2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

Experimental Evaluation of 3D-LIDAR Camera Extrinsic Calibration

Subodh Mishra[1], Philip R. Osteen[2], Gaurav Pandey[3] and Srikanth Saripalli[1]

***Abstract*— In this paper we perform an extensive experimental evaluation of three planar target based 3D-LIDAR camera calibration algorithms, on a sensor suite consisting multiple 3D-LIDARs and cameras, assessing their robustness to random initialization and by using metrics like Mean Line Re-projection Error (MLRE) and Factory Stereo Calibration Error. We briefly describe each method and provide insights into practical aspects like ease of data collection. We also show the effect of noisy sensor on the calibration result and conclude with a note on which calibration algorithm should be used under what circumstances.**

***Index Terms*— Extrinsic Calibration, Non-Linear Least Square, 3D-LIDAR, Camera**

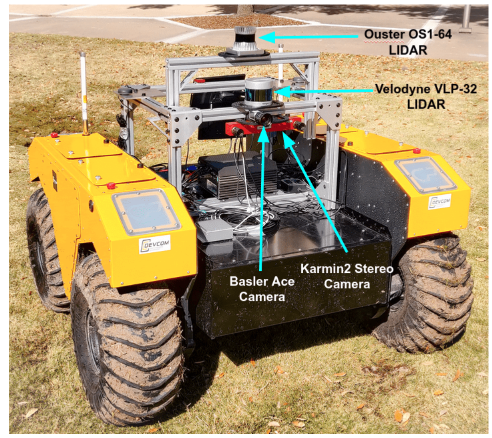


Fig. 1: **Experimental Platform:** Clearpath Robotics Warthog UGV with a). Ouster OS1 LIDAR, b). Velodyne VLP-32 LIDAR, c). Karmin2 Stereo Camera & d). Basler Ace Camera

I. Introduction

3D-LIDARs and cameras are ubiquitous to robots. Cameras provide color, texture and appearance information which LIDARs lack and LIDARs provide depth information which cameras lack. Modern multi-sensor perception and state estimation stacks depend on accurate calibration between sensors so that data from all sensors can be expressed in a common spatial frame of reference. Although the area of extrinsic calibration of 3D-LIDAR and camera has seen contributions from various robotics labs and research groups, a comprehensive work which analyzes different methods and provides experimental evaluation is lacking. In this work, we experimentally evaluate commonly used approaches for estimating 3D-LIDAR-to-camera extrinsic calibration, offering insights into the strengths and weaknesses of various formulations and providing interesting avenues for further work.

A. Literature Survey

Existing 3D-LIDAR camera extrinsic calibration algorithms can be broadly classified into target based [1], [2], [3], [4], [5], and targetless approaches [6], [7], [8], [9]. A target based approach requires a known object in the sensors' common Field of View (FoV) to ease requirements on data association and establish geometric constraints between features detected across sensors. While targetless approaches have the obvious advantage of not requiring any calibration target, most however need good initialization which usually comes from target-based methods. They also need reliable data association across modalities, which is still an open research problem.

The solution to target based 3D-LIDAR camera extrinsic calibration problem is inspired from target based 2D-LIDAR camera extrinsic calibration [10], [11], [12], [13], *etc*. The geometric constraint used in [11] is easy to use in 3D-LIDAR camera calibration scenario and has been exploited in the work presented in [1] & [2] and extended to 3D-LIDAR omnidirectional camera calibration in [3]. [4] present a 3D-LIDAR camera calibration technique in which the rotation matrix is estimated first and then a *point to plane* constraint (similar to ones in [1], [2], [3]) is used to determine the transformation parameters. [5] adds more geometric constraints by introducing line correspondences in addition to the previously used plane correspondences. [1], [2], [3], [4] and [5] use a planar target with a checkerboard pattern and require several observations from geometrically distinct view points. [14] and [15] present calibration methods that use a rigid plane with one and four circular perforations respectively. [16] presents a single shot calibration technique, but uses several checkerboard planes. In addition to the *point to plane* constraint that form the basis of methods described in [1], [2] and [3], the *point to back-projected plane* constraint has been exploited in works described in [12] and [13], but only for calibrating 2D-LIDAR camera systems. [17] exploits the *point to back-projected plane* constraint for cross calibrating 3D LIDAR camera pair. The methods described so far are pair-wise 3D-LIDAR camera calibration techniques. For robots with multiple cameras and LIDARs, joint calibration techniques like [18] and [19] have been found to be useful. Most target based 2D/3D-LIDAR camera extrinsic calibration methods, which use one or more planar surfaces, use checkerboards or ArUco [20] or AprilTags [21] for easy detection of planar target in the camera. In cases where such markers are not used, perforated [14] & [15] or spherical targets [22] are utilized.

[1]with the Department of Mechanical Engineering, Texas A&M University subodh514@tamu.edu
[2]with the Army Reasearch Lab, USA
[3]with the Ford Motor Company, USA

978-1-7281-6211-9/20/$31.00 ©2020 IEEE

B. *Contributions*

In this work, we experimentally evaluate three different planar target based 3D-LIDAR camera extrinsic calibration algorithms *viz.* [2], [17] and [19]. Our focus is to evaluate methods that only require a single, easy to build calibration target. Unlike [2] and [17] which are pair wise 3D-LIDAR camera calibration algorithms, [19] is a multi-sensor graph based optimization algorithm that jointly calibrates an arbitrary set of such sensors. We have evaluated these algorithm on the sensor suite shown in Figure 1, and have demonstrated the varying robustness of each approach to noisy sensor data. All three methods compared here use a single planar target (with known physical characteristics) as the calibration object.

