

Automatic Extrinsic Calibration of a Camera and a 3D LiDAR using Line and Plane Correspondences

Lipu Zhou, Zimo Li, and Michael Kaess

Abstract—In this paper, we address the problem of extrinsic calibration of a camera and a 3D Light Detection and Ranging (LiDAR) sensor using a checkerboard. Unlike previous works which require at least three checkerboard poses, our algorithm reduces the minimal number of poses to one by combining 3D line and plane correspondences. Besides, we prove that parallel planar targets with parallel boundaries provide the same constraints in our algorithm. This allows us to place the checkerboard close to the LiDAR so that the laser points better approximate the target boundary without loss of generality. Moreover, we present an algorithm to estimate the similarity transformation between the LiDAR and the camera for the applications where only the correspondences between laser points and pixels are concerned. Using a similarity transformation can simplify the calibration process since the physical size of the checkerboard is not needed. Meanwhile, estimating the scale can yield a more accurate result due to the inevitable measurement errors of the checkerboard size and the LiDAR intrinsic scale factor that transforms the LiDAR measurement to the metric measurement. Our algorithm is validated through simulations and experiments. Compared to the plane-only algorithms, our algorithm can obtain more accurate result by fewer number of poses. This is beneficial to the large-scale commercial application.

I. INTRODUCTION

Cameras and LiDARs provide complementary information of the environment. Cameras capture the color, texture and appearance information, and LiDARs supply the 3D structure information of the environment. Therefore, they are often jointly used to perceive the environment for the robots. Extrinsic parameters, consisting of rotation and translation, are prerequisite to fuse the information from the two sensors. The number of robots equipped with LiDARs and cameras has been increasing rapidly, with the widespread use of Unmanned Ground Vehicles (UGVs) and Unmanned Aerial Vehicles (UAVs). Thus, an effective and efficient extrinsic calibration algorithm is required for large scale commercial applications. This paper addresses the extrinsic calibration of a camera and a 3D LiDAR using 3D line and plane correspondences from a checkerboard. Our algorithm can estimate the extrinsic parameters from one pose of the checkerboard.

In the literature, one or more checkerboards are generally used to calibrate a camera and a 3D LiDAR [1, 2, 3, 4, 5, 6, 7]. As only the plane correspondence is adopted in the previous works, at least 3 poses of the checkerboard are

The authors are with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213, USA {lipuz, zimol, kaess}@andrew.cmu.edu

This work was partially supported by Autel Robotics under award number A020215.

needed in these algorithms. Generally, we can reduce the calibration error to a certain level by increasing the number of poses. In this paper, we aim to reduce the number of poses required in the extrinsic calibration process, which is important to large-scale commercial applications. For this purpose, we utilize the boundaries of the checkerboard. Lines are generally exploited for the extrinsic calibration between a camera and a 2D laser range finder [8, 9, 10, 11, 12, 13, 14]. In previous works, they generally use the laser points on the boundary and the corresponding 2D lines in the image to generate constraints on the extrinsic parameters. The resulting polynomial system is not easy to solve [11, 13, 14]. We use the boundaries of the checkerboard to yield 3D line correspondences between the LiDAR and the camera. The 3D line in the camera frame is obtained by calculating the intersection of the back-projected plane of a line and the checkerboard. The 3D line in the LiDAR coordinate system is estimated from the laser points on the boundary. Combining the plane and line correspondence, our algorithm is able to estimate the extrinsic parameters by one checkerboard pose.

Except for the number of poses, the diversity of the measurements is also important for an accurate result. Similar measurements may cause bias or overfitting. However, this problem is seldom considered in the previous works. In this paper, we discuss the equivalent measurements of our algorithm. This can guide the data collection process.

In some applications, such as object detection [15, 16], they only need the correspondences between the laser points and the pixels. The Euclidean metric is not important for them. We can use the similarity transformation to replace the rigid transformation for these applications. Using the similarity transformation, we can simplify the calibration process, as we do not need to measure the scale of the calibration target. This avoids the measurement error of the checkerboard size. Besides, 3D LiDAR needs a scale factor to transform its measurement to the metric measurement [3]. Using similarity transformation, we can refine the scale factor. Although similarity transformation has one more unknown, it can also be estimated from one pose of the checkerboard. The contributions of this paper are threefold:

Firstly, we propose a new extrinsic calibration algorithm for a camera and a 3D LiDAR using 3D line and plane correspondences. The new algorithm reduces the number of poses required by the extrinsic calibration to one. The experimental results show that our algorithm obtains more accurate results given the same number of poses. This is beneficial to industrial applications.

Secondly, we prove that parallel planar targets with parallel boundaries provide the same constraints in our algorithm. As the laser point can approximate the boundary more accurately for a closer checkerboard, we can place the checkerboard close to the LiDAR and camera without loss of generality.

Thirdly, we extend our algorithm to estimate the similarity transformation between the 3D LiDAR and the camera. The similarity transformation can be applied to the applications which only concern the correspondence between the laser point and the pixel [15, 16]. We experimentally show that a similarity transformation is a better alternative than the rigid transformation, because there exist inevitable measurement errors of the checkerboard size and the intrinsic LiDAR scale factors that transform its measurement to the metric measurement.

II. RELATED WORK

Checkerboard is widely used for the extrinsic calibration of various cameras and range sensors. Zhang and Pless [17] first introduced the checkerboard into the extrinsic calibration of a perspective camera and a laser rangefinder. They place a checkerboard with different poses in front of the laser rangefinder and the camera. The laser points on the plane together with the plane parameters estimated in the camera coordinate yield the plane-line correspondence. They use such correspondences to construct constraints on the extrinsic parameters. As Zhang and Pless only consider the linear constraints from the plane-line correspondences, their algorithm needs at least 5 poses. Theoretically, the extrinsic parameters can be recovered from 3 poses. Vasconcelos *et al.* [18] give solutions to this minimal problem. Unnikrishnan *et al.* [1] use the checkerboard to calibrate a perspective camera and a 3D LiDAR. Plane parameters are estimated in the 3D LiDAR frame and the camera frame. They use such plane-plane correspondence to decouple the initial calculation of rotation and translation, then refine the initial solution by jointly minimizing the point to plane distance. Their algorithm needs at least 3 poses. Pandey *et al.* [2] apply the checkerboard to solve the extrinsic calibration of a 3D LiDAR and an omnidirectional camera. Mirzaei *et al.* [3] also use a checkerboard to yield the plane-plane correspondence to calibration the intrinsic parameters of the LiDAR and the extrinsic parameters of the LiDAR and the camera. Zhou *et al.* [4] use the uncertainty of the normal vector estimated from the image to calculate the weight of the measurement. Their experiments show that farther poses generally have lower weights. As shown in this paper, moving the checkerboard is not necessary, thus the effect of the weight can be replaced by proper data collection scheme. Geiger *et al.* [5] place several checkerboards in the scene. Their algorithm detects the checkerboards and registers them with the planes in the laser point cloud. As there are several checkerboards in the scene, their algorithm only needs one shot of the scene. In [7], they exploit the Branch-and-Bound algorithm to extract the checkerboard in the laser point cloud. Checkerboard is also used in the extrinsic calibration of other

range sensors and cameras, such as stereo and laser range finder [19, 20], color and depth camera [21], and multi-planar LiDAR and a camera [22]. In these works, they generally only use the plane information of the checkerboard. They do not consider the boundary of the checkerboard.

