Optimal Extrinsic Calibration between a Stereoscopic System and a LIDAR

You LI, Yassine Ruichek, Cindy Cappelle

Abstract—Current perception systems of intelligent vehicles not only make use of visual sensors, but also take advantages of depth sensors. Extrinsic calibration of these heterogeneous sensors is required for fusing information obtained separately by vision sensors and LIDARs (Light Detection And Ranging). In this paper, an optimal extrinsic calibration algorithm between a binocular stereo vision system and a 2D LIDAR is proposed. Most extrinsic calibration methods between cameras and a LIDAR proceed by calibrating separately each camera with the LIDAR. We show that by placing a common planar chessboard with different poses in front of the multi-sensor system, the extrinsic calibration problem is solved by 3D reconstruction of the chessboard and geometric constraints between the views from the stereovision system and the LIDAR. Furthermore, our method takes sensor noise into account that it provides optimal results under Mahalanobis distance constraints. To evaluate the performance of the algorithm, experiments based on both computer simulation and real data sets are presented and analyzed. The proposed approach is also compared with a popular camera/LIDAR calibration method to show the benefits of our method.

Index Terms—Intelligent vehicle, Computer vision, Sensors

I. INTRODUCTION

Intelligent vehicles as well as mobile ground robots always need to be armed with powerful perception systems. Such perception systems are usually composed of multiple heterogeneous sensors, like cameras, 2D or 3D LIDARs (Light Detection And Ranging). In [1] and [2] all the test vehicles joined in DARPA grand challenge were equipped with multiple vision cameras, 2D or 3D LIDARs. Visual sensors offer abundant information which are of great help to some key functions of perception systems, e.g. visual SLAM [3], visual detection and tracking [4], visual classification and recognition [5]. For range sensors, the detection area is always large, and they can acquire depth information of surroundings directly and quickly. Merely based on range sensors, like 2D or 3D LIDARs, a great number of techniques are also developed for similar functions [6], [7].

However, despite many impressive past progresses in either computer vision, or range vision, no technique based on single kind of sensors can perform reliably in complex environments. The visual sensors are always limited in narrow field of views (FOV) and require huge cost of computation for some complex algorithms. On the other hand, providing only depth information, the range sensors are not enough for more complex tasks. Hence, achieving effective and robust perception systems requires executing the fusion of different kinds of sensors. Fusion of visual and range data for intelligent vehicles has attracted more and more attention. In [8], [9], and [10], various multiple heterogeneous based fusion methods are proposed.

The problem of fusing data between visual cameras and LIDARs arises from the heterogeneity of the sensors. LIDAR data is about depth information in a certain scan plan, while the data of a camera can be viewed as a projection of a 3D world into an image plane. Thus, how to bridge the gap between a visual camera and a LIDAR is a preliminary problem to be solved before performing data fusion. A common solution is to map the data of LIDAR measurements into the camera coordinate system. In [11], [12], and [13], based on transforming measurements of LIDARs into camera coordinate system, proposed perception systems get better performances. In order to fuse precisely the two kinds of information in a same coordinate system, the geometric relationship in Euclidean space between these two sensors has to be known.

Extrinsic calibration between a LIDAR and a vision sensor can address this problem. In contrast to intrinsic calibration, extrinsic calibration calculates the rigid geometrical transformation (a $3 \times 3$ rotation matrix and a $3 \times 1$ translation vector) between two sensors’ coordinate systems. In general, it is implemented as the first step for further integration of data from different sensors. In [11], [12], and [13], the successful camera and LIDAR based fusion methods are all based on extrinsic calibration between the two types of sensors. This paper presents a novel extrinsic calibration method between a single layer LIDAR and a stereovision system. The proposed method is based on [14] and extended from [15]. The main contributions of the method are:

1) Extension from monocular camera based to binocular stereoscopic system based calibration. Instead of repeating a same calibration process from the LIDAR to each camera separately, we utilize the geometric connection between the two cameras and hence treat them as a whole.

2) Optimal 3D triangulation based extrinsic calibration. In order to compute the 3D plane equation as exactly as possible, an optimal 3D triangulation method, which outperforms the traditional middle point method, is adopted.

