point on line<br>$\mathbf{D}_i$: unit vectors perpendicular to the line |
| $\phi_2$: plane | $f^{\phi_2}(\mathbf{p}) = (\mathbf{p}-\mathbf{o})^T \cdot (\mathbf{D}_1 \cdot \mathbf{D}_1^T) \cdot (\mathbf{p}-\mathbf{o}) = 0$ | $\mathbf{o}$: a point on plane<br>$\mathbf{D}_1$: normal vector of the plane |
| $\phi_3$: cone | $f^{\phi_3}(\mathbf{p}) = (\mathbf{p}-\mathbf{o})^T \cdot (\frac{\mathbf{D}_1 \cdot \mathbf{D}_1^T}{V_1} + \frac{\mathbf{D}_2 \cdot \mathbf{D}_2^T}{V_2} - \mathbf{D}_3 \cdot \mathbf{D}_3^T) \cdot (\mathbf{p}-\mathbf{o}) = 0$ | $\mathbf{o}$: the apex of the cone<br>$\mathbf{D}_3$: principle direction vector of the cone<br>$\mathbf{D}_1, \mathbf{D}_2$: unit vectors perpendicular to $\mathbf{D}_3$<br>$V_i$: scale factors |

too, i.e. $M_A^i \neq I$, yielding changes in the geometric shape of the 3D point set. Thus estimation of the above defined geometric features could be affected by the calibration error. However, if we restrict the vehicle's motion on a straight path, which means that the rotation term in $M_V^i$ is a constant, deviation of the 3D points made by a calibration error $M_{e_j}$ can be proved as an invertible affine transformation (Eq. 13). A derivation can be found in Appendix A. Therefore, the type of a geometric feature keeps consistency in case of calibration errors.

Thus the proposed approach has a constraint on the vehicle's motion during calibration, i.e. on a straight path. Although the completely straight path is impossible, we can assume that the path is straight in a short range and then the corresponding 3D points of geometric feature will fullfill the invertible affine transformation. Although there exists a minor deviation, its influence can be reduced by confidence measures.

### C. Multi-type Geometric Features Extraction

As defined previously, a geometric feature has four components, i.e. a 3D point set $P = \{\mathbf{p}^i\}$, the type of the geometric shape $\phi$, a geometric formula $f^\phi$ and its confidential measures $(\mu, \sigma^2)$.

In order to extract sets of 3D points, which represent the geometric shapes that are modeled in this research, 3D data segmentation is conducted firstly. The Point Cloud Library (PCL) [23] provides several methods such as on region growing, min-cut, difference of normals (DoN) etc, which are used in our research. We first segment a 3D data using both region growing and DoN, then manually select some of the data pieces that have simple geometric shapes for geometric feature estimation. For example, the 3D points on poles and road curbs have linear shapes, those on wall and ground surface are planar, the upper apexes of traffic cones can be used in generating point features.

Given a set of 3D points $P = \{\mathbf{p}^i\}$, the rest in a geometric feature can be estimated automatically. For a specific type of geometric shape $\phi$, a geometric feature $G_\phi$ is estimated by fitting a geometric formula $f^\phi$ on the 3D points using a Least Square Method, and the confidential measures $(\mu, \sigma^2)$ are estimated as well. By enumerating all potential types of geometric shapes, a set of geometric features $\{G_\phi\}$ is obtained. The $G_\phi$ that minimize $E(f^\phi(\mathbf{p}^i)^2)$ is selected as the result.

In this research, for implementation reasons, only three types $\phi \in \{point, line, plane\}$ are concerned in experiments. Currently, geometric feature extraction needs some manual work, e.g. identifying a data piece of simple geometric shape for feature estimation. These will be improved through future work by developing an automated data selection method.

## III. PAIRWISE LIDAR CALIBRATION USING MULTI-TYPE GEOMETRIC FEATURES

### A. Pairwise LIDAR Calibration

With a LIDAR $L_1$ as the reference, here we present a method of calibrating another LIDAR $L_2$ with $L_1$ using multi-type geometric features, i.e. pairwise calibration. If there exists an overlapped scan area of these two LIDARs, then the alignment of paired geometric features in this area will determine the transformation between $L_1$ and $L_2$.

*1) A distance measure on multi-type geometric features:* In many ICP (Iterative Closest Point) methods, the distance measure between two 3D point sets is the average distance of closest 3D point pair or the average distance between point and the closest local plane [24]. However, in order to increase the robustness of the distance measure, the confidence degree $(\mu, \sigma^2)$ in Eq. 4 should be considered. Inspired by Gaussian Models [25], we defined a weighted distance measure of geometric features.

$$D(G_1, G_2) = \frac{1}{n_2} \sum_{i=1}^{n_2} \frac{\left(f_1^\phi(\mathbf{p}_2^i)^2 - \mu_1\right)^2}{\sigma_1^2} \tag{6}$$

$f_1^\phi(\mathbf{p}_2^i)$ means spatial distance between a 3D point $\mathbf{p}_2^i$ in geometric feature $G_2$ and the geometric shape $\phi$ in geometric feature $G_1$.