Boundary of the object is generally exploited to calibrate a 2D laser rangefinder and a camera. In the literature, the v-shaped target [8, 9], right-angled triangulation board [10], and a board with intertwining black and white band [11] are used to generate such correspondence. Gomez-Ojeda *et al.* [12] apply an orthogonal trihedron to generate the plane-line and plane-point constraints to solve the extrinsic calibration problem. In [13], the line feature is also used to calibrate depth and color cameras. In [14], they present an algorithm that can recover the extrinsic parameters from one observation of a v-shaped target composed of two non-coplanar triangles with checkerboard inside. In these work, they generally use the line feature to generate 2D line-point [8, 9, 10] or plane-point (back-projected plane from 3D line) [11, 12, 14] or plane-line [12, 13, 14] constraints. The resulting polynomial system is not easy to solve [11, 13, 14]. In our work, we use the 3D line-line correspondence to generate constraints on the extrinsic parameters, which can be easily combined with the constraints from the plane-plane correspondence.

Some previous works seek to reduce or eliminate the dependence of a certain calibration target. Gao *et al.* [23] use an arbitrary trihedron to establish the plane-plane correspondences to solve the extrinsic calibration problem. Their algorithm needs at least two observations of the trihedron. Scaramuzza *et al.* [24] formulate the extrinsic calibration problem as a PnP problem [25] and manually generate the point-point correspondence. Manual operation is not practical for commercial applications. In [26, 27, 28], they exploit the stochastic relationship between the laser reflectivity and the image intensity of a nature scene to address the calibration problem. In [29], a deep neural is trained end-to-end to calculate the extrinsic parameters. In [30], they solve the extrinsic calibration problem by minimizing the SURF descriptor [31] error of the projection of the laser points among different frames. Generally, extrinsic calibration of a LiDAR and a camera needs to know the intrinsic parameters of the camera. Using a checkerboard, we can simultaneously calibrate the intrinsic parameters of the camera and the extrinsic parameters of the LiDAR and camera. Therefore, this paper studies the checkerboard as the calibration target and focuses on reducing the number of poses in the extrinsic calibration process.

III. PROBLEM DEFINITION AND NOTATIONS

Throughout this paper we use italic, boldfaced lowercase and boldfaced uppercase letters to represent scalars, vectors and matrices, respectively. The extrinsic calibration problem of a LiDAR and a camera is to estimate the relative rotation and translation between the two sensors. We denote the rotation matrix and the translation vector from the LiDAR frame to the camera frame as $\mathbf{R}_L^C \in SO(3)$ and \mathbf{t}_L^C , respectively.

In this paper, we use a checkerboard as the calibration target. The checkerboard plane and its four boundaries are exploited to estimate \mathbf{R}_L^C and \mathbf{t}_L^C , as demonstrated in Fig. 1.

Given the i th pose of the checkerboard, we can estimate the checkerboard plane π_i^C and its four boundaries l_{ij}^C ($j = 1, 2, 3, 4$) in image plane. Assume the intrinsic parameter of a camera is \mathbf{K} , and the homogeneous coordinates of the 2D line l_{ij}^C is \mathbf{l}_{ij}^C . The back-projected plane of l_{ij}^C is $[\mathbf{K}^T \mathbf{l}_{ij}^C; 0]$ [32]. Denote the parameterization of π_i^C and L_{ij}^C as $[\mathbf{n}_i^C; \mathbf{d}_i^C]$ and $[\mathbf{d}_{ij}^C; \mathbf{p}_{ij}^C]$, where \mathbf{n}_i^C is the normal of π_i^C , and \mathbf{d}_{ij}^C and \mathbf{p}_{ij}^C are the direction and a point of L_{ij}^C , respectively. \mathbf{n}_i^C and \mathbf{d}_{ij}^C are of length one. On the other hand, we can also estimate the checkerboard plane π_i^L and its four boundaries L_{ij}^L for the LiDAR. Denote the laser points on π_i^L as $\{\mathbf{P}_{im}^L\}$ ($m = 1, \dots, N_i$), and the points on L_{ij}^L as $\{\mathbf{Q}_{ijk}^L\}$ ($k = 1, \dots, K_{ij}$). Given $\{\mathbf{P}_{im}^L\}$ and $\{\mathbf{Q}_{ijk}^L\}$, we can calculate the normal \mathbf{n}_i^L and the centroid $\bar{\mathbf{P}}_i^L$ of π_i^L , and also the directions \mathbf{d}_{ij}^L and the centroid $\bar{\mathbf{Q}}_{ij}^L$ of L_{ij}^L in the LiDAR frame. In this paper, we exploit the correspondences $\pi_i^C \leftrightarrow \pi_i^L$ and $L_{ij}^C \leftrightarrow L_{ij}^L$ to establish constraints for \mathbf{R}_L^C and \mathbf{t}_L^C . We describe the 3D line and plane extraction for the camera and the LiDAR in section IV, and detail our calibration algorithm in section V.

IV. AUTOMATIC FEATURE EXTRACTION

We estimate the 3D information of the checkerboard plane and its boundaries from the image and the laser point cloud, as shown in Fig. 3.

For the LiDAR, we roughly figure out the area of the checkerboard. The laser points on the plane are detected by the RANSAC algorithm [33]. Then we find the boundary of each scan line. This gives the left and right boundaries of the checkerboard. We calculate the direction of the line defined by two consecutive points on the boundary. We further divide the left and right boundaries into two parts by finding the largest direction change. As the points on the boundary sometimes have large noise, we reduce the noise by the following steps demonstrated in Fig. 2. We first project all the laser points onto the plane. Next, we fit a line for each scan line. The boundary points are then projected to each scan line. Although the scan line should be a conic, here we use a line to approximate it, as the curvature of the conic is small, and the checkerboard is not large. Finally, we use the RANSAC algorithm to get rid of the potential outliers in these boundary points.