II. 3D-LIDAR Camera Calibration

A. *The Problem*

For the pinhole projection model (π), the relationship between a homogeneous 3D point, P_i^L, and its image projection p_i^C, is given by

$$p_i^C = \pi(K[{}^C R_L, {}^C t_L] P_i^L). \tag{1}$$

Our goal is to estimate $[{}^C R_L, {}^C t_L]$, the extrinsic parameters that transform the lidar coordinate system to that of the camera, where ${}^C R_L \in SO(3)$ & ${}^C t_L \in R^3$. The camera intrinsics K is assumed to be known or estimated using established camera intrinsic calibration methods (e.g., [23]).

III. 3D-LIDAR Camera Extrinsic Calibration Algorithms

In this section we describe three different 3D-LIDAR camera calibration algorithms *viz.* **PPC-Cal** [2] (also implemented in [1] & [3]), **PBPC-Cal** [17] and **MSG-Cal** [19]. We will analyze the results of experimentally evaluating these methods in Section V.

A. ***PPC-Cal:*** *Point to Plane Constraint Calibration*

PPC-Cal is a widely used method, implemented in [1], [2] and [3]. A checkerboard pattern is printed on the planar target to facilitate the estimation of the target's plane parameters in the camera frame.

1) Data Collection: For the i^{th} observation of the planar target, the points $\{P_{im}^L\}$ on its surface in LIDAR frame can be detected by a RANSAC [25] based plane segmentation algorithm available with the Point Cloud Library (PCL) [26] and the plane parameters π_i^C in the Camera frame can be estimated by OpenCV's [27] checkerboard detection module. Given the i^{th} pose of the planar target, each P_{im}^L and π_i^C ($\pi_i^C = [n_i^C; d_i^C]$) pair satisfy a *point to plane* constraint (Equation 2) which involves $[{}^C R_L, {}^C t_L]$.

$$n_i^C.({}^C R_L P_{im}^L + {}^C t_L - d_i^C) = 0 \tag{2}$$

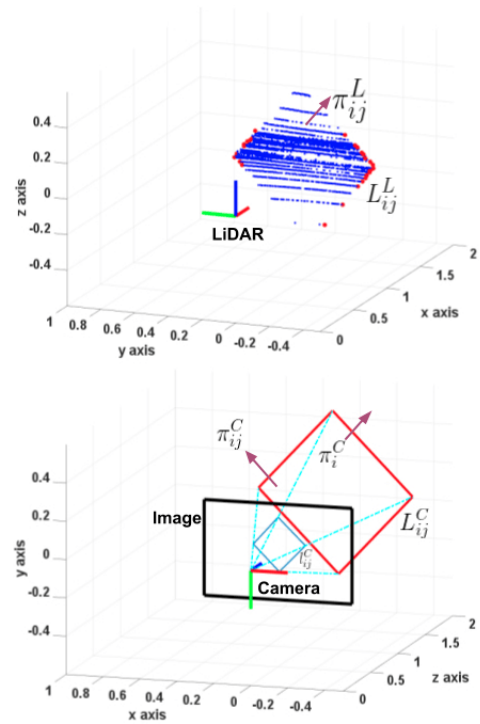


Fig. 2: **Notations:** A plane in 3D is parameterized as $\pi^F = [n_{3\times1}^F; d_{3\times1}^F]$, here n^F is the normal to the plane in the F frame of reference and d^F is the vector joining the origin of the F frame to the origin of the plane's frame in the F frame of reference. In LIDAR, the i^{th} pose of the planar target yields planar points $\{P_{im}^L\}$ (blue, where $m = \{1, 2, ..., p_i\}$) and boundary points $\{Q_{ijn}^L\}$ (red, where $j = \{1, 2, 3, 4\}$ & $n = \{1, ..., q_{ij}\}$, and q_{ij} is the number of points on j^{th} line). $\{P_{im}^L\}$ can be used to estimate π_i^L. In Camera, the i^{th} pose of the planar target yields lines l_{ij}^C (where $j = \{1, 2, 3, 4\}$) and planes π_i^C & π_{ij}^C (where $j = \{1, 2, 3, 4\}$). π_i^C is the parameterization of the plane defined by the planar target's surface and $\pi_{ij}^C (= [K^T l_{ij}^C; 0_{3\times1}])$ is the back-projected plane defined by the camera center and the line l_{ij}^C [24].