This distance measure takes both spatial distance measure and surface equation reliability into consideration, which increases its robustness.

*2) Optimization for geometric features alignment:* Assume there is a set of paired geometric features with different types of shapes in overlapped scan area of $L_1$ and $L_2$.

$$\left\{< G_1^j, G_2^j >\right\}$$

Then the optimization formula $H(\hat{M}_{L_2})$ is defined as the sum of all weighted distances (Eq. 6) of paired geometric features, where the superscript $*$ means this is a new value after alignment and it is a homogeneous notation.

$$\begin{aligned} \hat{M}^*_{L_2} &= \arg\min_{\hat{M}_{L_2}} H\left(\hat{M}_{L_2}\right) \\ &= \arg\min_{\hat{M}_{L_2}} \sum_j D\left(G^j_1, G^j_2\left(\hat{M}_{L_2}\right)\right) \end{aligned} \tag{7}$$

Here, we defined a function $G\left(\hat{M}_L\right)$ for convenient expression on the transformation of a geometric feature's 3D point set caused by $\hat{M}_L$. ($\hat{M}_L$ is the estimated orientation and position of a LIDAR in the vehicle frame)

$$G\left(\hat{M}_L\right) = \left\{\left\{M^i_V \cdot \hat{M}_L \cdot \tilde{\mathbf{q}}^i\right\}, \phi, f^\phi, \{\mu, \sigma^2\}\right\}$$

As addressed previously that the deviation of the true 3D points is an invertible affine transformation, alignment of paired geometric features is feasible, which means the optimization formula theoretically has exactly one solution (the relationship between the calibration error and the invertible affine transformation is monomorphism). This optimization formula can be solved by numerical method with an initial estimated orientation and position of $L_2$. Previously, in our experiment, we used $fmincon$ function in Matlab (a NLM optimizer with constraints) to solve this problem, with 12 variables. Currently, we use $fminsearch$ function in Matlab (a NLM optimizer without constraints) to solve this problem, with 6 variables. And in addition, we use data-sample and warm-start strategies to accelerate the optimization process: the sample rate in each iteration will increase and every iteration will use its last result as initial value.

### *B. Calibration of a Reference LIDAR with the Vehicle Frame*

If the reference LIDAR $L_1$ has calibration error $M_{e_1}$, then after pairwise calibration, another LIDAR $L_2$ will have a new calibration error $M^*_{e_2}$. One important thing is that after pairwise calibration, $M_{e_1} \neq M^*_{e_2}$ in general (see Appendix B). This means we cannot directly eliminate calibration error $M^*_{e_2}$ by eliminating calibration error $M_{e_1}$, which is done by reference LIDAR calibration, because $M^{-1}_{e_1} \cdot M^*_{e_2} \neq I$.

So in order to get well-calibrated $L_2$, the reference LIDAR $L_1$ must be calibrated to the vehicle before applying the pairwise calibration between $L_1$ and $L_2$.

In our experiment, the reference LIDAR calibration method is similar to [15], but instead we use a vertical wall in natural scene.

The reference LIDAR is regarded as a target for another LIDAR, so the calibration error of reference LIDAR will pass to the calibrated LIDAR. But we have shown the relationship between these errors in theory and in our experiment, we proved that the error of calibrated LIDAR and the error of reference LIDAR are in the same level. The next step of our work is to develop a reference LIDAR calibration method which also only uses geometric features and has low error.

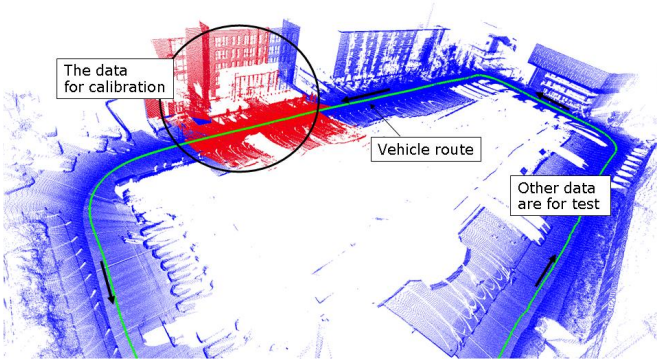


Fig. 2. A view of the experimental data for both calibration and testing.