The planar parameters in the camera coordinate system can be easily computed. The checkerboard can be automatically detected in the image by the off-the-shelf software, such as MATLAB and OpenCV [34]. Then the planar parameters are calculated by the homography between the checkerboard and its image [32, 35]. We detect the lines in the image by the LSD algorithm [36]. Given the detected checkerboard, we detect the 4 lines enclosing the checkerboard as the 4 boundaries of the board. The corresponding 3D boundary is the intersection of the back-projected plane of the 2D boundary and the checkerboard plane. In the next section, we will show that our algorithm can estimate \mathbf{R}_L^C and \mathbf{t}_L^C

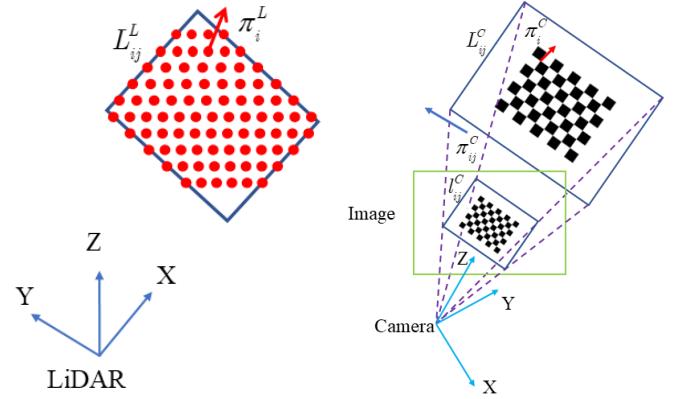


Fig. 1. Geometrical constraints. The i th pose of the checkerboard yields $\pi_i^C \leftrightarrow \pi_i^L$ and $L_{ij}^C \leftrightarrow L_{ij}^L$ ($j = 1, 2, 3, 4$). L_{ij}^C is the intersection line of plane π_i^C and the back-projected plane π_i^L of line l_{ij}^C .

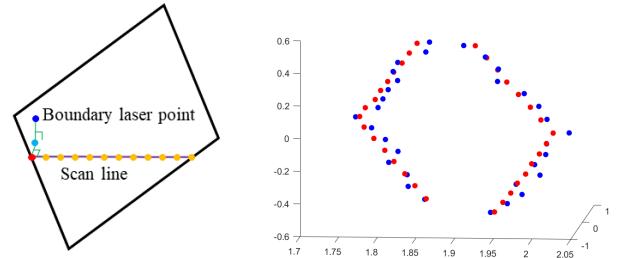


Fig. 2. Boundary laser points denoise. Left figure gives a schematic of our denoising approach. The boundary point is first projected onto the plane, then projected onto the scan line. Right figure demonstrates an example of denoising. Blue points are original points, and red points are the ones after denoising.

from N poses of the checkerboard using the aforementioned features.

V. EXTRINSIC CALIBRATION

In this section, we detail our extrinsic calibration algorithm for a LiDAR and a camera.

A. Geometric Constraints and Equivalent Set

Considering one line correspondence $L_{ij}^C \leftrightarrow L_{ij}^L$, we can obtain the constraints on \mathbf{R}_L^C and \mathbf{t}_L^C as follows:

$$\mathbf{R}_L^C \mathbf{d}_{ij}^L = \mathbf{d}_{ij}^C, \quad (1a)$$

$$(\mathbf{I} - \mathbf{d}_{ij}^C (\mathbf{d}_{ij}^C)^T) (\mathbf{R}_L^C \mathbf{Q}_{ijk}^L - \mathbf{P}_{ij}^C + \mathbf{t}_L^C) = \mathbf{0}_{3 \times 1}, \quad (1b)$$

where \mathbf{I} is the identity matrix, \mathbf{P}_{ij}^C is a point on the line in camera frame. One $L_{ij}^C \leftrightarrow L_{ij}^L$ gives four independent constraints on \mathbf{R}_L^C and \mathbf{t}_L^C . To simplify notation, we define $\mathbf{A}_{ij} = \mathbf{I} - \mathbf{d}_{ij}^C (\mathbf{d}_{ij}^C)^T$ in the following description.

Given one plane correspondence $\pi_i^C \leftrightarrow \pi_i^L$, we can have the following equations for \mathbf{R}_L^C and \mathbf{t}_L^C :

$$\mathbf{R}_L^C \mathbf{n}_i^L = \mathbf{n}_i^C, \quad (2a)$$

$$\mathbf{n}_i^C \cdot (\mathbf{R}_L^C \mathbf{P}_{im}^L + \mathbf{t}_L^C) + d_i^C = 0, \quad (2b)$$

where \cdot represents the dot product. One $\pi_i^C \leftrightarrow \pi_i^L$ gives three independent constraints on \mathbf{R}_L^C and \mathbf{t}_L^C .

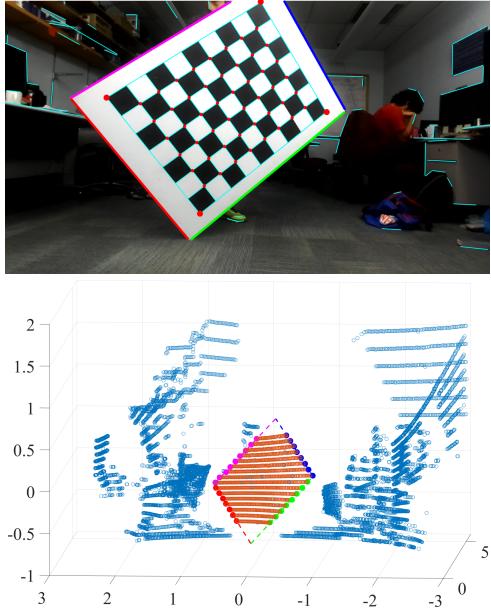


Fig. 3. Extract lines and planes from the image (up) and the laser point cloud (down). The line segments observed by the LiDAR and the camera can be different, even without overlap.

We define an equivalent set as the poses of the checkerboard providing the identical constraints on \mathbf{R}_L^C and \mathbf{t}_L^C in our algorithm. To increase the accuracy of the calibration result, we should increase the diversity of measurements and avoid equivalent measurements. For the planar target, we have the following proposition:

Proposition The parallel planar targets with parallel boundaries form an equivalent set for \mathbf{R}_L^C and \mathbf{t}_L^C estimation.