2) Optimization: The cost function formed by the *point to plane* constraint (Equation 2) is given in Equation 3.

$$P_1 = \sum_{i=1}^{M} \frac{1}{p_i} \sum_{m=1}^{p_i} \left\| (n_i^C)^T ({}^C R_L P_{im}^L + {}^C t_L - d_i^C) \right\|^2 \tag{3}$$

Here, p_i is the number of LIDAR points lying on the planar target in the i^{th} observation and M is the total number of observations. To obtain an estimate $[{}^C \tilde{R}_L, {}^C \tilde{t}_L]$, Equation 3 needs to be minimized with respect to $[{}^C R_L, {}^C t_L]$.

$$[{}^C \tilde{R}_L, {}^C \tilde{t}_L] = \underset{[{}^C R_L, {}^C t_L]}{\operatorname{argmin}} \; P_1 \tag{4}$$

We use ceres [28] to solve Equation 4. We need atleast 3 non-co-planar views ([3]) to solve the optimization problem (Equation 4) but in practice it is advisable to collect numerous observations to better constrain the optimization.

3) Remarks: We used about 30 observations in the experiments for each LIDAR camera pair. **PPC-Cal** requires only the planar points in LIDAR frame and plane parameters in camera frame to estimate $[{}^C R_L, {}^C t_L]$ and uses checkerboard for plane detection in camera. Therefore, data collection is easy and fast.

B. ***PBPC-Cal:*** *Point to Back-projected Plane Constraint Calibration*

PBPC-Cal has been implemented in [17] for 3D-LIDAR camera calibration. In addition to the *point to plane* constraint (Equation 2) used in **PPC-Cal**, **PBPC-Cal** uses a *point to back projected plane* constraint (Equation 5). This method requires detection of both the plane and the edges of the planar target, in both sensing modalities.

1) Data Collection: The planar $\{P^L_{im}\}$ and edge $\{Q^L_{ijn}\}$ points in LIDAR frame are detected using RANSAC based respective plane and line segmentation algorithms available in PCL. In the camera frame, the target's plane parameters π^C_i and the edge parameters l^C_{ij} are estimated using OpenCV's Line Segment Detector (LSD) [29]. Unlike **PPC-Cal**, instead of using a checkerboard for plane detection in camera frame, **PBPC-Cal** uses the points of intersection p^C_{ij} of edges l^C_{ij} and the known physical dimensions of the calibration target to solve a Perspective-n-Point (PnP) algorithm to estimate π^C_i. For the i^{th} pose of the planar target, each Q^L_{ijn} and l^C_{ij} pair satisfy the *point to back projected plane* constraint (Equation 5) which involves $[^C R_L, {}^C t_L]$.

$$n^C_{ij}.({}^C R_L Q^L_{ijn} + {}^C t_L) = 0 \tag{5}$$

Where $n^C_{ij} = K^T l^C_{ij}$ is the normal to the back-projected plane formed by the camera center and the line l^C_{ij} (Fig 2) and K is the camera instrinsic matrix.

2) Optimization: The cost function formed by *point to back projected plane* constraint (Equation 5) is given in Equation 6.

$$P_2 = \sum_{i=1}^{N}\sum_{j=1}^{4}\frac{1}{q_{ij}}\sum_{n=1}^{q_{ij}}\left\|(n^C_{ij})^T({}^C R_L Q^L_{ijn} + {}^C t_L)\right\|^2 \tag{6}$$

Here q_{ij} is the number of points lying on the j^{th} line in the i^{th} observation and N is the number of observations. To obtain an estimate $[{}^C\tilde{R}_L, {}^C\tilde{t}_L]$, Equation 6 needs to be minimized with respect to $[{}^C R_L, {}^C t_L]$

$$[{}^C\tilde{R}_L, {}^C\tilde{t}_L] = \underset{[{}^C R_L, {}^C t_L]}{\operatorname{argmin}}\ P_2 \tag{7}$$

In this method, Equation 4 is solved first and used to initialize Equation 7 which is solved using the ceres solver [28]. The *point to back-projected plane* constraint (Equation 5) is equivalent to the line correspondence equation given in [24] (2004, p. 180), the solution to which requires at least 6 noise free line correspondences [30] between the LIDAR and camera views. Since the planar target has 4 sides, theoretically we need at least 2 distinct views to solve this system but use of several frames is advised.

3) Remarks: We used about 30 observations in the experiments for each LIDAR camera pair. Compared to **PPC-Cal**, **PBPC-Cal** requires both planar points and the points lying on the edges of the target, in the LIDAR frame and therefore data collection is tedious as successful detection of all edges in LIDAR pointcloud depends on the way the target is held. This method requires the target to be held in a diagonal sense as shown in Figures 4 and 5 such that any edge of the target is not parallel to the scan lines of the LIDAR. Since this method doesn't use any fiducial marker like checkerboard or ArUco or AprilTag, the detection of planar target in image depends on OpenCV's Line Segment Detector (LSD) which may be affected by illumination.

C. ***MSG-Cal:*** *Multi-Sensor Graph based Calibration*

PPC-Cal and **PBPC-Cal** do pair-wise calibration of a 3D-LIDAR and camera system but **MSG-Cal** described in [19] adds another layer over pair-wise calibration of sensors by utilizing a graph based optimization paradigm to jointly calibrate several sensors. The first step involves pair-wise calibration of all the sensors present in the sensor suite and the second step involves a global optimization using g2o [31], a general framework for graph optimization. **PPC-Cal** and **PBPC-Cal** described previously can only cross calibrate 3D-LIDARs and cameras but **MSG-Cal** can directly cross calibrate across all pair-wise sensing modalities[1] except for a 2D-LIDAR with 2D-LIDAR, and can jointly calibrate any configuration of 3D-lidars, cameras, and 2D-lidars.