3) Optimal minimization of Mahalanobis distance. We firstly introduce sensor models of noises for LIDAR and cameras in the extrinsic calibration problem. We apply a Monte-Carlo method and a closed-form solution to
simulate the errors of the sensors, which are taken into account in the optimization process.

We will show that with these new features, the results of our method are better when compared to those provided by Zhang’s [16] popular extrinsic camera/LIDAR calibration method.

This paper is outlined as follows: Section II surveys several existing extrinsic calibration approaches between a LIDAR and a camera. Section III presents the geometric problem in extrinsic calibration. Section IV describes in detail the steps of the proposed calibration method. Section V shows experimental results using simulation and real data sets. In addition, a comparison between our method and Zhang’s [16] popular camera/LIDAR calibration method is also given. Finally, section VI concludes the paper and proposes some future works.

II. RELATED WORK

In literature, various proposed methods for extrinsic calibration between a LIDAR and a camera can be roughly divided into three categories:

1) Methods based on supplemental sensors. In [17], with the aid of an IMU (Inertial Measurement Unit) and a visible laser point, the Euclidean 3-dimensional transformation between a LIDAR and a camera is acquired by rotating the scan plane, where the angular information is estimated by the inertial sensor. Núñez et al. [18] also use an IMU to construct geometric constraints between a LIDAR and a camera system for calculating the transformation. However, being equipped with an IMU is a prerequisite for this kind of approaches.

2) Methods based on specially designed calibration boards. Li et al. [19] design a right-angled triangular checkerboard and employs the invisible intersection points of the LIDAR’s scan plane with the edges of the checkerboard to set up constraint equations. Rodriguez et al. [20] adopt a calibration board with two concentric circles where the inner circle is hollowed. The authors use the correspondence of the center of the inner circle between several different poses to achieve extrinsic calibration. By making use of special features in the designed calibration board, such approaches enable one to acquire the geometric relationship between a LIDAR and a camera. However, specially designed calibration boards impede themself for more widespread uses.

3) Methods based on common calibration chessboards. Zhang and Pless [16] firstly proposed a method utilizing an ordinary calibration chessboard which is widely used in camera calibration. This method is based on the fact that LIDAR measurements on the calibration chessboard satisfy certain geometric constraints with the plane of the chessboard. Following Zhang’s idea [16], Huang and Barth [21] utilize similar geometric constraints and generalize them into a situation of multi-layer LIDAR. Vasconcelos et al [22] improve Zhang’s method [16] by reducing the minimum number of various poses from 6 to 5. This kind of methods can handle with intrinsic calibration for cameras and extrinsic calibration between a LIDAR and a camera at the same time. Also, these methods require no extra equipments or specially designed calibration boards. Thus, in our point of view, they are more practical in real applications.

According to the above analysis, we prefer the third approach for its simplicity. Since stereovision can provide a complete three-dimensional view of a scene, its application for intelligent vehicles is more and more comprehensive nowadays. However, all the kinds of methods mentioned above are applied between a monocular camera and a 2D LIDAR. Calibrating a stereo vision system with a LIDAR can be performed by calibrating each camera with the LIDAR separately, as all existing methods perform, or more reasonably, by calibrating the considered stereovision system with the LIDAR, as proposed in this paper. Basing on 3D reconstruction of a common calibration board, we propose a novel calibration approach for obtaining the relative position and orientation between a stereo vision system and a LIDAR. It is worthy to mention that the advantages of our approach are threefold: Compared with the first and second categories of methods, our method is more convenient since it uses one standard chessboard; Compared with the third category, we will show that owing to the consideration of the whole stereovision system, the proposed extrinsic calibration algorithm performs more precisely. Moreover, the introduction of Mahalanobis distance contraints and the consideration of sensor noise models make the calibration results more accurate and robust.