Proof: Firstly, we show that parallel lines give identical constraints on \mathbf{R}_L^C and \mathbf{t}_L^C . Suppose L_1 and L_2 are two parallel lines with direction \mathbf{d}^C in the camera frame. It is obvious that they give the same constraint as (1a). Assume that \mathbf{P}_1^C and \mathbf{P}_1^L are arbitrary two points on L_1 in the camera and the LiDAR frame, respectively. Let \mathbf{P}_2^C and \mathbf{P}_2^L represent the counterparts on L_2 . Let \mathbf{A} represent $\mathbf{I} - \mathbf{d}^C(\mathbf{d}^C)^T$. Using (1b) on L_1 and L_2 , we get

$$\begin{aligned} \mathbf{A}\mathbf{t}_L^C + \mathbf{A}(\mathbf{R}_L^C\mathbf{Q}_1^L - \mathbf{P}_1^C) &= \mathbf{0}, \\ \mathbf{A}\mathbf{t}_L^C + \mathbf{A}(\mathbf{R}_L^C\mathbf{Q}_2^L - \mathbf{P}_2^C) &= \mathbf{0}. \end{aligned} \quad (3)$$

In (3), we find the first parts of the two equations are the same. Additionally, we have

$$\begin{aligned} &\mathbf{A}(\mathbf{R}_L^C\mathbf{Q}_1^L - \mathbf{P}_1^C) \\ &= \mathbf{A}(\mathbf{R}_L^C(\mathbf{Q}_2^L + \mathbf{Q}_1^L - \mathbf{Q}_2^L) - (\mathbf{P}_2^C + \mathbf{P}_1^C - \mathbf{P}_2^C)) \\ &= \mathbf{A}(\mathbf{R}_L^C\mathbf{Q}_2^L - \mathbf{P}_2^C) + \mathbf{A}\mathbf{R}_L^C(\mathbf{Q}_1^L - \mathbf{Q}_2^L) - \mathbf{A}(\mathbf{P}_1^C - \mathbf{P}_2^C) \\ &= \mathbf{A}(\mathbf{R}_L^C\mathbf{Q}_2^L - \mathbf{P}_2^C). \end{aligned}$$

$\mathbf{A}\mathbf{R}_L^C(\mathbf{Q}_1^L - \mathbf{Q}_2^L)$ and $\mathbf{A}(\mathbf{P}_1^C - \mathbf{P}_2^C)$ cancel each other out, since they both represent the vector between and perpendicular to L_1 and L_2 , as demonstrated in Fig. 4. Therefore, L_1 and L_2 provide the identical constraints on \mathbf{R}_L^C and \mathbf{t}_L^C .

Secondly, parallel planes provide equivalent constraints on \mathbf{R}_L^C and \mathbf{t}_L^C . Suppose π_1 and π_2 are two parallel planes. Define $[\mathbf{n}^C; \mathbf{d}_1^C]$ and $[\mathbf{n}^C; \mathbf{d}_2^C]$ as the parameters of π_1 and

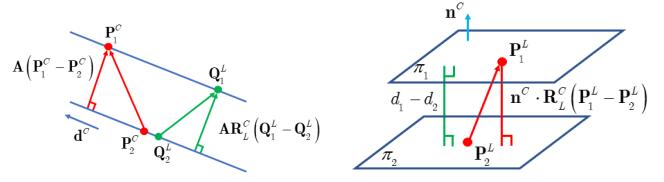


Fig. 4. A schematic of the relationship between two parallel lines (left) and two parallel planes (right). We use \mathbf{A} to represent the projection matrix $\mathbf{I} - \mathbf{d}^C(\mathbf{d}^C)^T$.

π_2 in the camera frame. Obviously, they yield the same constraints on \mathbf{R}_L^C as (2a). Assume \mathbf{P}_1^L and \mathbf{P}_2^L are two points on π_1 and π_2 in the LiDAR frame, respectively. Applying (2b) on π_1 and π_2 , we obtain

$$\begin{aligned} \mathbf{n}^C \cdot \mathbf{t} + \mathbf{n}^C \cdot \mathbf{R}_L^C \mathbf{P}_1^L + d_1^C &= 0, \\ \mathbf{n}^C \cdot \mathbf{t} + \mathbf{n}^C \cdot \mathbf{R}_L^C \mathbf{P}_2^L + d_2^C &= 0. \end{aligned} \quad (4)$$

In (4), the first parts of the two equations are the same. Moreover, we have

$$\begin{aligned} &\mathbf{n}^C \cdot \mathbf{R}_L^C \mathbf{P}_1^L + d_1^C \\ &= \mathbf{n}^C \cdot \mathbf{R}_L^C (\mathbf{P}_2^L + \mathbf{P}_1^L - \mathbf{P}_2^L) + d_2^C + d_1^C - d_2^C \\ &= \mathbf{n}^C \cdot \mathbf{R}_L^C \mathbf{P}_2^L + d_2^C + (\mathbf{n}^C \cdot \mathbf{R}_L^C (\mathbf{P}_1^L - \mathbf{P}_2^L) + d_1^C - d_2^C) \\ &= \mathbf{n}^C \cdot \mathbf{R}_L^C \mathbf{P}_2^L + d_2^C. \end{aligned}$$

We have the final equation, because the absolute value of $\mathbf{n}^C \cdot \mathbf{R}_L^C (\mathbf{P}_1^L - \mathbf{P}_2^L)$ and $d_1^C - d_2^C$ is the distance between the two planes, and they are of opposite signs, as demonstrated in Fig. 4. Therefore, π_1 and π_2 provide the identical constraints.

According to the above results, the parallel planar targets with parallel boundaries provide identical constraints on \mathbf{R}_L^C and \mathbf{t}_L^C . They are equivalent in our algorithm. ■

Although all the parallel planar targets with parallel boundaries are equivalent in theory, they are different in practice as laser points are discrete. They only approximate the ground truth boundaries. Therefore, the closer the checkerboard is, the better the approximation is. The above proposition ensures that simply rotating the checkerboard near the LiDAR can yield all the constraints, and moving the checkerboard is not necessary.

B. Pose Estimation from One Pose

One pose of the checkerboard can provide enough constraints for \mathbf{R}_L^C and \mathbf{t}_L^C . As a checkerboard has two pairs of parallel boundaries, one pose of the checkerboard provides 1 plane correspondence and 2 line correspondences according to the proposition mentioned above. As described in (1a) and (2a), three constraints only involving \mathbf{R}_L^C can be obtained. If we treat the normal and the two directions as three points, \mathbf{R}_L^C can be easily calculated according to [37]. For \mathbf{t}_L^C , we consider the constraints (1b) generated by the two non-parallel boundaries with direction \mathbf{d}_{i1}^C and \mathbf{d}_{i2}^C . The ranks of \mathbf{A}_{i1} and \mathbf{A}_{i2} are two. Additionally, the rows of \mathbf{A}_{i1} and \mathbf{A}_{i2} span the spaces perpendicular to \mathbf{d}_{i1}^C and \mathbf{d}_{i2}^C , respectively. Thus, the span of rows in \mathbf{A}_{i1} and \mathbf{A}_{i2} should have dimension 3, otherwise the two lines must be parallel. This contradicts the fact that the two lines are perpendicular.

Therefore, the two line correspondences are already adequate for \mathbf{t}_L^C estimation.