1) Data Collection: For LIDAR pointcloud, **MSG-Cal** uses PCL to make a model of the environment using the first frame (with no calibration target present), and when the target is introduced into the environment in subsequent frames, it is detected by background subtraction from the pre-built model. The result of background subtraction gives a dominant plane and many other points which may be sparse and random. With simple heuristics such as density of points and approximate size of the target, it is easy to filter out the dominant plane ($\pi^L = [n^L; d^L]$). An AprilTag pattern is used for detection of the planar target ($\pi^C = [n^C; d^C]$) in camera.

2) Pair-wise Calibration: For 3D-LIDAR↔camera, 3D-LIDAR↔3D-LIDAR and camera↔camera calibration the constraints are given by Equation 8 and Equation 10. For the i^{th} observation, the normal alignment constraint is given by Equation 8

$$n^C_i - {}^C R_L n^L_i = 0 \tag{8}$$

Then, a point lying on a planar surface satisfies Equation 9.

$$n^L_i.(P^L_{im} - d^L_i) = 0 \tag{9}$$

Using Equation 8 and Equation 9 in Equation 2 we have a modified version of the *point to plane* constraint (which can be called a *plane to plane* constraint as P^L_{im} has been eliminated),

$$n^C_i.{}^C t_L + n^L_i.d^L_i - n^C_i.d^C_i = 0 \tag{10}$$

Estimation of pair-wise SE(3) transformation parameters for *plane to plane* correspondences across sensors is done by minimizing a joint cost function (Equation 11) formed by Equation 8 and 10,

[1] *i.e.* 3D-LIDAR↔3D-LIDAR, 3D-LIDAR↔Camera, 3D-LIDAR↔2D-LIDAR, 2D-LIDAR↔Camera & Camera↔Camera

$$P_3 = \sum_{i=1}^{M} \left\| (n_i^C - {}^C R_L n_i^L) \right\|^2 + \sum_{i=1}^{M} \left\| (n_i^C . {}^C t_L + n_i^L . d_i^L - n_i^C . d_i^C) \right\|^2 , \quad (11)$$

where M is the number of observations. The minimization problem is given in Equation 12.

$$[{}^C \tilde{R}_L, {}^C \tilde{t}_L] = \underset{[{}^C R_L, {}^C t_L]}{\operatorname{argmin}} \; P_3 \quad (12)$$

3) Global Calibration: In this phase, a hypergraph composed of several node and edge types that exploit the pair-wise relative transforms as an initialization for the global sensor pose graph is constructed. The goal of the global graph approach is to incorporate all the information into a unified optimization structure, requiring a single optimization run to calibrate many sensors. The sensor poses are the unknowns that are estimated simultaneously in a global frame. In contrast to **PPC-Cal** and **PBPC-Cal**, **MSG-Cal** incorporates new global graph constraints for camera↔camera sensor pairs that incorporate the positions of individual AprilTags seen by multiple cameras. We collected about 100 observations for the experiments to ensure all the sensor pairs have sufficient detections.

4) Remarks: Like **PPC-Cal**, **MSG-Cal** needs only points lying on the planar target in LIDAR frame and uses an AprilTag for easy detection of the planar target in camera frame which makes data collection relatively easy.

IV. System Description

Our sensor suite (Figure 1) consists of an Ouster OS1 64 Channel LIDAR, a Velodyne VLP-32 LIDAR, a Basler Ace camera $[1600 \times 1200]$ and the Karmin2 Stereo Vision System (which comprises two Basler Cameras $[800 \times 600]$) such that the factory stereo calibration is known.

V. Experiments and Results

We compare the performance [2] of **PPC-Cal**, **PBPC-Cal** and **MSG-Cal** by using these methods to calibrate our sensor suite (Figure 1). First, we evaluate their robustness to random initial conditions (Figure 3) drawn from a zero mean normal distribution with standard deviation of 90° and 50 cm for rotation and translation respectively. We notice that **PPC-Cal** and **PBPC-Cal** are robust to initialization while **MSG-Cal** exhibits divergence in a few cases, in which the optimization arrives at the same incorrect local minima. Besides, we notice that all the methods converge to nearly the same rotational values but show variation in translation values. This is because the *point to plane* constraint (which is used in all the three methods) is good at constraining rotation but it needs several observations to constrain translation. The *point to back-projected plane* constraint (used in **PBPC-Cal**) helps translation estimation accuracy by providing additional constraints at each measurement. While we don't expect such bad initial guesses in practice (and initial guesses of Identity converged for each algorithm across multiple datasets), we are effectively showing how well each formulation constrains the optimization. Since the minimization problem(s) (Equation 4, 7, 12) solved to estimate $[{}^C R_L, {}^C t_L]$ are highly non-linear and involve parameters on manifolds, the convergence over several random initialization assures the user that the calibration process can be executed with any initial guess.

In the absence of ground truth we verify our algorithms by using the estimated parameters *a*) to compare it against the factory stereo calibration [3] and *b*) to project points lying on the edges of the planar target in LIDAR frame on the Camera image and calculate the mean line re-projection errors (MLRE).[4] MLRE is an independent evaluation metric since none of the methods we compare in this work use it as a residual in their respective optimizations.