III. PROBLEM FORMULATION

The setup and geometric models of the considered multisensor system are described below:

A. Multi-sensor System

Our experimental GEM vehicle is equipped with a stereoscopic system (Bumblebee XB3) mounted on the top of the vehicle and two 2D LIDARs (SICK LMS 221 and 291) fixed in the front face of the vehicle, one is mounted on the top, the other is installed at the bottom. Both of the LIDARs provide range measurements by a scanning plane of 180 degrees, with an angular resolution of 0.5 degree and a max range of 80m. The Bumblebee XB3 has three collinear cameras, with a distance of 0.12m apart from each other. Every image has a resolution of $1280 \times 960$. The left and right cameras are used as a stereoscopic system with a baseline 0.24m. Fig. 1 illustrates the experimental vehicle with the sensors. All of the sensors are fixed rigidly.

B. Coordinate Systems and Sensor Models

1) Coordinate Systems: To analyze such a multi-sensor system, we set several coordinate systems with respect to the different sensors. Let $\mathbb{R}^3_{\text{stereo}}, \mathbb{R}^3_{\text{left}}, \mathbb{R}^3_{\text{right}}, \mathbb{R}^2_{\text{left}}, \mathbb{R}^2_{\text{right}}$, and $\mathbb{R}^3_{\text{lidar}}$ be the coordinate systems attached to the stereoscopic system, the left and right cameras, the left and right image frames and the LIDAR respectively. Furthermore, since the stereoscopic system can be regarded as two separate cameras, we assume the left camera coordinate system as the reference of the stereoscopic coordinate system, hence, $\mathbb{R}^3_{\text{stereo}} = \mathbb{R}^3_{\text{left}}$. 
(we recall that the form of Cartesian coordinates:
\(X^T = (x, y, z)^T\). Suppose a point \(P\) in the image are given by:

\[
2) \text{Sensor Models: The left and right cameras are modeled by the classical pinhole model. Suppose a point } P \text{ in the left camera coordinate system: } P_l = (X_l, Y_l, Z_l)^T. \text{ Then, with zero skew, its corresponding projection coordinates } p_l = [x_l, y_l] \text{ in the image are given by:}
\]

\[
z \cdot [x_l \ y_l \ 1]^T = K^L \cdot P_l
\]

where \(K^L\) denotes the intrinsic matrix of the left camera, which is defined by the focal lengths \(f^L_x\) and \(f^L_y\) in pixels in x- and y-direction, principle point \((u^L_0, v^L_0)\), and a scale factor \(z\). For the right camera, the projection model is the same. We denote the intrinsic matrix of the right camera as \(K^R\).

For the LIDAR coordinate system, we specify the origin as the point which emits laser rays. The directions \(X, Y, Z\) are set as rightward, upward and forward from the sensor respectively. A LIDAR provides distance and direction of each scan point in polar coordinates \((\rho, \theta)\), which is converted into the form of Cartesian coordinates: \((X_{\text{lidar}}, Y_{\text{lidar}}, Z_{\text{lidar}})^T\).

3) Relationships: The right camera coordinate system \(\mathbb{R}^3_{\text{right}}\) is linked to the stereoscopic coordinate system \(\mathbb{R}^3_{\text{stereo}}\) (we recall that \(\mathbb{R}^3_{\text{stereo}} = \mathbb{R}^3_{\text{left}}\)) by a rigid transformation composed of a rotation matrix \(R^r\) and a translation vector \(T^r\). Suppose a point \(P\) in the right camera frame as: \(P_r = (X_r, Y_r, Z_r)^T\), then, its corresponding coordinates in the stereoscopic system frame are:

\[
P_{\text{stereo}} = P_l = R^l \cdot P_r + T^l
\]

In a similar way, a point in the stereoscopic coordinate system can be mapped into the right camera coordinate system as Eq. (2), by substituting \(R^l\) and \(T^l\) as:

\[
R^r_l = R^l T^r, T^r_l = -R^l T^r_l
\]