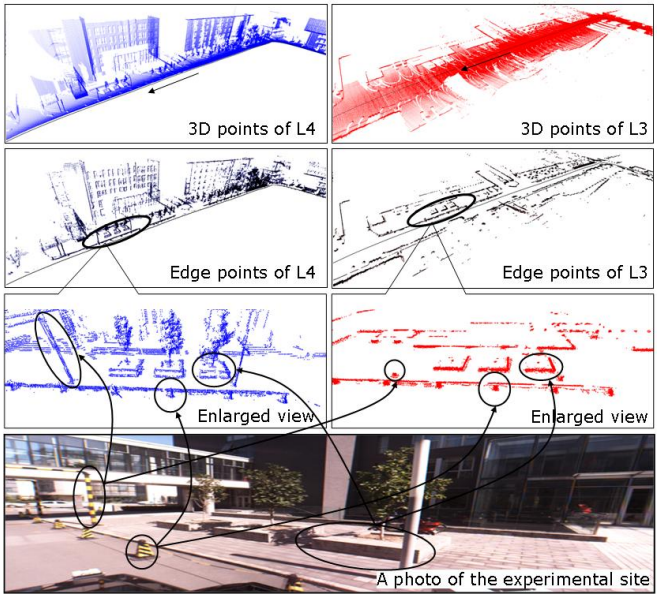


Fig. 3. The 3D points measured by the LIDAR sensors $L3$ and $L4$.

## IV. Experimental Results

The vehicle platform POSS-V is used in this research to evaluate the algorithm of multiple LIDAR calibration. As shown in Fig. 1, the LIDAR sensor $L_3$, which scans downward to the ahead road surface, and $L_4$, which scans vertically to the right of the vehicle, are calibrated with the vehicle body frame that is defined by the GPS/IMU navigation system. The calibration is achieved by matching the multi-type geometric features that are extracted from the 3D data of each LIDAR sensor. Data collection was conducted during a vehicle run about 400 meters in the campus of Peking University. As shown in Fig. 2, the 3D points in red are used for calibration, while others are used to examine the accuracy. Below we present experimental results in calibration, while focus will be cast on examining its accuracy and robustness concerning noisy and occluded data, which are the major challenges in using the data of a nature scene.

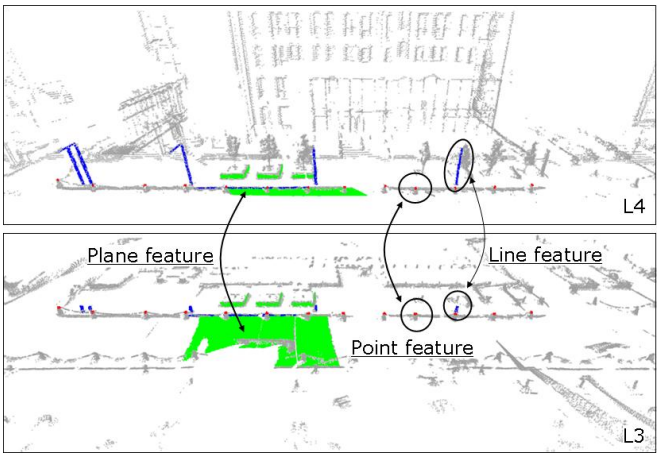


Fig. 4. Three kinds of geometric features are extracted for calibration. They are points, lines and planes.

### A. Calibration on Multi-Type Geometric Features

A pair of 3D data that are measured by $L_3$ and $L_4$ are shown in Fig. 3, while for a better visualization of the geometric details, edge points were extracted by filtering out the points that have low variance in local surface fitting [26]. It is obvious that correspondence between two LIDAR data sets could be generated using the geometric features of their measurements to common objects. The multi-type geometric features are extracted as defined in section 2. As shown in Fig. 4, the apexes of traffic cones are extracted as point features, road curbs or light poles are extracted as line features, road or wall surfaces are extracted as plane features. In the experiment, the cone geometric features are extracted, but the estimation error of surface equation is large. Because of the robustness of this calibration algorithm, this error or this geometric feature will not play a big role in the final results and we did not add the quadric features. In the future, we will improve the estimation method of quadric surface equation estimation.

A list of the multi-type geometric features that are extracted for calibration is given in Table. II, where reliability of the features are evaluated on $\mu$ as defined in section 2. In this experiment, the geometric features are extracted with some manual work. However we consider that these features could be extracted automated through future developments.

A calibration result by matching the multi-type geometric features using the proposed algorithm is shown in Fig. 5. The data of $L_3$ is shown in blue, while $L_4$ in red. Before calibration, there were obvious displacements between the data sets of $L_3$ and $L_4$, which is demonstrated using the edge points extracted from the 3D point clouds of each sensor. After calibration, a better consistency is achieve between the two data sets.

As mentioned previously, the data used for calibration and testing are different, which are shown in Fig. 2. The calibration result is examined using the data beyond those for calibration. Three sets of results on testing data are shown in Fig. 6, each consists of a view before and after calibration.