In Table I the factory stereo calibration error is presented for the stereo camera and both the lidars. For Velodyne VLP-32, we can see that **PPC-Cal** and **MSG-Cal** show error in the order of 1 cm along the stereo baseline dimension (Z axis) as compared to 4.5 mm in **PBPC-Cal**. For Ouster, **PPC-Cal** shows better performance than the others. We can see that **MSG-Cal** gives the same error for both the lidars. This is because **MSG-Cal** is a graph based approach which does joint optimization of all the sensors together and also does camera↔camera pair-wise calibration. It is difficult to draw definitive conclusions by comparing only the stereo errors. Hence, we proceed to compare the MLRE in Table II and Figure 4.

From Table II it can be concluded that the **PBPC-Cal** performs best among all the three methods. If we compare **PPC-Cal** and **PBPC-Cal** we can see that the result of **PBPC-Cal** is consistent for all the sensor pairs, as expressed by a low standard deviation (0.089039 pixel) but **PPC-Cal** shows greater variation as evident from a high standard deviation (1.9723 pixel). As discussed in [17], the Ouster LIDAR is a noisy sensor and **PPC-Cal** doesn't perform well when the Ouster Lidar is used. **MSG-Cal** produced consistent results when using various M-estimators such as Huber and Tukey cost functions, as well as testing various confidence parameters of the Ouster sensor, where the confidence value of a sensor impacts both the pair-wise and global calibration steps. We can hypothesize that the graph based approach **MSG-Cal** which does joint optimization will have all its nodes affected by Ouster's noise and therefore gives poor performance as evident from a high reprojection error for all sensor pairs (Table II). To prove our hypothesis we re-calibrate the sensor suite using **MSG-Cal** but with the Ouster LIDAR removed and the results are presented in Table III and Figure 5.

From Figure 5 we can conclude that **MSG-Cal** shows

[2] blue: best performance, red: worst performance

[3] We use the estimated $T_L^{C_1}$ and $T_L^{C_2}$ and compare $T_L^{C_1}(T_L^{C_2})^{-1}$ with the given factory stereo calibration $T_{C_2}^{C_1}$

[4] MLRE is the average $\perp$ distance between $\{l_{ij}^C\}$ and $\{Q_{ijn}^L\}$ projected on the image using the estimated $[{}^C R_L, {}^C t_L]$

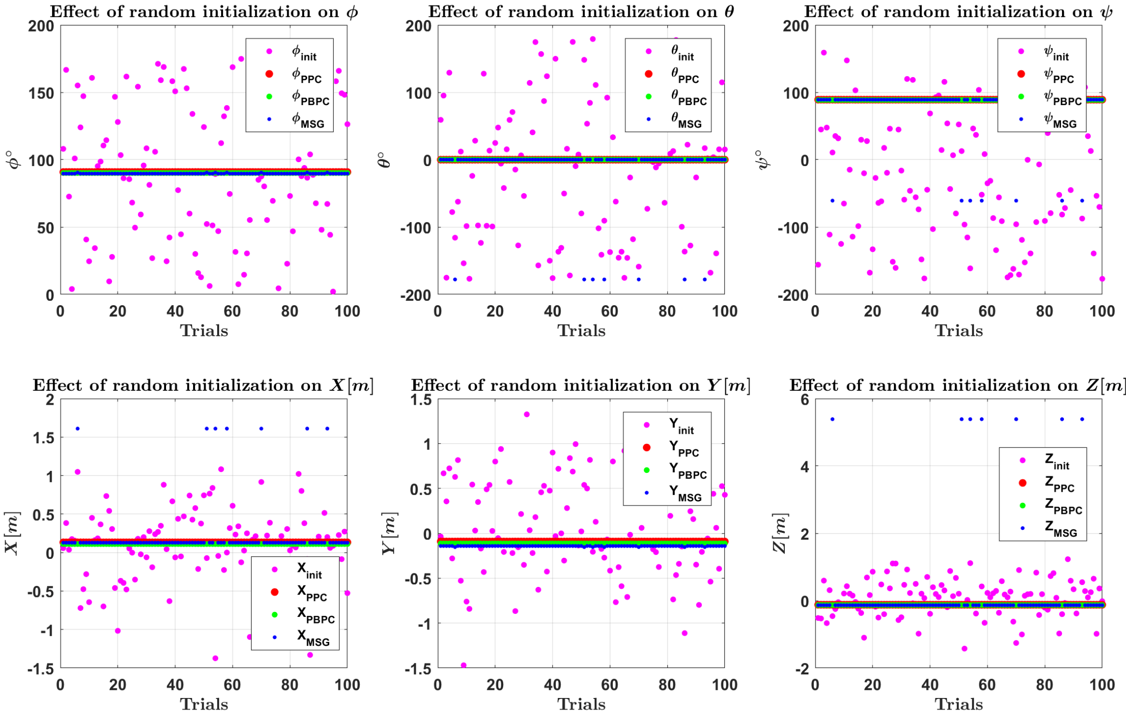
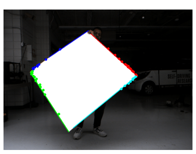
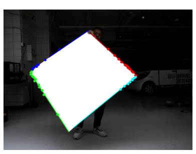


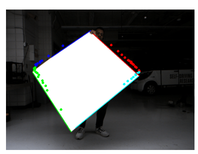
Fig. 3: Comparing the robustness of **PPC-Cal**, **PBPC-Cal** and **MSG-Cal** to random initialization. This figure shows the calibration result for the left stereo camera and Velodyne VLP-32 LIDAR under random initialization.



