Extrinsic Calibration of a 3D-LIDAR and a Camera

Subodh Mishra[1], Gaurav Pandey[2] and Srikanth Saripalli[1]

Abstract— This work presents an extrinsic parameter estimation algorithm between a 3D LIDAR and a Projective Camera using a marker-less planar target, by exploiting Planar Surface Point to Plane and Planar Edge Point to back-projected Plane geometric constraints. The proposed method uses the data collected by placing the planar board at different poses in the common Field of View (FoV) of the LIDAR and the Camera. The steps include, detection of the target and the edges of the target in LIDAR and Camera frames, matching the detected planes and lines across both the sensing modalities and finally solving a cost function formed by the aforementioned geometric constraints that link the features detected in both the LIDAR and the Camera using non-linear least squares. We have extensively validated our algorithm using two Basler Cameras, Velodyne VLP-32 and Ouster OS1 LIDARs.

Index Terms— Extrinsic Calibration, LIDAR, Camera, Optimization

I. INTRODUCTION

Cameras and LIDARs help robots perceive the environment by providing complementary information. Cameras lack depth information which LIDARs provide and LIDARs lack color, texture and appearance information which cameras provide, hence systems using both these sensing modalities can compensate each others' weaknesses by leveraging respective advantages. Cameras and LIDARs are being used together for perception, multi-sensor state estimation, (semantic) mapping and localization. Cameras are better sensors when compared to LIDARs for place recognition and detecting loop closures but are affected by illumination, while LIDARs are immune to illumination changes and can provide dense metric maps. Simultaneous Localization and Mapping (SLAM) algorithms need both Cameras and LIDARs for robust state estimation. Hence, the knowledge of the extrinsic calibration is of paramount importance for fusing information from both these sensing modalities. Most perception and state estimation algorithms assume the extrinsic calibration to be known *a-priori*. This calls for methods to estimate the Euclidian 6 Degrees of Freedom (DoF) transformation between a LIDAR center and a Camera center using data generated by them.

II. RELATED WORK

In literature, the 3D-LIDAR Camera extrinsic calibration problem can be classified into two broad categories *a*) Target based approaches and *b*) Target less approaches. Target less approaches can be further divided into *i*) Scene based approaches and *ii*) Motion based approaches. Although target-less methods are convenient to use, they often rely on a
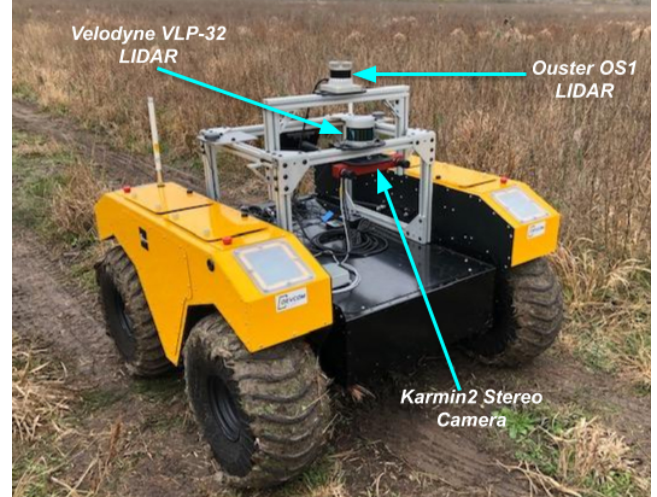
[1] with the Department of Mechanical Engineering, Texas A&M University subodh514@tamu.edu
[2] with the Ford Motor Company, USA

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

good initial guess, and are also sensitive to the calibration environment. In this paper we focus only on target based 3D-LIDAR Camera calibration because they are computationally light weight and provide good initialization for target less approaches if available. Target based approaches simplify the data association problem. In the works described in [1] and [2], the authors have to manually label correspondence between the camera and LIDAR. In this paper the feature association is done automatically.