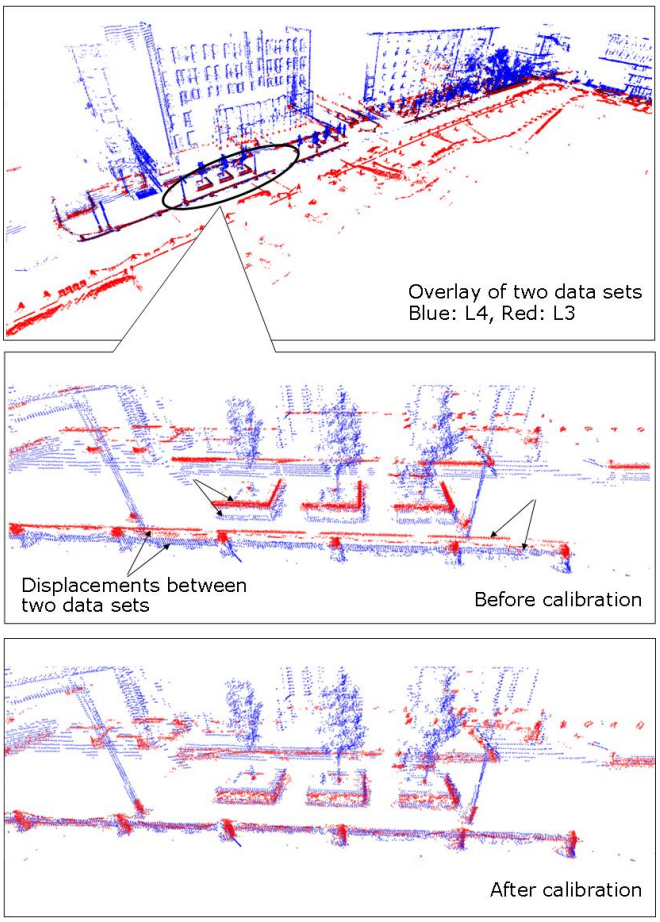


Fig. 5. Calibration results: visualization of the result on calibration data.

Points are colored in the same way with those in Fig. 5, representing the data of $L_3$ and $L_4$ respectively. It could be found that after calibration, data consistency between sensors has been greatly improved. For example in the data view 3, $L_4$ measured the whole body of a pedestrian, while $L_3$ captured only his legs. After calibration, the matching of the two data sets are improved, and the legs of the pedestrian are much clearer than before. In fact, $L_3$, which scans downward to the ahead road surface, measured the legs data earlier than $L_4$. A good data consistency will not only help for fusion of the two data sets, but also make collaborative usage of different sensors possible, e.g. after detecting a candidate in ahead through coarse sensing, the system could inform another sensor to find a better measurement to validate the candidate.

TABLE II
THE GEOMETRIC FEATURES USED IN CALIBRATION

| Geo-feature | Object | Number | Reliability |
|---|---|---|---|
| Point | Traffic Cone apex | 13 | High |
| Line | Light pole | 5 | High |
| | Road curb | 4 | Low |
| Plane | Flower bed wall | 6 | Very High |
| | Road surface | 1 | Middle |

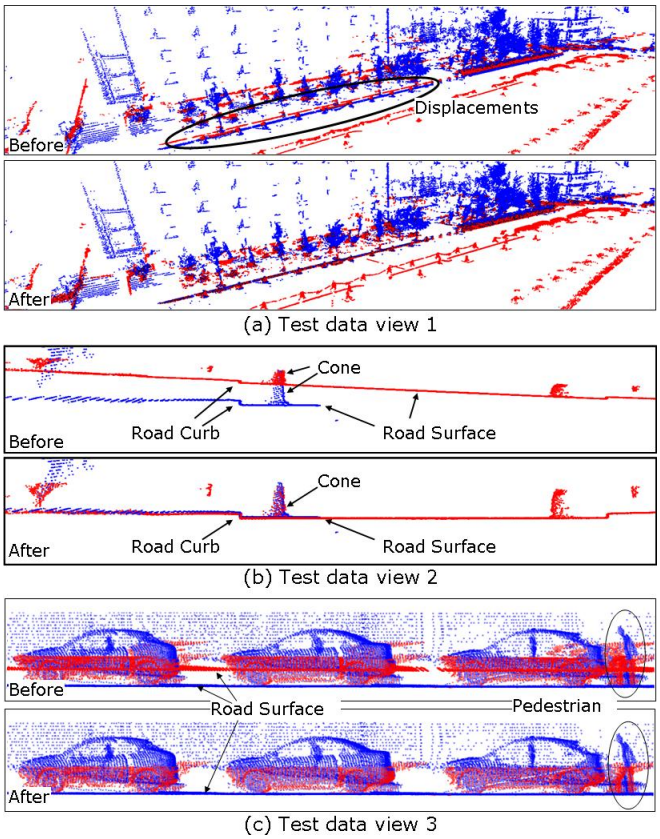


Fig. 6. Examination of the calibration results: visualization of the result on testing data.

In addition, we manually extracted 50 pairs of planes from the testing data of both sensors, and compared their difference before and after calibration to evaluate the calibration accuracy. Three parameters are estimated. They are the averages of the angular difference, the expectation and deviation of the spatial distance $D_S$ between corresponding planes. The result is given in Table. III, that shows how the values of all three parameters are reduced greatly after calibration.