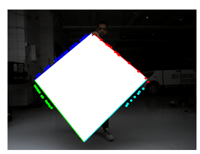
((a)) MLRE = 2.94664 with **PPC-Cal** for VLP-32 and Right Stereo Camera



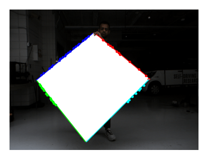
((b)) MLRE = 1.88719 with **PBPC-Cal** for VLP-32 and Right Stereo Camera



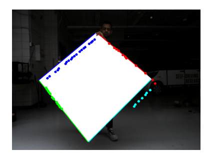
((c)) MLRE = 11.3415 with **MSG-Cal** for VLP-32 and Right Stereo Camera



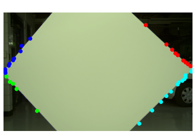
((d)) MLRE = 5.63206 with **PPC-Cal** for OS1-64 and Right Stereo Camera



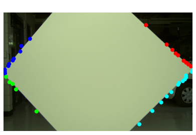
((e)) MLRE = 1.74138 with **PBPC-Cal** for OS1-64 and Right Stereo Camera



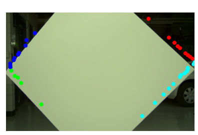
((f)) MLRE = 11.0735 with **MSG-Cal** for OS1-64 and Right Stereo Camera



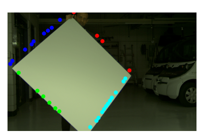
((g)) MLRE = 2.21383 with **PPC-Cal** for VLP-32 and Basler Ace Camera



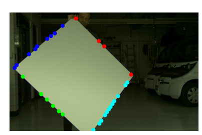
((h)) MLRE = 1.9423 with **PBPC-Cal** for VLP-32 and Basler Ace Camera



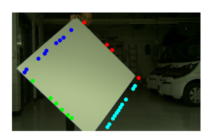
((i)) MLRE = 8.8254 with **MSG-Cal** for VLP-32 and Basler Ace Camera



((j)) MLRE = 6.95193 with **PPC-Cal** for OS1-64 and Basler Ace Camera



((k)) MLRE = 1.80414 with **PBPC-Cal** for OS1-64 and Basler Ace Camera



((l)) MLRE = 11.7995 with **MSG-Cal** for OS1-64 and Basler Ace Camera

Fig. 4: Comparing performance of **PPC-Cal**, **PBPC-Cal** and **MSG-Cal** using Mean Line Re-projection Error

	PPC-Cal		**PBPC-Cal**		**MSG-Cal**	
	VLP	**OS**	**VLP**	**OS**	**VLP**	**OS**
α°_{err}	-0.0055535	0.51756	0.19277	0.068189	0.10103	0.10103
β°_{err}	0.097271	0.037753	0.19995	0.12717	0.057334	0.057334
γ°_{err}	-0.081701	-0.061076	-0.12867	-0.22650	-0.11242	-0.11242
$X_{err}\ [m]$	0.00304	-0.00101	0.00640	-0.00507	-0.00113	-0.00113
$Y_{err}\ [m]$	-0.00439	-0.00102	-0.00352	-0.00622	-0.00377	-0.00377
$Z_{err}\ [m]$	0.01124	0.00877	0.00459	0.00475	0.01072	0.01072

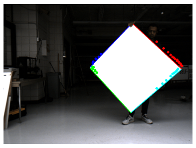
TABLE I: Errors with respect to factory stereo calibration for Velodyne VLP-32 LIDAR (**VLP**) / Ouster 64 Channel LIDAR (**OS**) and the stereo rig.

3D-LIDAR Camera Pair	MLRE		
	PPC-Cal	**PBPC-Cal**	**MSG-Cal**
VLP-32 ↔ Stereo Left	2.57316	1.94707	11.6547
VLP-32 ↔ Stereo Right	2.94664	1.88719	11.3415
OS1 ↔ Stereo Left	5.51552	1.76985	10.3954
OS1 ↔ Stereo Right	5.63206	1.74138	11.0735
VLP-32 ↔ Basler	2.21383	1.9423	8.8254
OS1 ↔ Basler	6.95193	1.80414	11.7995
Standard Deviation	1.9723	0.089039	1.1087

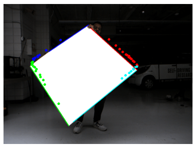
TABLE II: MLRE (in pixel) for various 3D-LIDAR Camera Pairs with **PPC-Cal**, **PBPC-Cal** and **MSG-Cal**

3D-LIDAR Camera Pair	MLRE		
	PPC-Cal	PBPC-Cal	MSG-Cal
VLP-32 ↔ Stereo Left	2.57316	1.94707	3.25161
VLP-32 ↔ Stereo Right	2.94664	1.88719	2.77772
VLP-32 ↔ Basler	2.21383	1.9423	2.5893