Most target based 3D-LIDAR Camera extrinsic calibration methods draw their inspiration from 2D-LIDAR Camera extrinsic calibration techniques. One of the first 2D-LIDAR Camera calibration methods is presented in [3] which relies on observation of a planar checkerboard by both the sensors. The checkerboard gives the pose of the planar surface in the camera frame. The extrinsic parameters between the Camera and LIDAR is determined by solving a geometric constraint formed by projecting the LIDAR points on the checkerboard plane in the camera coordinate system. Observations from several view points need to be taken to ensure that all the DoF are observable. [4] calibrate a 2D LIDAR Camera system using a geometric constraint that doesn't require the knowledge of the planar target in the camera frame. It rather uses the back-projected plane (formed by a line detected on the image and the camera center in the camera frame) and a LIDAR point lying on the aforementioned line (that defines the back-projected plane) to form a geometric constraint. Several observations of the planar target are recorded and the constraint is solved using an algebraic approach which gives

a closed form solution. [5] calibrate a 2D-LIDAR Camera system using a V-Shaped target and determine the extrinsic parameters by minimizing the distance between the 3D features projected on the image plane and the corresponding image edge features. [6] presents a solution to 2D-LIDAR Camera calibration by introducing an additional line to plane constraint to the one used in [3]. The additional constraint is formed by the normal vector detected in the camera frame by checkerboard detection and the scan line direction vector on the checkerboard plane in the LIDAR frame. This additional constraint helps solve for the rotation parameters in a decoupled way. [7] use a target with orthogonal trihedrons using geometric point to back-projected plane constraint described [4] and the line to plane constraint described in [6].

The work by [3] was extended to a 3D-LIDAR and Camera system by [1] and [8] using the same geometric constraints. They differ only by the manner they have solved the constrained system of non-linear equations. [9] use similar geometric constraints established in [3] to calibrate a 3D-LIDAR and Omni-Directional Camera System. [10] present a 3D-LIDAR Camera calibration technique, much similar to the one presented in [6] in which the rotation matrix is first determined by line to plane constraint and then a point to plane constraint (similar to ones in [3], [1], [8], [9]) is used to determine the transformation parameters. [11] presents a new method for 3D LIDAR Camera Calibration by adding line to line constraints between lines detected in the 3D-LIDAR and the Camera in addition to the point to plane constraints already exploited in several works described above.

Most target based 2D/3D-LIDAR Camera extrinsic calibration methods, which use one or more planar surfaces, use checkerboards or ArUco or any other fiducial visual marker for easy detection of planar target in the camera frame. In cases where such markers are not used, V shaped [5], orthogonal trihedron shaped [7], perforated targets [12] or spherical targets [13] are employed.

In this work we use a simple planar surface without any visual markers for external calibration of the 3D LIDAR Camera system. To the best of our knowledge this is the first approach that calibrates a 3D LIDAR Camera system using a simple planar target without any visual marker(s) on them. The geometrical constraint due to back-projected plane has been exploited (in [4]) to solve the 2D LIDAR-Camera Extrinsic Calibration problem but we haven't come across any work which uses this constraint to calibrate 3D-LIDAR and Camera. We use the geometric constraint due to back-projected plane in our work. Our method requires the planar surface to be bounded by lines and not curves. Although any polygonal plane can be used, we use a square planar surface for the sake of simplicity.

Section III-A describes the notations used in the paper. Section III-B elucidates the feature extraction steps for both LIDAR & Camera data. Section III-C lays down the geometrical constraints formed by associated features between LIDAR & Camera and Section III-D depicts how the aforementioned constraints can form an Optimization problem. A brief system description is given in Section IV
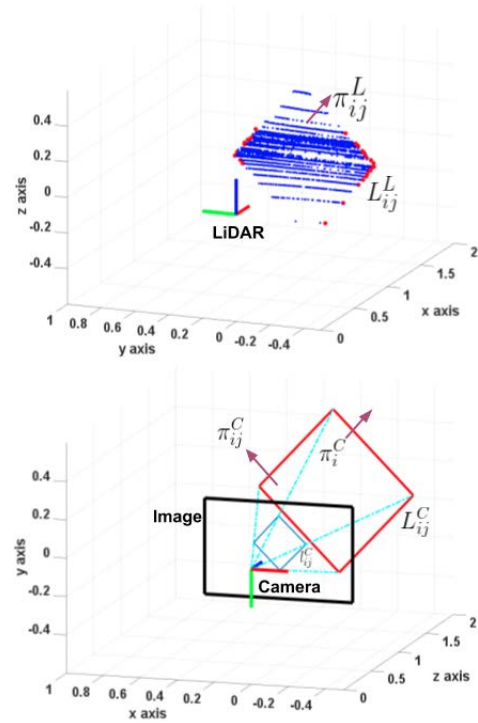