TABLE III
EVALUATION OF CALIBRATION ACCURACY

| | $E(Angle)$ | $E(D_S)$ | $D(D_S)$ |
|---|---|---|---|
| Before Calibration | 2.600109 | 0.043408 | 0.002392 |
| After Calibration | 0.891429 | 0.009808 | 0.000105 |

### B. Robustness and Accuracy Examination

In this section, we present the experimental results in more intensive study of the calibration performance. The examinations focus on two aspects, i.e. the efficiency in combining multi-type geometric features and the robustness concerning noisy feature extractions. Accuracies in the calibrations using mono-type, or combinations of two or three types are compared to examine the efficiency in making use of multi-type features. On the other hand, noisy features are introduced, which are extracted from such as bended walls or curved curbs, with high $\mu$ values in Eq. 4. Accuracies in the calibrations with and without noisy features are compared too. The results are listed in Table. IV. Calibration accuracy is evaluated in the same way with those in Table. III by estimating the averages of the expectation and deviation of the spatial distance $D_S$ between corresponding planes. Lower values stand for higher accuracy. On the other hand, significance of noisy features' effect on the calibration is evaluated using the Error Ratio $E_2/E_1$, where $E_1$ and $E_2$ are the estimated calibration errors without and with noisy features, respectively. Subjective evaluations to the algorithms are also scored by the operator for reference.

From Table. IV, we could find the following summaries:

1) In the calibration on mono-type features, plane tends to give higher accuracy than point and line, but is more vulnerable to noisy extractions;
2) Combined use of multi-type features can generally improve calibration accuracy and robustness;
3) More types of features tends to increase robustness. Point-Line-Plane combination raises the best calibration results.

In the calibrations on mono-type features, the calibration error without noisy features is 0.01939 on planes, while 0.03232 on points or 0.07797 on lines. However the accuracy of plane-based calibration is greatly degraded, which has an error ratio of 3.031 comparing with 1.427 on points or 1.994 on lines. As for the calibrations on multi-type feature combination, there is one exception, i.e. Line-Plane combination. Without noisy features, lines degraded the accuracy of plane-based calibration, i.e. from 0.01939 to 0.0247. However, in case that noisy features exist, lines help to improve plane-based calibration accuracy (i.e. from 0.05878 to 0.05510) and robustness (i.e. from 3.031 to 2.229). Among all the calibrations, the combinative use of all types of features raise the best results concerning both accuracy and robustness.

## V. CONCLUSION

Aiming at developing a calibration method for multiple LIDAR sensing systems, which could be conducted in a general outdoor environment using the features of a nature scene, a multi-type geometric feature based calibration algorithm is proposed, which extracts the features such as point, line, plane and quadric from the 3D points of each LIDAR sensing, transformation parameters from each sensor to the frame of moving platform is estimated by matching the multi-type features. Special focus has been cast on solving the noisy sensing in a complex environment and the occlusions caused by largely different sensor viewpoints. Experiments are conducted using the data sets of an intelligent vehicle platform (POSS-V) through a driving in the campus of Peking University. Results of calibrating two LIDAR sensors with largely different viewpoints are presented, demonstrating their accuracy and robustness concerning noisy feature extractions. Future work will address on an automated feature extraction and selection algorithm,

TABLE IV

ACCURACY AND ROBUSTNESS EVALUATION CONCERNING MULTI-TYPE FEATURE COMBINATION AND EXTRACTION ERROR

| **Multi-Types** | **Geometric Features** | **Error without noisy features** $E_1 = E(D_S) \pm D(D_S)$ | **Error with noisy features** $E_2 = E(D_S) \pm D(D_S)$ | **Error Ratio=** $E_2/E_1$ | **Accuracy Evaluation** |
|---|---|---|---|---|---|
| Mono | Point | $0.03232 \pm 0.00316$ | $0.04615 \pm 0.00620$ | 1.427 | Bad |
| | Line | $0.07797 \pm 0.00896$ | $0.15548 \pm 0.02944$ | 1.994 | Bad |
| | Plane | $0.01939 \pm 0.00042$ | $0.05878 \pm 0.00521$ | 3.031 | Good |
| Two | Point + Line | $0.01363 \pm 0.00053$ | $0.01429 \pm 0.00071$ | 1.048 | Good |
| | Point + Plane | $0.00980 \pm 0.00011$ | $0.00993 \pm 0.00010$ | 1.013 | Excellent |
| | Line + Plane | $0.02471 \pm 0.00076$ | $0.05510 \pm 0.00475$ | 2.229 | Good |
| Three | Point + Line + Plane | $0.00981 \pm 0.00011$ | $0.00989 \pm 0.00010$ | 1.008 | Excellent |

which is a basis in extending the method to an online based calibration.