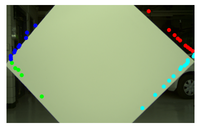
TABLE III: MLRE (in pixel) for various 3D-LIDAR Camera Pairs with **PPC-Cal**, **PBPC-Cal** and **MSG-Cal** without Ouster OS1 64 LIDAR



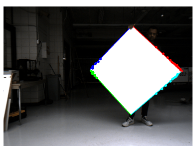
((a)) MLRE = 11.6547 with Ouster Lidar in the sensor suite for VLP-32 and Left Stereo Camera



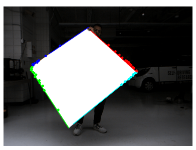
((b)) MLRE = 11.3415 with Ouster Lidar in the sensor suite for VLP-32 and Right Stereo Camera



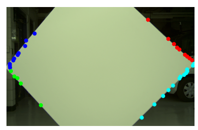
((c)) MLRE = 8.8254 with Ouster Lidar in the sensor suite for VLP-32 and Basler Ace Camera



((d)) MLRE = 3.25161 without Ouster Lidar in the sensor suite for VLP-32 and Left Stereo Camera



((e)) MLRE = 2.77772 without Ouster Lidar in the sensor suite for VLP-32 and Right Stereo Camera



((f)) MLRE = 2.5893 without Ouster Lidar in the sensor suite for VLP-32 and Basler Ace Camera

Fig. 5: Comparing performance of **MSG-Cal** with (Figure 5(a), Figure 5(b), Figure 5(c)) and without (Figure 5(d), Figure 5(e), Figure 5(f)) Ouster LIDAR in the sensor suite(Figure 1)

significant improvement when used in the absence of Ouster LIDAR and Table III conveys that without the Ouster in the graph optimization framework, the results of **MSG-Cal** are similar to those of **PPC-Cal** which makes sense because the pair-wise calibration in **MSG-Cal** uses similar constraints as **PPC-Cal**. Irrespective of Ouster LIDAR's presence or absence, **PBPC-Cal** performs the best.

3D-LIDAR Camera Pair	Solver Time (s)		
	PPC-Cal	PBPC-Cal	MSG-Cal
VLP-32 ↔ Stereo Left	10.7293	6.72805	0.738409996
VLP-32 ↔ Stereo Right	9.51054	8.15072	0.747992039
OS1 ↔ Stereo Left	7.56012	9.55696	0.59610796
OS1 ↔ Stereo Right	7.19404	5.07019	0.604768038
VLP-32 ↔ Basler	12.7641	10.9075	1.043144942
OS1 ↔ Basler	8.63144	3.27802	0.847215891

TABLE IV: Time taken (s) by solver to converge for various 3D-LIDAR Camera Pairs with **PPC-Cal**, **PBPC-Cal** and **MSG-Cal**

To complete the evaluation of the three methods we measure the execution time taken by the non linear least square solver to converge in all the three methods and present the result in Table IV. **MSG-Cal** noticeably takes significantly less time to converge when compared against **PPC-Cal** and **PBPC-Cal**. It is so because **PPC-Cal** and **PBPC-Cal** use lidar points directly to form the residuals of the optimization problem while **MSG-Cal** uses parametric representations derived from those points. Therefore the number of residuals used in the minimizer is significantly larger in **PPC-Cal** and **PBPC-Cal** and therefore they take longer to converge.

VI. Discussion

In this work we compared three LIDAR Camera extrinsic calibration algorithms *viz.* **PPC-Cal** ([1], [2] & [3]), **PBPC-Cal** [17] and **MSG-Cal** [19]. We presented the mathematical framework behind the working of all these methods and extensively evaluated them on a multi-sensor platform comprising 3 distinct cameras and 2 LIDARs. We concluded that **PPC-Cal** & **PBPC-Cal** were robust to random initialization for all trials while **MSG-Cal** diverged in a few trials (Figure 3). Nevertheless, barring a few cases in **MSG-Cal**, all the three frameworks can be initialized with any intial condition and will still converge. We showed that the **PPC-Cal** which uses only *point to plane* constraint shows deterioration in performance when a noisy sensor is used (Table II). The use of additional *point to back-projected* plane constraint in **PBPC-Cal** helps reduce the effect of noisy sensor by introducing more geometrical constraints to the non-linear cost function. We also showed that the global graph based optimization method **MSG-Cal**, which uses a variant of the *point to plane* constraint has all final pair wise calibrations (as evident from high MLRE from Table II) affected in the presence of a noisy sensor but gives comparable performance to **PPC-Cal** when the noisy sensor is removed (Table III). **PBPC-Cal** exhibits similar performance both with and without the noisy sensor and performs better than both **PPC-Cal** and **MSG-Cal** under all circumstances (Tables II & III, Figure 4). If we do not have a noisy sensor and need a quick calibration result then using **PPC-Cal** is a good option but if we have multiple sensors (with low noise) then **MSG-Cal** should be the algorithm of choice, as collecting data for both these methods is easier. Instead, if we have noisy sensors then **PBPC-Cal** should be used. In the future we want to use **PBPC-Cal** in the pair-wise calibration step of **MSG-Cal**, thus bringing the benefits of robust pair-wise calibration and joint global optimization together.

VII. Acknowledgements

The authors would like to thank Peng Jiang for his invaluable help with data collection.