Fig. 2: **Notations:** In LIDAR, the ith pose of the planar target yields planar points $\{P^L_{im}\}$ and boundary points $\{Q^L_{ijn}\}$, where $j = \{1, 2, 3, 4\}$, which can be used to estimate π^L_{ij} and L^L_{ij} respectively. In Camera, the ith pose of the planar target yields lines l^C_{ij}, where $j = \{1, 2, 3, 4\}$, which can be used to calculate π^C_i, π^C_{ij} and L^C_{ij}.

which follows a Section V on Experiments and Results. We finally conclude in Section VI.

III. PROBLEM FORMULATION

For the usual pinhole camera model, the relationship between a homogeneous 3D point, P^L_i, and its homogeneous image projection p^C_i, is given by

$$p^C_i = K[{}^C R_L, {}^C t_L] P^L_i. \tag{1}$$

The extrinsic parameters that transform the laser coordinate system to that of the camera are captured by $({}^C R_L, {}^C t_L)$, where ${}^C R_L$ is the orthonormal rotation matrix parametrized by the Euler angles $[\phi, \theta, \psi]^\top$ and $t^C_L := [x, y, z]^\top$ is the translation vector. The camera intrinsics are captured by the matrix K and is assumed to be known or estimated using Zhang's method [14]. We use a planar target that is visible in both camera and lidar frame to establish the geometric constraints that allows us to estimate the rigid body transformation $({}^C R_L, {}^C t_L)$ between the two sensors (Figure 2).

A. Notations

The ith pose of the planar target in camera frame is parameterized as $\pi^C_i = [n^C_i; d^C_i]$, where n^C_i and d^C_i are the normal to the target's plane and it's distance from the origin of the camera frame of reference. The 4 edges (lines) of the planar target in the image are denoted as l^C_{ij}. Given the intrinsic camera calibration matrix K, the back-projected
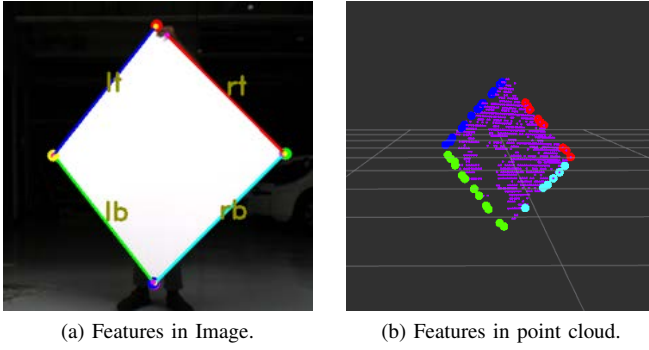
(a) Features in Image. (b) Features in point cloud.

Fig. 3: Result of Feature Detection in Image and LIDAR.

plane π_{ij}^C associated with each line l_{ij}^C is given by $\pi_{ij}^C = [K^T l_{ij}^C; 0]$ and hence, $n_{ij}^C = K^T l_{ij}^C$ [15]. The corresponding 3D equation (L_{ij}^C) of the line l_{ij}^C can be determined by computing the intersection of planes π_i^C and π_{ij}^C. The 3D corners P_{ij}^C of π_i^C can be obtained by intersection of π_i^C and the back-projected planes π_{ij}^C of two adjacent lines on image plane.

For the ith pose of the planar target detected in LIDAR, the normal n_i^L to π_i^L and centroid $\bar{P}_i^L$ can be calculated by using the points $\{P_{im}^L\}$ on π_i^L. Similarly, using points $\{Q_{ijn}^L\}$ on L_{ij}^L, the direction d_{ij}^L of L_{ij}^L and the centroid $\bar{Q}_{ij}^L$ can be obtained.

B. Feature Extraction

Lines (l_{ij}^C) in the image are detected using Line Segment Detector [16] available in OpenCV [17]. For a four sided polygon shown in Figure 2, the intersection of adjacent lines l_{ij}^C provide the corner points (p_{ij}^C) in the image plane. The known dimensions of the planar target are used to solve a PnP problem and π_i^C is obtained. To summarize, the features obtained from the image are the edges l_{ij}^C (Figure 3) of the planar target, the corners p_{ij}^C and the plane π_i^C.