Let \((\Phi^l, \Delta^l)\) and \((\Phi^r, \Delta^r)\) be the 3D rigid transformation from the LIDAR coordinate system \(\mathbb{R}^3_{\text{lidar}}\) to the left and right camera coordinate systems \(\mathbb{R}^3_{\text{left}}\) and \(\mathbb{R}^3_{\text{right}}\) respectively, where \(\Phi^l\) and \(\Phi^r\) are orthogonal rotation matrices, \(\Delta^l\) and \(\Delta^r\) are translation vectors. Suppose a fixed point \(P\) observed by both the LIDAR and the stereoscopic system, and denoted as \(P_{\text{lidar}} = (X_{\text{lidar}}, Y_{\text{lidar}}, Z_{\text{lidar}})^T\) in the LIDAR’s coordinate system, \(P_l\) and \(P_r\) in the left and right camera coordinate systems. The three coordinates are connected by:

\[
P_l = \Phi^l \cdot P_{\text{lidar}} + \Delta^l
\]

\[
P_r = \Phi^r \cdot P_{\text{lidar}} + \Delta^r
\]

\[
P_l = R^l \cdot P_r + T^l
\]

All the sensor models and relationships between the multiple coordinate systems are depicted in Fig. 2.

4) Performance Evaluation: In practice, it is difficult to obtain the ground truth of the real extrinsic parameters between a camera and a LIDAR. In [16], [13], the extrinsic calibration methods are evaluated just by intuition (whether the projected laser points in images are reasonable or not). In this paper, we define an indicator \(\varepsilon\) as a precision measure of the multi-sensor calibration system:

\[
\varepsilon^2 = \frac{1}{L} \sum_{i=1}^{L} \|R^l_i \cdot (\Phi^r_i P^i_{\text{lidar}} + \Delta^r) + T^l_i - (\Phi^l_i P^i_{\text{lidar}} + \Delta^l)\|^2
\]

where \(L\) refers to the number of LIDAR points on the chessboard adopted in the extrinsic calibration. \(\varepsilon\) is a direct error with respect to the ground truth. Indeed, it can be viewed as a precision measure evaluating the conformity of the results with respect to the geometric relationships within the multi-sensor system. We hope \(\varepsilon\) to be as small as possible.

IV. 3D PLANE RECONSTRUCTION BASED EXTRINSIC CALIBRATION

A flowchart in Fig. 3 illustrates all the steps of our method. It requires putting a calibration chessboard in front of all the sensors at different positions and orientations and make sure that all the sensors can detect it. For each pose, we automatically extract the corner points of the chessboard in the images of the two cameras, and triangulate their 3D positions in the stereoscopic coordinate system. Then, from the reconstructed points, a PCA based method is used to estimate the 3D plane, which best fits the reconstructed points of the chessboard in the stereoscopic coordinate system. Finally, considering the geometrical constraints, a non-linear 3D minimization is carried out in order to calculate the extrinsic parameters. Our stereoscopic system delivers rectified stereo image pairs automatically, all the intrinsic/extrinsic parameters of the stereo-rig are calculated by [23] (implemented in [24]). Therefore, the intrinsic/extrinsic parameters \(K^L, K^R, R^l, T^l\), are known and \(K^L = K^R\).
A. Corner Points Triangulation

3D triangulation is a process of reconstructing 3D structure from corresponding points, giving the configuration parameters of the cameras. Unfortunately, correspondence errors are inevitable in practice, thus a middle point method is commonly adopted (Fig. 4(a)). However, this simple method doesn’t give best results. Both [25] and [26] proposed optimal triangulation methods, which aim to minimize:

\[ d(p_l, p'_l)^2 + d(p_r, p'_r)^2 \]  \( (6) \)

subject to the epipolar constraint:

\[ p_l^T F p_r' = 0 \]  \( (7) \)

where \( d(\ast, \ast) \) represents Euclidean distance, \( F \) is the fundamental matrix, \( p_l \) and \( p_r \) are the extracted corresponding points, \( p'_l \) and \( p'_r \) are the points after optimal correction. The 3D point position is computed from the after-correction corresponding points \( p'_l \) and \( p'_r \), as in Fig. 4(b). Here, we adopt the method of [26] for optimal correction.

We apply a sub-pixel corner points extraction [24] and an optimal correction algorithm [26] to get sets of corresponding corner point pairs \( p'_l \) and \( p'_r \). Triangulation corresponds to an intersection of rays \( \langle O_l, p'_l \rangle \) and \( \langle O_r, p'_r \rangle \) in space, as shown in Fig. 4(b).