References

[1] R. Unnikrishnan and M. Hebert, "Fast extrinsic calibration of a laser rangefinder to a camera," Tech. Rep., 2005.

[2] L. Huang and M. Barth, "A novel multi-planar lidar and computer vision calibration procedure using 2d patterns for automated navigation," in *2009 IEEE Intelligent Vehicles Symposium*, June 2009, pp. 117–122.

[3] G. Pandey, J. McBride, S. Savarese, and R. Eustice, "Extrinsic calibration of a 3d laser scanner and an omnidirectional camera," *IFAC Proceedings Volumes*, vol. 43, no. 16, pp. 336 – 341, 2010, 7th IFAC Symposium on Intelligent Autonomous Vehicles.

[4] L. Zhou and Z. Deng, "Extrinsic calibration of a camera and a lidar based on decoupling the rotation from the translation," in *2012 IEEE Intelligent Vehicles Symposium*, 2012, pp. 642–648.

[5] L. Zhou, Z. Li, and M. Kaess, "Automatic extrinsic calibration of a camera and a 3d lidar using line and plane correspondences," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 5562–5569.

[6] G. Pandey, J. R. McBride, S. Savarese, and R. M. Eustice, "Automatic extrinsic calibration of vision and lidar by maximizing mutual information," *Journal of Field Robotics*, vol. 32, no. 5, pp. 696–722, 2015.

[7] J. Levinson and S. Thrun, "Automatic online calibration of cameras and lasers," in *Robotics: Science and Systems*, 2013.

[8] D. Scaramuzza, A. Harati, and R. Siegwart, "Extrinsic self calibration of a camera and a 3d laser range finder from natural scenes," in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2007, pp. 4164–4169.

[9] Z. Taylor and J. Nieto, "Motion-based calibration of multimodal sensor extrinsics and timing offset estimation," *IEEE Transactions on Robotics*, vol. 32, no. 5, pp. 1215–1229, Oct 2016.

[10] S. Wasielewski and O. Strauss, "Calibration of a multi-sensor system laser rangefinder/camera," in *Proceedings of the Intelligent Vehicles '95. Symposium*, Sep. 1995, pp. 472–477.

[11] Qilong Zhang and R. Pless, "Extrinsic calibration of a camera and laser range finder (improves camera calibration)," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, vol. 3, Sep. 2004, pp. 2301–2306 vol.3.

[12] R. Gomez-Ojeda, J. Briales, E. Fernandez-Moral, and J. Gonzalez-Jimenez, "Extrinsic calibration of a 2d laser-rangefinder and a camera based on scene corners," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, May 2015, pp. 3611–3616.

[13] O. Naroditsky, A. Patterson, and K. Daniilidis, "Automatic alignment of a camera with a line scan lidar system," in *2011 IEEE International Conference on Robotics and Automation*, May 2011, pp. 3429–3434.

[14] S. A. Rodriguez F., V. Fremont, and P. Bonnifait, "Extrinsic calibration between a multi-layer lidar and a camera," in *2008 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, Aug 2008, pp. 214–219.

[15] M. Veĺas, M. Španěl, Z. Materna, and A. Herout, "Calibration of rgb camera with velodyne lidar," in *WSCG 2014 Communication Papers Proceedings*, vol. 2014, no. 22. Union Agency, 2014, pp. 135–144.

[16] A. Geiger, F. Moosmann, O. Car, and B. Schuster, "Automatic camera and range sensor calibration using a single shot," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 3936–3943, 05 2012.

[17] S. Mishra, G. Pandey, and S. Saripalli, "Extrinsic calibration of a 3d-lidar and a camera," in *IEEE Intelligent Vehicles Symposium*, 2020.

[18] Q. V. Le and A. Y. Ng, "Joint calibration of multiple sensors," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2009, pp. 3651–3658.

[19] J. L. Owens, P. R. Osteen, and K. Daniilidis, "Msg-cal: Multi-sensor graph-based calibration," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 3660–3667.

[20] S. Garrido-Jurado, R. Muñoz Salinas, F. Madrid-Cuevas, and R. Medina-Carnicer, "Generation of fiducial marker dictionaries using mixed integer linear programming," *Pattern Recognition*, vol. 51, 10 2015.

[21] E. Olson, "AprilTag: A robust and flexible visual fiducial system," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, May 2011, pp. 3400–3407.

[22] J. Kümmerle, T. Kühner, and M. Lauer, "Automatic calibration of multiple cameras and depth sensors with a spherical target," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2018, pp. 1–8.

[23] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000.

[24] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. New York, NY, USA: Cambridge University Press, 2003.

[25] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981.

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

[27] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[28] S. Agarwal, K. Mierle, and Others, "Ceres solver," http://ceres-solver.org.

[29] R. Grompone von Gioi, J. Jakubowicz, J. Morel, and G. Randall, "Lsd: A fast line segment detector with a false detection control," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 4, pp. 722–732, April 2010.

[30] B. Pribyl, P. Zemcík, and M. Cadík, "Pose estimation from line correspondences using direct linear transformation," *CoRR*, vol. abs/1608.06891, 2016.

[31] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "G2o: A general framework for graph optimization," in *2011 IEEE International Conference on Robotics and Automation*, 2011, pp. 3607–3613.