Point cloud from LIDAR is filtered to remove points which have very low probability of lying in the common FoV of the camera and the LIDAR. Next, RANSAC [18] based plane segmentation (from Point Cloud Library (PCL) [19]) is used to obtain points $\{P_{im}^L\}$ lying on the planar target. A three-step approach is followed to detect the points lying along the boundaries of the planar target. First, we apply an edge detection algorithm [20] to the filtered point cloud to obtain an edge point cloud. Second, to remove outliers and enforce planarity, we apply plane segmentation to the edge point cloud and obtain a point cloud formed by points which lie on the planar target's boundaries (*viz.* boundary point cloud). Third, we apply a RANSAC based line detection algorithm (available in PCL [19]) to fit four lines to the boundary point cloud. As scan lines are parallel to the LIDAR's horizontal plane, any edge parallel to the LIDAR's horizontal plane will be difficult to detect. Therefore edge/line detection in LIDAR is easier when the planar target is held in a way similar to Figure 3. In summary, the features obtained from the LIDAR data are the points on the plane $\{P_{im}^L\}$ and the points on the

respective edges $\{Q_{ijn}^L\}$ of the planar target (Figure 3).

C. Constraints

Using the features extracted in the previous section, several geometric constraints can be established which involve the rotation $^C R_L$ and the translation $^C t_L$ between the LIDAR and Camera centers. Two such constraints pertinent to this work are discussed.

1) Point to Plane Constraint: Given the ith pose of the planar target, a point P_{im}^L lying on the planar target in the LIDAR coordinate frame and the plane parameters π_i^C satisfies the geometric constraint given in Equation 2.

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

As declared earlier, $\pi_i^C = [n_i^C; d_i^C]$. This is the standard point-to-plane projection constrain used by other target based methods as well [3].

2) Point to Back-projected Plane Constraint: This is the additional constrain that we introduce in this paper that helps in refining the calibration parameters. Given the ith pose of the planar target, a point Q_{ijn}^L lying on the jth edge of the planar target in the LIDAR coordinate frame and a corresponding line l_{ij}^C in the image satisfy the following geometric constraint:

$$n_{ij}^C \cdot (^C R_L Q_{ijn}^L +^C t_L) = 0 \qquad (3)$$

Where $n_{ij}^C = K^T l_{ij}^C$ is the normal to the back-projected plane formed by the camera center and the line l_{ij} (Fig 2).

Equations 2 and 3 are equations of the planar target and the back-projected plane respectively, in the *Hessian* normal form, in the camera frame.

D. Optimization

Cost functions for solving an optimization problem are formulated in the following ways using the geometrical constraints given in Equation 2 and 3:

1) Cost Function from Point to Plane Constraint: The cost function formed by Point to Plane Constraint is given in Equation 4.

$$P_1 = \sum_{i=1}^{M} \frac{1}{p_i} \sum_{m=1}^{p_i} \left\| (n_i^C)^T (^C R_L P_{im}^L +^C t_L - d_i^C) \right\|^2 \quad (4)$$

Here, p_i is the number of LIDAR points lying on the planar target in the ith observation and M is the total number of observations. To obtain an estimate $[^C \tilde{R}_L, ^C \tilde{t}_L]$, Equation 4 needs to be minimized with respect to $[^C R_L, ^C t_L]$.

$$[^C \tilde{R}_L, ^C \tilde{t}_L] = \underset{[^C R_L, ^C t_L]}{\operatorname{argmin}} \ P_1 \qquad (5)$$

2) Cost Function from Point to Back-projected Plane Constraint: The cost function formed by Point to Back-projected Plane Constraint is given in Equation 6.

$$P_2 = \sum_{i=1}^{N} \sum_{j=1}^{4} \frac{1}{q_{ij}} \sum_{n=1}^{q_{ij}} \left\| (n_{ij}^C)^T (^C R_L Q_{ijn}^L +^C t_L) \right\|^2 \quad (6)$$

1767

Here q_{ij} is the number of points lying on the jth line in the ith 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 $[^CR_L, ^Ct_L]$.

$$[^C\tilde{R}_L, ^C\tilde{t}_L] = \underset{[^CR_L, ^Ct_L]}{\operatorname{argmin}} P_2 \qquad (7)$$

We have found experimentally that the minimization problem given in Equation 7 is affected by the quality of initialization, especially of the rotational variables. The minimization problem given in Equation 5 converges to correct rotation parameters but it consistently converges to in-correct translation parameters, irrespective of the value of initialization. Therefore, we solve Equation 5 first and then use the results obtained to initialize the problem given in Equation 7 before solving it. We use the Ceres [21] non-linear least squares solver in our work.