C. Pose Estimation from N Poses

Given N poses of the checkerboard, we can use (1a) and (2a) to get \mathbf{R}_L^C . In [1], they only use the normal constraint (1a). Here we add the direction constraint (2a). As \mathbf{n}_i^L is perpendicular to \mathbf{d}_{ij}^L , introducing the line correspondences significantly increase the diversity of the measurements. As the data have noises, we solve \mathbf{R}_L^C by minimizing the following cost function:

$$\tilde{\mathbf{R}}_L^C = \arg \min_{\mathbf{R}_L^C} \sum_{i=1}^N \sum_{j=1}^4 \left\| \mathbf{R}_L^C \mathbf{d}_{ij}^L - \mathbf{d}_{ij}^L \right\|^2 + \left\| \mathbf{R}_L^C \mathbf{n}_i^L - \mathbf{n}_i^C \right\|^2. \quad (5)$$

This problem has a closed-form solution [37]. Define

$$\begin{aligned} \mathbf{M}^L &= [\mathbf{n}_1^L, \mathbf{d}_{11}^L, \dots, \mathbf{d}_{14}^L, \dots, \mathbf{n}_N^L, \mathbf{d}_{N1}^L, \dots, \mathbf{d}_{N4}^L], \\ \mathbf{M}^C &= [\mathbf{n}_1^C, \mathbf{d}_{11}^C, \dots, \mathbf{d}_{14}^C, \dots, \mathbf{n}_N^C, \mathbf{d}_{N1}^C, \dots, \mathbf{d}_{N4}^C]. \end{aligned}$$

Assume that the Singular Value Decomposition (SVD) of $\mathbf{M}^L (\mathbf{M}^C)^T$ is $\mathbf{U} \mathbf{S} \mathbf{V}^T$. Then $\tilde{\mathbf{R}}_L^C = \mathbf{V} \mathbf{U}^T$ according to [37]

Given the estimation of \mathbf{R}_L^C , \mathbf{t}_L^C can be solved by the constraints 1(b) and 2(b). The points on the boundary are much fewer than the points on the plane. To avoid bias, we use the centroid of the plane $\bar{\mathbf{P}}_i^L$ and the centroids of the lines $\bar{\mathbf{Q}}_{ij}^L$ to estimate \mathbf{t}_L^C . Using (1b), (2b), and the definition of \mathbf{A}_{ij} mentioned above, we have

$$\begin{aligned} \mathbf{n}_i^C \cdot \mathbf{t}_L^C &= -\mathbf{n}_i^C \cdot \mathbf{R}_L^C \bar{\mathbf{P}}_i^L - d_i^C, \\ \mathbf{A}_{ij} \mathbf{t}_L^C &= -\mathbf{A}_{ij} (\mathbf{R}_L^C \bar{\mathbf{Q}}_{ij}^L - \mathbf{P}_{ij}^C). \end{aligned} \quad (6)$$

Stacking the above equations for all the poses, we get a linear system for \mathbf{t}_L^C . This is a linear least-squares problem and has a closed-form solution $\tilde{\mathbf{t}}_L^C$.

After obtaining the initial estimation $\tilde{\mathbf{R}}_L^C$ and $\tilde{\mathbf{t}}_L^C$, we jointly optimize them by minimizing the following cost function:

$$\begin{aligned} &\left(\hat{\mathbf{R}}_L^C, \hat{\mathbf{t}}_L^C \right) = \\ &\arg \min_{\mathbf{R}_L^C, \mathbf{t}_L^C} \sum_{i=1}^N \frac{1}{N_i} \sum_{m=1}^{N_i} \left\| \mathbf{n}_i^C \cdot (\mathbf{R}_L^C \mathbf{P}_{im}^L + \mathbf{t}) + d_i^C \right\|^2 + \\ &\sum_{i=1}^N \sum_{j=1}^4 \frac{1}{K_{ij}} \sum_{k=1}^{K_{ij}} \left\| \mathbf{A}_{ij} (\mathbf{R}_L^C \mathbf{Q}_{ijk}^L - \mathbf{P}_{ij}^C + \mathbf{t}_L^C) \right\|^2. \end{aligned} \quad (7)$$

We adopt Levenberg-Marquardt (LM) method [38] to solve this nonlinear optimization problem.

D. Similarity Transformation

In some applications, such as [15, 16], they only need the correspondence between the laser point and the pixel. The real Euclidean relationship between the two sensors is not necessary. Under this condition, we can use the similarity transformation to replace the rigid transformation. Using similarity transformation, we can simplify the calibration process, as we do not need to measure the physical size of the checkerboard. This in turn avoids the potential measurement error of gauging. Furthermore, there is a scale factor that

transforms the LiDAR data to the physical measurements [3]. Denote the distance measured by the k th shot of the i th laser beam as $\hat{\rho}_{ik}$. As shown in [3], the real distance ρ_{ik} to the object is calculated by

$$\rho_{ik} = \alpha_i (\hat{\rho}_{ik} + \rho_{oi}), \quad (8)$$

where α_i is the scale factor for the i th laser beam, and ρ_{oi} is the range offset. These parameters are provided by the manufactory. Here we use a common scale s to refine α_i . This is a similarity transformation.

The calculation of \mathbf{R}_L^C for the similarity transformation is the same as the rigid transformation. For the s and \mathbf{t}_L^C , we change (1b) and (2b) to

$$\begin{aligned} \mathbf{n}_i^C \cdot \mathbf{t}_L^C &+ \mathbf{n}_i^C \cdot \mathbf{R}_L^C \bar{\mathbf{P}}_i^L s = -d_i^C, \\ \mathbf{A}_{ij} \mathbf{t}_L^C &+ \mathbf{A}_{ij} \mathbf{R}_L^C \bar{\mathbf{Q}}_{ij}^L s = \mathbf{A}_{ij} \mathbf{P}_{ij}^C. \end{aligned} \quad (9)$$

s and \mathbf{t}_L^C can be estimated by solving the linear system. After we obtain the initial solution \tilde{s} , $\tilde{\mathbf{R}}_L^C$, and $\tilde{\mathbf{t}}_L^C$, we refine them by minimizing the cost function for the similarity transformation as

$$\begin{aligned} &\left(\hat{s}, \hat{\mathbf{R}}_L^C, \hat{\mathbf{t}}_L^C \right) = \\ &\arg \min_{s, \mathbf{R}_L^C, \mathbf{t}_L^C} \sum_{i=1}^N \frac{1}{N_i} \sum_{m=1}^{N_i} \left\| \mathbf{n}_i^C \cdot (s \mathbf{R}_L^C \mathbf{P}_{im}^L + \mathbf{t}) + d_i^C \right\|^2 + \\ &\sum_{i=1}^N \sum_{j=1}^4 \frac{1}{K_{ij}} \sum_{k=1}^{K_{ij}} \left\| \mathbf{A}_{ij} (s \mathbf{R}_L^C \mathbf{Q}_{ijk}^L - \mathbf{P}_{ij}^C + \mathbf{t}_L^C) \right\|^2. \end{aligned} \quad (10)$$

We summarize our algorithm in Algorithm 1.