Because only the 3D positions of the corresponding corner points are needed, we do not apply pixel-wise stereo disparity algorithm. Instead, a 3D triangulation algorithm similar to [27] is adopted for convenience. Let \( \tilde{p}_l' \) and \( \tilde{p}_r' \) be the normalized homogeneous coordinates of \( p'_l \) and \( p'_r \) in image coordinate system, corrected by the intrinsic parameters, respectively.

\[
\tilde{p}_l' = \left[ (p_{lx}' - c_x)/f, (p_{ly}' - c_y)/f, 1 \right] \\
\tilde{p}_r' = \left[ (p_{rx}' - c_x)/f, (p_{ry}' - c_y)/f, 1 \right] 
\]  \( (8) \)

where \( f, (c_x, c_y) \) are the focal length, principal point position in the calibrated intrinsic matrix \( K \), respectively. We have:

\[
Z_l \cdot \tilde{p}_l' = Z_r R_l^{12} \cdot \tilde{p}_r' + T_r^l 
\]  \( (9) \)

which yields to the following linear equation:

\[
\begin{bmatrix} -R_l^{12} \tilde{p}_r' & \tilde{p}_l' \end{bmatrix} \begin{bmatrix} Z_r \\ Z_l \end{bmatrix} = T_r^l 
\]  \( (10) \)

Let \( A = [-R_l^{12} \tilde{p}_r', \tilde{p}_l'] \) be a \( 3 \times 2 \) matrix. The least-squares solution for Eq. (10) is:

\[
\begin{bmatrix} Z_r \\ Z_l \end{bmatrix} = (A^T A)^{-1} A^T T_r^d 
\]  \( (11) \)

Let \( \alpha = -R_l^{12} \tilde{p}_r' \), then

\[
\begin{bmatrix} Z_r \\ Z_l \end{bmatrix} = \frac{1}{\|\alpha\|^2 \|p'_l\|^2 - \langle \alpha, p'_l \rangle^2} \begin{bmatrix} \|\alpha\|^2 \|p'_l\|^2 - \langle \alpha, p'_l \rangle^2 \\ -\langle \alpha, p'_l \rangle \end{bmatrix} \begin{bmatrix} \alpha^T \\ p^T_l' \end{bmatrix} T_r^d 
\]  \( (12) \)

where \( \langle \cdot, \cdot \rangle \) is the standard scalar product operator. From Eq. (12). The closed-form expressions for \( Z_l \) and \( Z_r \) can be induced:

\[
Z_l = \frac{\|\alpha\|^2 \langle \tilde{p}_l', T_r^l \rangle - \langle \alpha, \tilde{p}_l' \rangle \langle \alpha, T_r^l \rangle}{\|\alpha\|^2 \|p'_l\|^2 - \langle \alpha, p'_l \rangle^2} \\
Z_r \text{ has the similar formation as Eq. (13). Then, triangulated coordinates of a corner point pair } (p_l, p_r) \text{ in the left and right camera coordinate systems are: } \{l, r\} = Z_l, Z_r. \\

By triangulating all the corner point pairs extracted from the two views, the corresponding 3D coordinates attached to the left and right camera coordinate systems are acquired.
Although errors exist in the results of triangulation, the 3D points cloud is nearly flat. The next step is to estimate a 3D plane that best fits all the reconstructed 3D points.

B. 3D Plane Estimation

A plane in 3D Euclidean space can be defined by a known point on the plane and a normal vector perpendicular to its surface. In homogeneous coordinate system, the equation of the plane can be expressed as:

\[ \vec{n} \cdot [X, Y, Z, 1]^T = 0 \quad (14) \]

where \( \vec{n} \) is a 1 \times 3 normal vector, \( d = -\vec{n} \cdot [X_0, Y_0, Z_0]^T \), \( [X_0, Y_0, Z_0] \) is a known point on the plane.

Given a set of 3D points \( S = \{P_1, \ldots, P_i, \ldots, P_N\} \), the evaluation of the best fitting plane to \( S \) is a classic linear regression problem in mathematics. Either Least Squares Fitting (LSF) or Principal Component Analysis (PCA) \([28]\) can effectively solve this problem. We choose PCA method for its simplicity, detailed comparisons can be found in \([29]\).