As mentioned in [9], we need at-least 3 non-coplanar views to solve the optimization problem formed by Equation 5. This is ensured by monitoring the value of π_i^C which is obtained in the feature detection step.

The *Point to Back-projected Plane* constraint given in Equation 3 is equivalent to the line correspondence equation given in [15] (2004, p. 180). The solution to such a system of equation is given by the DLT-Lines method and requires at least 6 noise free line correspondences [22] between the LIDAR and camera views. Since our planar target has 4 sides, ideally, we need at least 2 distinct views to solve this system.

In practice it is advisable to get sufficient number of views, for both the constraints, because that ensures diversity of measurement and robustness of the solution.

The overall methodology is summarized in Algorithm 1.

Algorithm 1 Extrinsic Calibration using Plane and Line Correspondences

Input: N pairs of LIDAR and Camera Data containing the Planar Target in FoV

Output: Rigid transformation parameters $[R_L^C, t_L^C]$ from LIDAR to Camera

Algorithm Steps

1. Collect N pairs of LIDAR and Camera Data with Planar Target in FoV.

2. Detect the edges $\{l_{ij}^C\}$ of the Planar Target in Camera Images and use them to detect the Planar Target π_i^C in Camera Frame.

3. Detect points $\{P_{im}^L\}$ lying on the Planar Target in LIDAR data.

4. Detect points $\{Q_{ijn}^L\}$ lying on the edges of the Planar Target in LIDAR data.

5. Use the detected features to form the Cost Functions 4 and 6.

6. Solve the minimization problem given in Equation 5.

7. Use the solution obtained above to initialize and solve the minimization problem given in Equation 7.

IV. SYSTEM DESCRIPTION

Our setup (Figure 1) consists of an Ouster OS1 64 Channel LIDAR, a Velodyne VLP-32 LIDAR and two Basler cameras which come as Karmin2 Stereo rig such that the factory stereo calibration is known and can be used as one of the methods to test the veracity of the result produced by our algorithm. The data from all the sensors come at 10 Hz.

V. EXPERIMENTS AND RESULTS

In this section we use our algorithm on real world data collected using sensors shown in Figure 1. The four possible LIDAR-Camera setups are Velodyne VLP-32 LIDAR with left Basler Camera, Velodyne VLP-32 LIDAR with right Basler Camera, Ouster LIDAR with left Basler Camera and Ouster LIDAR with right Basler Camera.

A. Data Collection

As our sensors are not time synchronized, it is advisable to record data only when the planar target is at rest. We have a motion detection module which triggers feature detection only when the target is at rest for a user-defined duration.

1) Data Required for Solving Point to Plane Constraint: The data required for solving the minimization problem (Equation 5) formed by the constraint given in Equation 2 are: the points lying on the planar target $\{P_{im}^L\}$, the normal vector from the camera to the planar surface in camera frame n_i^C and the vector from the origin of the camera frame to the origin of the planar target d_i^C in the camera frame. Data is collected by placing the Planar Target at several positions and orientations in the common FoV of the sensor suite.

The Ouster OS1 LIDAR was found to be noisier than the Velodyne VLP-32 LIDAR, because the plane segmentation algorithm returned fewer planar points when the same RANSAC parameters were used for both the LIDARs. On an average, we received 2600 planar points from Velodyne VLP-32 while we got 1200 planar points from Ouster OS1, despite the latter being a 64 channel LIDAR.

2) Data Required for Solving Point to Back-Projected Plane Constraint: The data required for solving the minimization problem (Equation 7) formed by the constraint given in Equation 3 are: the points lying on the edges of the planar target in the LIDAR frame $\{Q_{ijn}^L\}$ and the normal n_{ij}^C to the back-projected plane formed by line l_{ij}^C detected in the camera frame.