Algorithm 1. Extrinsic calibration using plane and line correspondences from a checkerboard

Input: N checkerboard poses

Output: Rigid transformation or similarity transformation from the LiDAR to the camera.

1. Detect the checkerboard and its boundaries in the image, and calculate their corresponding 3D plane π_i^C and 3D boundaries L_{ij}^C , $j = 1, 2, 3, 4$
2. Specify a rough position for the checkerboard. Detect the checkerboard by RANSAC to get the plane π_i^L and the boundary L_{ij}^L , $j = 1, 2, 3, 4$.
3. Use (5) and (6) to get $(\tilde{\mathbf{R}}_L^C, \tilde{\mathbf{t}}_L^C)$, or use (5) and (9) to get $(\tilde{s}, \tilde{\mathbf{R}}_L^C, \tilde{\mathbf{t}}_L^C)$.
4. Refine the solution by minimizing (7) to get $(\hat{\mathbf{R}}_L^C, \hat{\mathbf{t}}_L^C)$ for the rigid transformation, or minimizing (10) to get $(\hat{s}, \hat{\mathbf{R}}_L^C, \hat{\mathbf{t}}_L^C)$ for the similarity transformation.

VI. EXPERIMENTS

As we mentioned in section II, the algorithms [1, 2, 4, 5] which only use the plane information have the similar formulation. We compare our algorithm with Unnikrishnan's algorithm [1] by synthetic and real experiments. Given the

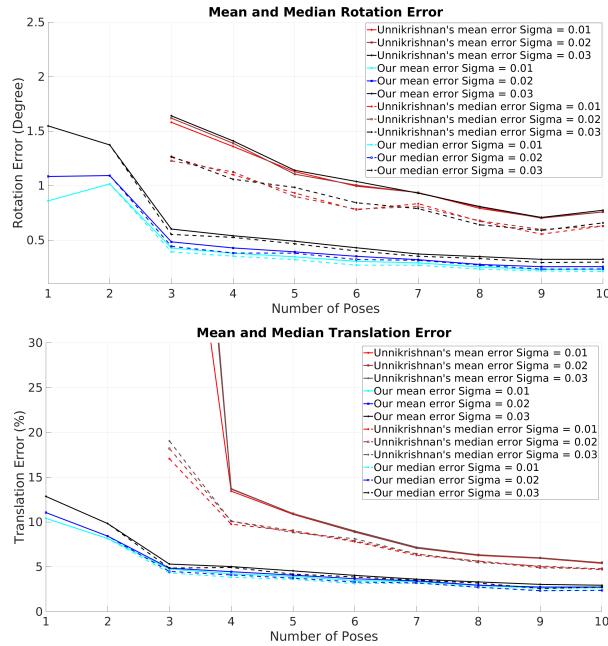


Fig. 5. Median and mean rotation (up) and translation (down) errors of 200 Monte-Carlo trials for each number of checkerboard poses.

estimation ($\hat{\mathbf{R}}, \hat{\mathbf{t}}$) and the ground truth (\mathbf{R}, \mathbf{t}), the rotation error is evaluated by the rotation angle of the angle-axis representation of $\hat{\mathbf{R}}\mathbf{R}^{-1}$ as in [39], and the translation error is calculated by $\|\hat{\mathbf{t}} - \mathbf{t}\|_2/\|\mathbf{t}\|_2$.

A. Synthetic Results

In the simulation, we randomly generate the configuration of LiDAR, camera and checkerboard poses. Specifically, the roll, pitch and yaw angle of the camera are within $\pm 45^\circ$, and the translation elements are uniformly distributed in $\pm 0.3m$ w.r.t the LiDAR. For the checkerboard frame, its x and y components of the translation are within $\pm 0.5m$, its z component is within $[1.5m, 2.5m]$, and its orientation is within $\pm 45^\circ$ relative to the camera. The checkerboard pose is generated in the view of the camera.

We add zero mean Gaussian noise to LiDAR and camera measurements to test the performance of different algorithms under various noise levels. The standard deviation of the LiDAR noise is set to $1cm$, $2cm$ and $3cm$, while the standard deviation of the image noise is fixed at 1 pixel. The number of checkerboard poses N is within $[1, 10]$. We run each algorithm 200 times for each N . Fig. 5 gives the results. It is clear that our algorithm has smaller rotation and translation errors under different noise levels compared to [1]. Additionally, our rotation and translation error are around 1.5° and 12% when only one pose is used and the standard deviation of the LiDAR noise is $3cm$. It verifies that our algorithm is able to provide accurate result using a single snapshot under large noise.

B. Quantitative Results

We use a Velodyne VLP-16 LiDAR and a ZED stereo camera to verify our algorithm, as shown in Fig. 6. The



Fig. 6. Velodyne VLP-16 LiDAR and ZED stereo camera used in our experiments.

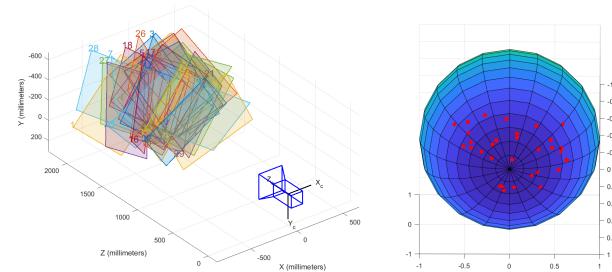


Fig. 7. Checkerboard poses (left) and their normals (right). The orientation of the checkerboard is well-distributed.

Velodyne VLP-16 LiDAR has 16 scan lines, $\pm 3cm$ range error and 360 degree horizontal and $\pm 15^\circ$ vertical field of view. The threshold of the RANSAC algorithm is set to $3cm$ in the experiments. The ZED stereo camera has about $12cm$ baseline and 1280×720 resolution.

As we do not have the ground truth of the extrinsic parameters of the LiDAR and the camera, we use the extrinsic parameters of the stereo camera to evaluate the performance of the algorithm as in [14]. Specifically, we estimate the extrinsic parameters of the LiDAR and the left camera ($\hat{\mathbf{R}}_{L_i}^{C_i}, \hat{\mathbf{t}}_{L_i}^{C_i}$), and the extrinsic parameters of the LiDAR and right camera ($\hat{\mathbf{R}}_L^{C_r}, \hat{\mathbf{t}}_L^{C_r}$), respectively. Then we compute the relative pose ($\hat{\mathbf{R}}_s, \hat{\mathbf{t}}_s$) between the left and right camera from $(\hat{\mathbf{R}}_{L_i}^{C_i}, \hat{\mathbf{t}}_{L_i}^{C_i})$ and $(\hat{\mathbf{R}}_L^{C_r}, \hat{\mathbf{t}}_L^{C_r})$. $(\hat{\mathbf{R}}_s, \hat{\mathbf{t}}_s)$ is compared with the stereo extrinsic parameters $(\mathbf{R}_s, \mathbf{t}_s)$ calculated by the MATLAB tool box. We also calculate the similarity transformation, and compute the error as the rigid transformation.