1) Covariance matrix construction. We calculate the covariance matrix of \( S \) by:

\[ \text{Cov}(S) = \frac{1}{N-1} \begin{bmatrix} X - \bar{X} & Y - \bar{Y} & Z - \bar{Z} \\ Y - \bar{Y} & \bar{X} & \bar{Y} \\ Z - \bar{Z} & \bar{Y} & \bar{Z} \end{bmatrix} \quad (15) \]

where \( \bar{X}, \bar{Y} \) and \( \bar{Z} \) are the mean coordinates of all 3D points in \( S \), and \( N \) is the number of points in \( S \).

2) Spectral Decomposition. The covariance matrix \( \text{Cov}(S) \) is a 3 \times 3 real-valued symmetric square matrix, which can be diagonalized by the Spectral Decomposition (SD):

\[ \text{Cov}(S) = \mathbf{V} \mathbf{D} \mathbf{V}^T \quad (16) \]

where \( \mathbf{V} \) is a 3 \times 3 matrix that has orthogonal columns so that \( \mathbf{V}^T \mathbf{V} = 1 \), and \( \mathbf{D} \) is a 3 \times 3 diagonal matrix with eigenvalues on the diagonal in decreasing order: \( \lambda_1 \geq \lambda_2 \geq \lambda_3 \geq 0 \).

3) Principle component analysis. In fact, the columns of \( \mathbf{V} \) are the eigenvectors of \( \text{Cov}(S) \). According to \([28]\), the eigenvector corresponding to the biggest eigenvalue represents the main component of the matrix. Therefore, the searched normal vector is the eigenvector attached to the smallest eigenvalue. Let \( v_i \) be the \( i \)th column of \( \mathbf{V} \), the normal vector is then given by the third column of \( \mathbf{V} \):

\[ \vec{n} = v_3 \quad (17) \]

For the known point on the plane, we define it as the average value of all the points in \( S \):

\[ \bar{P} = \frac{1}{N} \sum_{i=1}^{N} P_i \quad (18) \]

Applying the procedure described above, 3D plane estimation is performed for various positions and orientations of the calibration board. Fig. 5 illustrates an example of six 3D planes estimated according to six different sets of reconstructed 3D points. The black frames represent the estimated planes and the colored points within the frames are the reconstructed 3D corner points.

C. Automatic LIDAR measurements extraction

In \([16]\) and \([13]\), the ways of extracting LIDAR points on the calibration board are not mentioned. Here, we apply an automatic extraction approach by differencing the acquired measurements and background data in a static environment. The background LIDAR measurements are acquired before starting calibration. Then, the measurements acquired during the calibration process are compared with the background by a simple substraction operation to get the LIDAR points on the chessboard. According to \([30]\), when a laser ray hits on the edge of the chessboard, the measurement would be the average distance between the chessboard and the background.
Consequently, the LIDAR points in the edge of the chessboard which have unreasonable measurements should be discarded. The final results are shown in Fig. 6.

D. Optimal Estimation of the Rigid Transformation between the LIDAR and the Stereoscopic System

The steps presented above allow us to estimate 3D planes of the calibration board with various poses. In ideal situation, every LIDAR measurement on the calibration board should satisfy the plane equation of the calibration board:

$$\bar{N} \cdot \hat{P}_{\text{stereo}} = \bar{N} \cdot H \cdot \hat{P}_{\text{lidar}} = 0 \quad (19)$$

where $H = \begin{bmatrix} \Phi & \Delta \end{bmatrix}$, $\bar{N} = [\bar{n}_j, d_j]$ is the estimated 3D plane of the chessboard put at the $j$th pose. For the experimental data, all the errors can be represented as:

$$e = \{\bar{N}_1 \hat{P}_{\text{lidar}}^{1,1}, \ldots, \bar{N}_i \hat{P}_{\text{lidar}}^{i,1}, \ldots, \bar{N}_M \hat{P}_{\text{lidar}}^{1,M}, \ldots, \bar{N}_M \hat{P}_{\text{lidar}}^{L,M,M} \} \quad (20)$$

where $\hat{P}_{\text{lidar}}^{k,j}$ is the $k$th LIDAR point on the chessboard at the $j$th pose. In fact, $e$ represents all the possible errors in every measurement.