While collecting data, it is essential to note that the orientation of the target is well within the thresholds which allows good edge detection. Since Velodyne VLP-32 has a non-linear distribution of scan rings about y-axis, the rings are concentrated near the origin and sparser as one pitches up or down (can be seen in Figure 5). Therefore, the planar pattern has to be held at a suitable height so that sufficient points from its surface and edges can be collected.

We collect about 30 frames from each sensor pair with planar target at different view points and not all of the frames were used in both the minimization problems.

B. Results

We compare the performance of our method with [9] for all the experimental setups. For calibrating using [9] we use a 6×9 checkerboard with each checker sized 0.061 x 0.061 m^2. In the absence of ground truth we verify our result by using the estimated parameters a) to project LIDAR points on camera image as shown in Figures 4 & 5 , b) to project points lying on the edges of the planar target in LIDAR frame on the Camera image and calculate the average line re-projection errors[1] as shown in Figures 6 & 7 and c) to compare it against the factory stereo calibration as shown in Tables I & II[2].

(a) Results with method presented in [9] for Left Camera and Ouster LIDAR

(b) Results with our method for Left Camera and Ouster LIDAR

(c) Results with method presented in [9] for Right Camera and Ouster LIDAR

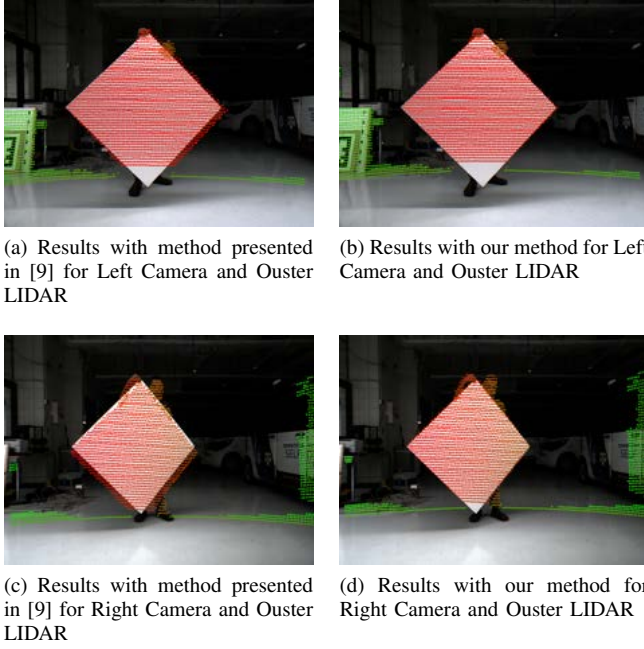(d) Results with our method for Right Camera and Ouster LIDAR

Fig. 4: Comparing [9] with our method for a setup comprising an Ouster 64 Channel LIDAR & Left (Figures 4a and 4b) and Right (Figures 4c and 4d) Basler Cameras of the sensor suite shown in Figure 1

Comparing Figures 4a, 4c, 5a, 5c with 4b, 4d, 5b, 5d respectively, the projection of LIDAR points with extrinsics estimated using [9] overshoot the planar target in the image plane but the projection shows perfect alignment when using our method, for all sensor pairs. Comparing Figures 6a, 6c, 7a, 7c with 6b, 6d, 7b, 7d respectively we can see that the average line re-projection errors are higher when using [9] as compared to our method, for all sensor pairs.

The errors in the estimated parameters with respect to the factory stereo calibration for Karmin2 Stereo Vision system for Ouster OS1 and Velodyne VLP-32 LIDARs are presented in Tables I and II respectively. For both the LIDARs, our method shows better performance for all the estimated parameters except for the angular error in β.