We collected 32 LiDAR and image pairs for the experiment, as shown in Fig. 7. N LiDAR and image pairs are randomly chosen from them. N is within $[1, 25]$ for our algorithm and $[3, 25]$ for Unnikrishnan's algorithm. For each $N \in [2, 25]$, we run the experiment 200 times. For $N = 1$, we estimate the extrinsic parameters for all the 32 poses of the checkerboard. Fig. 8 gives the result. It is obvious that our algorithm gives more accurate result. The result of our algorithm from 1 pose is comparable to the result of Unnikrishnan's algorithm when 6 poses are used. Using similarity transformation, we get very similar rotation error, but smaller translation error than using rigid transformation. This is because there exist inevitable measurement errors for the checkerboard size and the scale factor of the LiDAR in (8). These measurement errors mainly affect the estimation of the translation rather than the rotation, since the rotation matrix can be decoupled from the scale factor and the translation vector as shown in (5) and (9). Unnikrishnan's

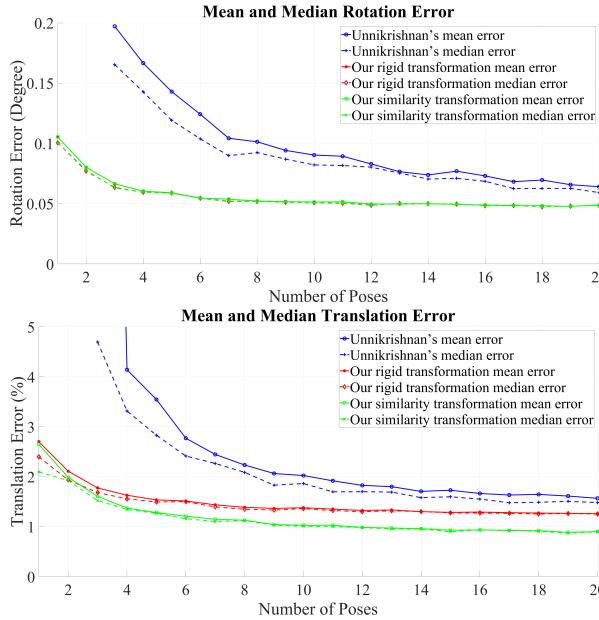


Fig. 8. Compare our algorithm with Unnikrishnan’s algorithm [1]. N poses are randomly chosen from 32 poses. We run the experiment 200 times for each $N \in [2, 25]$, and evaluate our algorithm for each pose when $N = 1$. Unnikrishnan’s result converges to our result as the number of checkerboard poses increases. 3D Line correspondences can significantly reduce the number of checkerboard poses required.

algorithm has large translation estimation error when $N = 3$. This is because there exist some planes with similar orientations, as shown in Fig. 7. These planes are degenerate for Unnikrishnan’s algorithm, but they are valid for our algorithm. The 3D line constraints significantly increase the diversity of the measurements. Therefore, our algorithm is more robust to the configuration of the poses and can get better result with fewer number of poses. This profits the large-scale commercial application.

C. Qualitative Results

Fig. 9 shows the extrinsic calibration results from one pose of the checkerboard. The boundaries of the checkerboard in the laser point cloud match the ones in the image. This verifies that our algorithm can provide accurate result even if only one pose is used. Fig. 10 shows the laser point back-projection results of our algorithm and Unnikrishnan’s algorithm using 3 poses. In this case, Unnikrishnan’s algorithm obtains poor results, as the configuration is degenerate for their algorithm. But our algorithm still gets accurate results. Fig. 11 shows the back-projection results from our similarity and rigid transformation using one pose. The results are very similar. Carefully observing the boundary, we find that the result from the similarity transformation is slightly better, as the projections of the boundary laser points are closer to the boundary in the image. As estimating the similarity transformation do not need the physical size of the checkerboard, this simplifies the calibration process and avoids the measurement error of the checkerboard size. This property is beneficial to the applications that do not consider the Euclidean metric information.

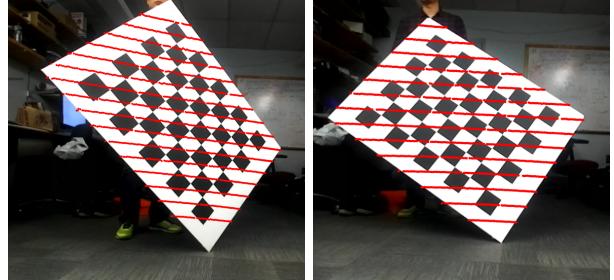


Fig. 9. Back-projection of the laser points using the extrinsic parameters from our algorithm using one pose.

- Unnikrishnan’s algorithm
- Our rigid transformation

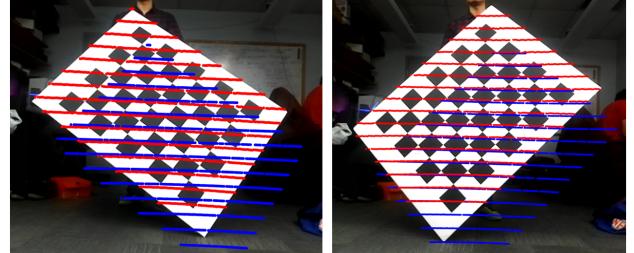


Fig. 10. Back-projection of the laser points using the extrinsic parameters from our algorithm (red dot) and Unnikrishnan’s algorithm (blue dot). Three poses are used in this example.

VII. CONCLUSION AND FUTURE WORK

In this paper, we present a new algorithm to solve the extrinsic calibration problem of a 3D LiDAR and a camera using the checkerboard. We introduce the 3D line correspondence into the original plane correspondence. This reduces the minimal number of poses required for this problem to one. We prove that the parallel planar targets with parallel boundaries are equivalent in our algorithm. This guarantees that we can obtain sufficient constraints by rotating the checkerboard near the LiDAR, where laser points approximate the real boundary better and have lower noise. We extend our algorithm to the similarity transformation that does not require the metric information of the checkerboard. This simplifies the calibration and avoids the measurement error of the checkerboard size. Experimental results show that our algorithm can get accurate result from one shot of the checkerboard. In addition, we can obtain better results with fewer poses compared to the plane-only algorithm. This is useful for the large-scale industrial application. Besides, we experimentally show that the similarity transformation is a better alternative to the rigid transformation if the application does not require the scale information.

For the future work, we plan to study the estimation of the LiDAR intrinsic parameters [3] by introducing the constraints on the boundary. We also plan to test our algorithm with different sets of sensors (different camera lens type and different 3D LiDAR).

REFERENCES

- [1] R. Unnikrishnan and M. Hebert, “Fast extrinsic calibration of a laser rangefinder to a camera,” 2005.
- [2] 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.