## VI. APPENDIX

### A. Invertible affine transformation of the true 3D points

1) If vehicle drives straightly, the $M_V^i$'s rotation term $R_V^i$ is constant (Eq. 8) and its translation term $\mathbf{T}_V^i$ can be derived from corresponding 3D point data $\mathbf{p}^i$ (Eq. 9).

$$R_V^i = R_V \tag{8}$$

$$\mathbf{T}_V^i = J \cdot (\mathbf{p}^i - \mathbf{O}_L^0) + \mathbf{T}_V^0 \tag{9}$$

$$\left( J = R_V \cdot \frac{\mathbf{t} \cdot \mathbf{n}^T}{\mathbf{n}^T \cdot \mathbf{t}} \cdot R_V^T \right)$$

- $\mathbf{t}$ :the vehicle forward direction in the vehicle frame;
- $\mathbf{n}$ :the LIDAR scan plane's normal in the vehicle frame;
- $\mathbf{T}_V^0$ :the straight path's start position in the world frame;
- $\mathbf{O}_L^0$ :the LIDAR's origin position corresponds to $\mathbf{T}_V^0$ in the world frame.

*Proof:* If vehicle drives straightly, then $\mathbf{T}_V^i = d^i(R_V \cdot \mathbf{t}) + \mathbf{T}_V^0$. Define $\mathbf{O}_L^i$ as the LIDAR's origin position in world frame corresponds to $\mathbf{T}_V^i$, and $\mathbf{O}_L^i = R_V \cdot \mathbf{T}_L + \mathbf{T}_V^i$ ($\mathbf{T}_L$ is the true LIDAR's position in vehicle frame).

Because $\mathbf{O}_L^i$ and 3D point data $\mathbf{p}^i$ are always on the same LIDAR scan plane.

Therefore

$$\begin{aligned}
(R_V \cdot \mathbf{n})^T \cdot (\mathbf{O}_L^i - \mathbf{p}^i) &= 0 \\
(R_V \cdot \mathbf{n})^T \cdot (R_V \cdot \mathbf{T}_L + \mathbf{T}_V^i - \mathbf{p}^i) &= 0 \\
(R_V \cdot \mathbf{n})^T \cdot (R_V \cdot \mathbf{T}_L + d^i(R_V \cdot \mathbf{t}) + \mathbf{T}_V^0 - \mathbf{p}^i) &= 0 \\
(R_V \cdot \mathbf{n})^T \cdot (d^i(R_V \cdot \mathbf{t}) + (\mathbf{O}_L^0 - \mathbf{p}^i)) &= 0 \\
d^i(\mathbf{n}^T \cdot \mathbf{t}) + \mathbf{n}^T \cdot R_V^T \cdot (\mathbf{O}_L^0 - \mathbf{p}^i) &= 0
\end{aligned}$$

If $\mathbf{n}^T \cdot \mathbf{t} \neq 0$, then

$$d^i = \frac{\mathbf{n}^T \cdot R_V^T \cdot (\mathbf{p}^i - \mathbf{O}_L^0)}{\mathbf{n}^T \cdot \mathbf{t}}$$

Because $\mathbf{T}_V^i = d^i(R_V \cdot \mathbf{t}) + \mathbf{T}_V^0$.

Therefore

$$\mathbf{T}_V^i = (R_V \cdot \mathbf{t}) \cdot \frac{\mathbf{n}^T \cdot R_V^T \cdot (\mathbf{p}^i - \mathbf{O}_L^0)}{\mathbf{n}^T \cdot \mathbf{t}} + \mathbf{T}_V^0$$

$$\mathbf{T}_V^i = (R_V \cdot \frac{\mathbf{t} \cdot \mathbf{n}^T}{\mathbf{n}^T \cdot \mathbf{t}} \cdot R_V^T) \cdot (\mathbf{p}^i - \mathbf{O}_L^0) + \mathbf{T}_V^0$$

Define $J = R_V \cdot \frac{\mathbf{t} \cdot \mathbf{n}^T}{\mathbf{n}^T \cdot \mathbf{t}} \cdot R_V^T$, then

$$\mathbf{T}_V^i = J \cdot (\mathbf{p}^i - \mathbf{O}_L^0) + \mathbf{T}_V^0$$

■

2) From the Eq. 3

$$\hat{\tilde{\mathbf{p}}}^i = M_A^i \cdot \tilde{\mathbf{p}}^i = (M_V^i \cdot M_e \cdot {M_V^i}^{-1}) \cdot \tilde{\mathbf{p}}^i$$

If vehicle drives straightly, then

$$\forall \mathbf{p}^i \in P,\ \exists \text{ an invertible } M_A : \hat{\tilde{\mathbf{p}}}^i = M_A^i \cdot \tilde{\mathbf{p}}^i = M_A \cdot \tilde{\mathbf{p}}^i$$

*Proof:* From Eq. 8 and Eq. 9, we have known the $M_V^i$'s rotation term $R_V^i$ and translation term $T_V^i$. Then

$$\begin{aligned}
&(M_V^i \cdot M_e \cdot {M_V^i}^{-1}) \cdot \tilde{\mathbf{p}}^i \\
= &\begin{bmatrix} R_V & \mathbf{T}_V^i \\ \mathbf{0} & 1 \end{bmatrix} \cdot \begin{bmatrix} R_e & \mathbf{T}_e \\ \mathbf{0} & 1 \end{bmatrix} \\
&\cdot \begin{bmatrix} R_V^{-1} & -R_V^{-1} \cdot \mathbf{T}_V^i \\ \mathbf{0} & 1 \end{bmatrix} \cdot \tilde{\mathbf{p}}^i
\end{aligned}$$