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

[2]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 $T_{C_2}^{C_1}$

(a) Results with method presented in [9] for Left Camera and Velodyne VLP-32 LIDAR

(b) Results with our method for Left Camera and Velodyne VLP-32 LIDAR

(c) Results with method presented in [9] for Right Camera and Velodyne VLP-32 LIDAR

(d) Results with our method for Right Camera and Velodyne VLP-32 LIDAR
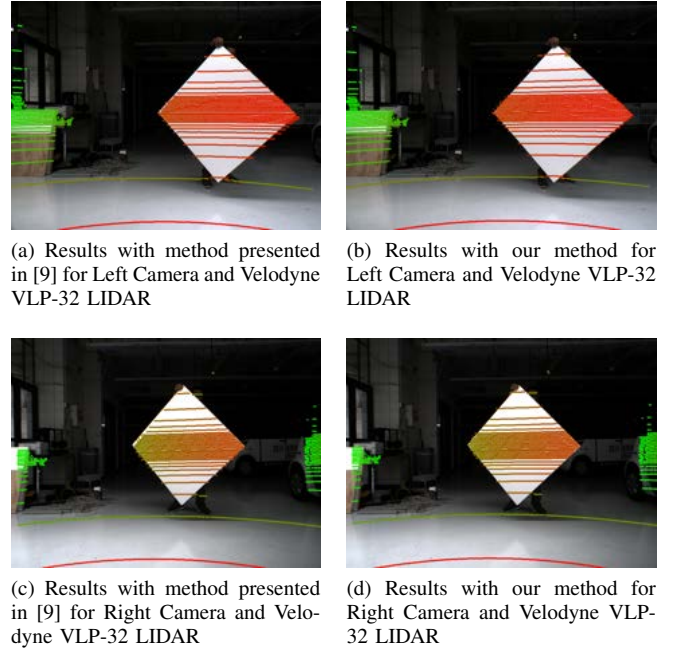
Fig. 5: Comparing [9] with our method for a setup comprising a Velodyne VLP-32 LIDAR & Left (Figures 5a and 5b) and Right (Figures 5c and 5d) Basler Cameras of the sensor suite shown in Figure 1

(a) Average line re-projection error of 7.01005 with method presented in [9].

(b) Average line re-projection error of 1.88829 with our method.

(c) Average line re-projection error of 9.64533 with method presented in [9].

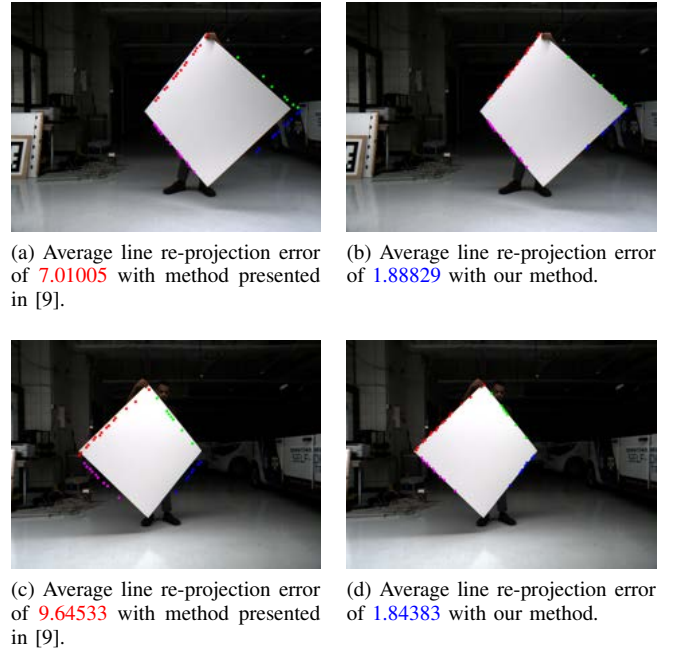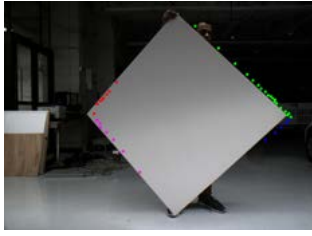(d) Average line re-projection error of 1.84383 with our method.