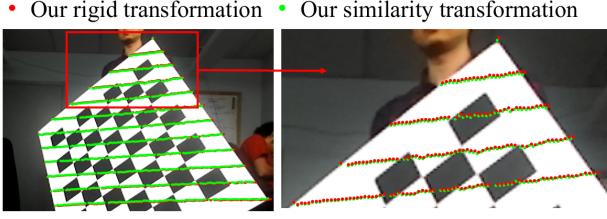


Fig. 11. Back-projection of the laser points using the similarity transformation (green dot) and the rigid transformation (red dot). One pose is used in this example. The two results are very similar. As shown in the enlarged part, the back-projection calculated from the similarity transformation is closer to the image boundary.

- Our rigid transformation
 - Our similarity transformation
- [3] F. M. Mirzaei, D. G. Kottas, and S. I. Roumeliotis, “3D LIDAR-camera intrinsic and extrinsic calibration: Identifiability and analytical least-squares-based initialization,” *The International Journal of Robotics Research*, vol. 31, no. 4, pp. 452–467, 2012.
- [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*, June 2012, pp. 642–648.
- [5] A. Geiger, F. Moosmann, . Car, and B. Schuster, “Automatic camera and range sensor calibration using a single shot,” in *2012 IEEE International Conference on Robotics and Automation*, May 2012, pp. 3936–3943.
- [6] L. Zhou, “A new minimal solution for the extrinsic calibration of a 2d lidar and a camera using three plane-line correspondences,” *IEEE Sens. J*, vol. 14, pp. 442–454, 2014.
- [7] A. Khosravian, T. J. Chin, and I. Reid, “A branch-and-bound algorithm for checkerboard extraction in camera-laser calibration,” in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, May 2017, pp. 6495–6502.
- [8] S. Wasielewski and O. Strauss, “Calibration of a multi-sensor system laser rangefinder/camera,” in *Intelligent Vehicles '95 Symposium, Proceedings of the*, Sep 1995, pp. 472–477.
- [9] 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*, Sept 2011, pp. 3283–3289.
- [10] G. Li, Y. Liu, L. Dong, X. Cai, and D. Zhou, “An algorithm for extrinsic parameters calibration of a camera and a laser range finder using line features,” in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2007, pp. 3854–3859.
- [11] 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.
- [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] A. Perez-Yus, E. Fernandez-Moral, G. Lopez-Nicolas, J. J. Guerrero, and P. Rives, “Extrinsic calibration of multiple RGB-D cameras from line observations,” *IEEE Robotics and Automation Letters*, vol. 3, no. 1, pp. 273–280, Jan 2018.
- [14] W. Dong and V. Isler, “A novel method for the extrinsic calibration of a 2-D laser-rangefinder a camera,” in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, May 2017, pp. 5104–5109.
- [15] J. Xu, K. Kim, Z. Zhang, H. W. Chen, and Y. Owechko, “2D/3D sensor exploitation and fusion for enhanced object detection,” in *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, June 2014, pp. 778–784.
- [16] J. Schlosser, C. K. Chow, and Z. Kira, “Fusing LIDAR and images for pedestrian detection using convolutional neural networks,” in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, May 2016, pp. 2198–2205.
- [17] Q. 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)*, vol. 3, Sept 2004, pp. 2301–2306 vol.3.
- [18] F. Vasconcelos, J. P. Barreto, and U. Nunes, “A minimal solution for the extrinsic calibration of a camera and a laser-rangefinder,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2097–2107, Nov 2012.
- [19] Y. Li, Y. Ruticheck, and C. Cappelle, “3D triangulation based extrinsic calibration between a stereo vision system and a LIDAR,” in *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, Oct 2011, pp. 797–802.
- [20] ———, “Extrinsic calibration between a stereoscopic system and a LIDAR with sensor noise models,” in *2012 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFIS)*, Sept 2012, pp. 484–489.
- [21] D. H. C., J. Kannala, and J. Heikkil, “Joint depth and color camera calibration with distortion correction,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 10, pp. 2058–2064, Oct 2012.
- [22] 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.
- [23] X. Gong, Y. Lin, and J. Liu, “3D LIDAR-camera extrinsic calibration using an arbitrary trihedron,” *Sensors*, vol. 13, no. 2, pp. 1902–1918, 2013.
- [24] 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.
- [25] Y. Zheng, Y. Kuang, S. Sugimoto, K. strm, and M. Okutomi, “Revisiting the PnP problem: A fast, general and optimal solution,” in *2013 IEEE International Conference on Computer Vision*, Dec 2013, pp. 2344–2351.
- [26] 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.
- [27] R. Wang, F. P. Ferrie, and J. Macfarlane, “Automatic registration of mobile LiDAR and spherical panoramas,” in *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, June 2012, pp. 33–40.
- [28] T. Scott, A. A. Morye, P. Pinis, L. M. Paz, I. Posner, and P. Newman, “Choosing a time and place for calibration of lidar-camera systems,” in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, May 2016, pp. 4349–4356.
- [29] N. Schneider, F. Piewak, C. Stiller, and U. Franke, “RegNet: Multi-modal sensor registration using deep neural networks,” in *2017 IEEE Intelligent Vehicles Symposium (IV)*, June 2017, pp. 1803–1810.
- [30] Z. Xiao, H. Li, D. Zhou, Y. Dai, and B. Dai, “Accurate extrinsic calibration between monocular camera and sparse 3D lidar points without markers,” in *2017 IEEE Intelligent Vehicles Symposium (IV)*, June 2017, pp. 424–429.
- [31] H. Bay, T. Tuytelaars, and L. Van Gool, “SURF: Speeded up robust features,” in *European conference on computer vision*. Springer, 2006, pp. 404–417.
- [32] R. Hartley and A. Zisserman, “Multiple view geometry in computer vision second edition,” Cambridge University Press, 2000.
- [33] M. A. Fischler and R. C. Bolles, “Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography,” in *Readings in computer vision*. Elsevier, 1987, pp. 726–740.
- [34] Itseez, “Open source computer vision library,” <https://github.com/opencv/opencv>, 2018.
- [35] Z. Zhang, “A flexible new technique for camera calibration,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, Nov 2000.
- [36] R. G. von Gioi, J. Jakubowicz, J. M. 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.
- [37] K. S. Arun, T. S. Huang, and S. D. Blostein, “Least-squares fitting of two 3-D point sets,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-9, no. 5, pp. 698–700, Sept 1987.
- [38] J. J. Moré, “The Levenberg–Marquardt algorithm: implementation and theory,” in *Numerical analysis*. Springer, 1978, pp. 105–116.
- [39] Z. Kukelova, J. Heller, and A. Fitzgibbon, “Efficient intersection of three quadrics and applications in computer vision,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 1799–1808.