Define

$$E = R_V \cdot R_e \cdot R_V^{-1} \tag{10}$$

$$R_A = E + (I - E) \cdot J \ (J: \text{ see Eq. 9}) \tag{11}$$

$$\mathbf{T}_A = R_V \cdot \hat{\mathbf{T}}_L - R_A \cdot \mathbf{O}_L^0 + T_V^0 \tag{12}$$

($\hat{\mathbf{T}}_L$ : LIDAR's estimated position in vehicle frame)

Then

$$\begin{aligned}
&(M_V^i \cdot M_e \cdot {M_V^i}^{-1}) \cdot \tilde{\mathbf{p}}^i \\
= &\begin{bmatrix} E & (I - E) \cdot \mathbf{T}_V^i + R_V \cdot \mathbf{T}_e \\ \mathbf{0} & 1 \end{bmatrix} \cdot \tilde{\mathbf{p}}^i \\
= &\begin{bmatrix} R_A & \mathbf{T}_A \\ \mathbf{0} & 1 \end{bmatrix} \cdot \tilde{\mathbf{p}}^i
\end{aligned}$$

$$\hat{\tilde{\mathbf{p}}}^i = M_A^i \cdot \tilde{\mathbf{p}}^i = \begin{bmatrix} R_A & \mathbf{T}_A \\ \mathbf{0} & 1 \end{bmatrix} \cdot \tilde{\mathbf{p}}^i = M_A \cdot \tilde{\mathbf{p}}^i \tag{13}$$

It is easy to demonstrate that if $(R_e \cdot \mathbf{n})^T \cdot \mathbf{t} \neq 0$, then $R_A$ is invertible. So $M_A$ is an invertible affine transformation.

■

### B. Relationship between calibration errors after geometric features alignment

Take LIDAR $L_1$ as reference LIDAR and assume $L_1$'s calibration error is $M_{e_1}$, then we align geometric features of LIDARs $L_1$ and $L_2$ to get the LIDAR $L_2$'s new estimated orientation and position $M^*_{L_2}$ (Eq. 7), whose calibration error is $M^*_{e_2}$. From Eq. 13, the geometric features alignment result can be expressed as Eq. 14 in theory.

$$\forall \mathbf{p}^i \in P_1 \cap P_2, M_{A_1} \cdot \tilde{\mathbf{p}}^i = M^*_{A_2} \cdot \tilde{\mathbf{p}}^i \tag{14}$$

where, $M_{A_1}$ is the deviation transformation of the true 3D points caused by $M_{e_1}$ and $M^*_{A_2}$ is the deviation transformation caused by $M^*_{e_2}$.

Because $M_{A_1}$ and $M^*_{A_2}$ are invertible, we can derive $R_{A_1} = R^*_{A_2}$ and $T_{A_1} = T^*_{A_2}$. Here, we just study the rotation term.

$$\begin{aligned} R_{A_1} &= R^*_{A_2} \\ E_1 + (I - E_1) \cdot J_1 &= E^*_2 + (I - E^*_2) \cdot J_2 \\ R_{e_1} + (I - R_{e_1}) \cdot \frac{\mathbf{t} \cdot \mathbf{n}_1^T}{\mathbf{n}_1^T \cdot \mathbf{t}} &= R^*_{e_2} + (I - R^*_{e_2}) \cdot \frac{\mathbf{t} \cdot \mathbf{n}_2^T}{\mathbf{n}_2^T \cdot \mathbf{t}} \end{aligned}$$

Therefore, if $\mathbf{n}_1 \neq \mathbf{n}_2$, then $R_{e_1} \neq R^*_{e_2}$ unless $R_{e_1} \cdot \mathbf{t} = \mathbf{t}$. Furthermore, $M_{e_1} \neq M^*_{e_2}$ in general.

In addition, from the last equation, we can derive that iff $M_{e_1} = I$, then $M^*_{e_2} = I$. This is the principle of our calibration method, which is based on geometric features alignment.

## REFERENCES


[1] J. Levinson and S. Thrun, "Robust vehicle localization in urban environments using probabilistic maps." in *IEEE Int. Conf. Robotics and Automation (ICRA)*, 2010, pp. 4372–4378.

[2] S. Hu and A. Zhang, "3D laser omnimapping for 3D reconstruction of large-scale scenes," in *IEEE Urban Remote Sensing Event*, 2009, pp. 1–5.

[3] B. Douillard, A. Brooks, and F. Ramos, "A 3D laser and vision based classifier," in *IEEE Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*, 2009, pp. 295–300.