Fig. 6: Comparing [9] with our method for a setup comprising an Ouster 64 Channel LIDAR & Left (Figures 6a and 6b) and Right (Figures 6c and 6d) Basler Cameras of the sensor suite shown in Figure 1

Method	α_{err}°	β_{err}°	γ_{err}°	X_{err} [m]	Y_{err} [m]	Z_{err} [m]
[9]	-0.45112	3.4151	-0.52642	-0.06923	-0.01183	0.01334
Proposed	-0.063075	4.5844	0.19586	0.00134	-0.00736	0.00626

TABLE I: Errors with respect to factory stereo calibration for Ouster 64 Channel LIDAR and the stereo rig
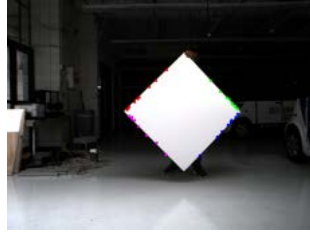
(a) Average line re-projection error of 3.43154 with method presented in [9].

(b) Average line re-projection error of 1.95631 with our method.

(c) Average line re-projection error of 3.0122 with method presented in [9].

(d) Average line re-projection error of 2.07706 with our method.

Fig. 7: Comparing [9] with our method for a setup comprising a Velodyne VLP-32 LIDAR & Left (Figures 7a and 7b) and Right (Figures 7c and 7d) Basler Cameras of the sensor suite shown in Figure 1

Method	α_{err}°	β_{err}°	γ_{err}°	$X_{err}\ [m]$	$Y_{err}\ [m]$	$Z_{err}\ [m]$
[9]	-1.1072	3.4868	-0.16217	-0.05363	-0.02342	0.00537
Proposed	-0.020025	4.4823	0.41197	0.00398	0.00268	-0.00205

TABLE II: Errors with respect to factory stereo calibration for Velodyne VLP-32 LIDAR and the stereo rig

If we compare the difference in average line re-projection errors between the method in [9] and our method for both Ouster OS1 and Velodyne VLP-32 in Figures 6 and 7 respectively, we find that it is much higher in the case of Ouster OS1 than Velodyne VLP-32. It is because the Ouster OS1 LIDAR is noisier than Velodyne VLP-32 according to our experiments described in Section V-A.1.

VI. CONCLUSIONS AND FUTURE WORK

In this work we showed how two different geometric constraints can be used together to estimate the extrinsic calibration of a 3D-LIDAR Camera system. We extensively used our algorithm across different LIDAR Camera pairs and the real world experiments show that the proposed method performed better than one presented in [9]. In the future we want to improve the feature detection by using learning based techniques and also explore Graph Based methods [23] for joint calibration of multiple sensors.

VII. ACKNOWLEDGEMENT

REFERENCES

[1] R. Unnikrishnan and M. Hebert, "Fast extrinsic calibration of a laserrangefinder to a camera," 07 2005.

[2] 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.

[3] 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.

[4] 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.

[5] K. Kwak, D. F. Huber, H. Badino, and T. Kanade, "Extrinsic calibration of a single line scanning lidar and a camera," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sep. 2011, pp. 3283–3289.

[6] L. Zhou, "A new minimal solution for the extrinsic calibration of a 2d lidar and a camera using three plane-line correspondences," *IEEE Sensors Journal*, vol. 14, no. 2, pp. 442–454, Feb 2014.

[7] 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.

[8] 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.

[9] 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. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1474667016350790

[10] L. Zhou and Z. Deng, "Extrinsic calibration of a camera and a lidar based on decoupling the rotation from the translation," 06 2012, pp. 642–648.

[11] L. Zhou, Z. Li, and M. Kaess, "Automatic extrinsic calibration of a camera and a 3d lidar using line and plane correspondences," 10 2018, pp. 5562–5569.

[12] M. Velas, M. Spanel, Z. Materna, and A. Herout, "Calibration of rgb camera with velodyne lidar."

[13] 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.

[14] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 11, p. 1330–1334, Nov. 2000. [Online]. Available: https://doi.org/10.1109/34.888718

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

[16] 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.

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

[18] 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. [Online]. Available: http://doi.acm.org/10.1145/358669.358692

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

[20] J. Levinson and S. Thrun, "Automatic online calibration of cameras and lasers," 06 2013.

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

[22] B. Pribyl, P. Zemcík, and M. Cadík, "Pose estimation from line correspondences using direct linear transformation," *CoRR*, vol. abs/1608.06891, 2016. [Online]. Available: http://arxiv.org/abs/1608.06891

[23] J. L. Owens, P. R. Osteen, E. Corporation, and K. Daniilidis, "Msg-cal: Multi-sensor graph-based calibration."