[4] F. Endres, C. Plagemann, C. Stachniss, and W. Burgard, "Unsupervised discovery of object classes from range data using latent Dirichlet allocation." in *Robotics: Science and Systems*, J. Trinkle, Y. Matsuoka, and J. A. Castellanos, Eds. The MIT Press, 2009.

[5] K. Lai and D. Fox, "3D laser scan classification using web data and domain adaptation." in *Robotics: Science and Systems*, J. Trinkle, Y. Matsuoka, and J. A. Castellanos, Eds. The MIT Press, 2009.

[6] D. Steinhauser, O. Ruepp, and D. Burschka, "Motion segmentation and scene classification from 3D LIDAR data," in *IEEE Intelligent Vehicles Symposium*, 2008, pp. 398–403.

[7] H. Zhao, Y. Liu, X. Zhu, Y. Zhao, and H. Zha, "Scene understanding in a large dynamic environment through a laser-based sensing." in *IEEE Int. Conf. Robotics and Automation (ICRA)*, 2010, pp. 127–133.

[8] A. Segal, D. Hähnel, and S. Thrun, "Generalized-ICP." in *Robotics: Science and Systems*, J. Trinkle, Y. Matsuoka, and J. A. Castellanos, Eds. The MIT Press, 2009.

[9] D. M. Cole, A. R. Harrison, and P. M. Newman, "Using naturally salient regions for SLAM with 3D laser data," in *IEEE Int. Conf. Robotics and Automation (ICRA), SLAM Workshop*, 2005.

[10] A. Nuchter, K. Lingemann, J. Hertzberg, and H. Surmann, "6d SLAM with approximate data association," in *IEEE Int. Conf. Advanced Robotics (ICAR)*. IEEE, 2005, pp. 242–249.

[11] C. Urmson, J. Anhalt, D. Bagnell, C. Baker, R. Bittner, M. Clark, J. Dolan, D. Duggins, T. Galatali, C. Geyer *et al.*, "Autonomous driving in urban environments: Boss and the urban challenge," *J. Field Robotics*, vol. 25, no. 8, pp. 425–466, 2008.

[12] J. P. Underwood, A. Hill, T. Peynot, and S. Scheding, "Error modeling and calibration of exteroceptive sensors for accurate mapping applications." *J. Field Robotics*, vol. 27, no. 1, pp. 2–20, 2010.

[13] H. Zhao, L. Xiong, Z. Jiao, J. Cui, H. Zha, and R. Shibasaki, "Sensor alignment towards an omni-directional measurement using an intelligent vehicle," in *IEEE Intelligent Vehicles Symposium*, 2009, pp. 292–298.

[14] C. Gao and J. R. Spletzer, "On-line calibration of multiple LIDARs on a mobile vehicle platform." in *IEEE Int. Conf. Robotics and Automation (ICRA)*, 2010, pp. 279–284.

[15] M. Sheehan, A. Harrison, and P. Newman, "Self-calibration for a 3D laser," *Int. J. Robotics Research*, vol. 31, no. 5, pp. 675–687, 2012.

[16] D. Scaramuzza, A. Harati, and R. Siegwart, "Extrinsic self calibration of a camera and a 3D laser range finder from natural scenes." in *IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, 2007, pp. 4164–4169.

[17] G. Lisca, P. Jeong, and S. Nedevschi, "Automatic one step extrinsic calibration of a multi layer laser scanner relative to a stereo camera," in *IEEE Int. Conf. Intelligent Computer Communication and Processing (ICCP)*, 2010, pp. 223–230.

[18] Q. Zhang and R. Pless, "Extrinsic calibration of a camera and laser range finder (improves camera calibration)." in *IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, 2004, pp. 2301–2306.

[19] J. P. Underwood, A. J. Hill, and S. Scheding, "Calibration of range sensor pose on mobile platforms." in *IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, 2007, pp. 3866–3871.

[20] R. Unnikrishnan and M. Hebert, "Fast extrinsic calibration of a laser rangefinder to a camera," Robotics Institute, Tech. Rep. CMU-RI-TR-05-09, 2005.

[21] O. Jokinen, "Self-calibration of a light striping system by matching multiple 3-D profile maps," in *IEEE 3-D Digital Imaging and Modeling*, 1999, pp. 180–190.

[22] J. Levinson and S. Thrun, "Unsupervised calibration for multi-beam lasers," in *International Symposium on Experimental Robotics*, 2010.

[23] R. B. Rusu and S. Cousins, "3d is here: Point cloud library (pcl)," in *IEEE Int. Conf. Robotics and Automation (ICRA)*, 2011, pp. 1–4.

[24] S. Rusinkiewicz and M. Levoy, "Efficient variants of the ICP algorithm," in *IEEE 3-D Digital Imaging and Modeling*, 2001, pp. 145–152.

[25] C. M. Bishop *et al.*, *Pattern recognition and machine learning*. springer New York, 2006, vol. 4, no. 4.

[26] Y. A. Ioannou, "Automatic urban modelling using mobile urban lidar data," Master's thesis, Queen's University, 2010.