Although minimizing squared errors in Eq. (20) makes sense in some cases, it doesn’t take into account enough the errors produced either from the 3D plane equations estimated by the stereovision system or from the LIDAR itself. Considering the error of 3D triangulation grows quadratically, this could make results worse when the calibration board is far away from the stereovision system. To make the method more robust against errors, we propose to minimize a squared Mahalanobis distance, rather than Euclidean distance. Compared with Euclidean metric, the Mahalanobis distance benefits can be viewed as follows: firstly, it corrects the scale factor. In our work, the errors come from both stereoscopic system and LIDAR. Euclidean distance is sensitive to the scale of various kinds of variables involved. For Mahalanobis distance, thanks to the measure of covariance, the problem of scale factor is resolved. Secondly, Mahalanobis distance corrects the correlation between variables. This advantage is also a consequence of calculation of covariance. Assuming $e \sim N(0, C)$, then the optimization of extrinsic parameters based on Mahalanobis distance becomes:

$$H(\Phi_i, \Delta_i) = \arg \min_{\Phi_i, \Delta_i} \{e^T C^{-1} e\} \quad (21)$$

where $C$ is the covariance matrix of $e$. Assuming the elements of $e$ in Eq.(20) are mutually independent, then $C$ is a diagonal matrix as:

$$C = \begin{bmatrix} C_{1,1} & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & C_{L,1} & 0 \\ 0 & \cdots & 0 & C_{L,M,M} \end{bmatrix} \quad (22)$$

where $C_{k,j} = \text{var}(\bar{N}_j \hat{P}_{\text{lidar}}^{k,j})$. Meanwhile, assuming the estimation $\bar{N}$ and measurement $P_{\text{lidar}}$ are two independent random vectors with multinormal distribution $N(\bar{N}, \Sigma)$ and $N(P_{\text{lidar}}, \Sigma_{P_{\text{lidar}}})$. A closed-form solution for every element of $C$ is complex. However, if we approximate $C$ by ignoring the middle term $H$, which can be considered as a constant, we obtain:

$$C_{k,j} \approx \text{var}(\bar{N}_j \hat{P}_{\text{lidar}}^{k,j}) = \bar{N}_j^T \Sigma_{P_{\text{lidar}}} \bar{N}_j + \hat{P}_{\text{lidar}}^{k,jT} \Sigma_{P_{\text{lidar}}} \hat{P}_{\text{lidar}}^{k,j} + \text{Tr}(\Sigma_{P_{\text{lidar}}} \Sigma_{P_{\text{lidar}}}^{*}) \quad (23)$$

where $\text{Tr}(\cdot)$ is the trace of a matrix. Hence, the covariance matrix can be approximated with the estimated distribution of $N(\bar{N}, \Sigma)$ and $N(P_{\text{lidar}}, \Sigma_{P_{\text{lidar}}})$. The estimation steps are as follows.

1) Estimating error distribution of 3D planes: The uncertainty of 3D plane estimation $\bar{N}_j$ stems from the corner extraction, matching stage and 3D triangulation stage, and finally propagates into the 3D plane regression. The noise model is simulated by a Monte-Carlo method with the distance between the stereovision system and 3D planes.

We virtually generate 1000 planes $\bar{N}_i(i = 1, ..., 1000)$, which are evenly distributed within 1m-10m from the stereovision system. Moreover, on every virtual plane, 100 corner points are produced and projected into the left and right images as virtual matched corner point pairs $M_i(i = 1, ..., 100)$. To simulate the errors, all the corner point pairs $M_i$ are disturbed by isotropic Gaussian noise with $\delta = 0.5\text{(pixel)}$. Then, the 3D planes $\bar{N}_i$ are computed following the process described above. By comparing ground truth planes $\bar{N}_i$ and estimated planes $\bar{N}_i$, empirical covariances $\Sigma_{\bar{N}_i}, \ldots, \Sigma_{\bar{N}_{1000}}$ are calculated and stored in a database. For regression of the covariance of a new plane, the nearest neighbor regression is adopted for computational efficiency. For a 3D plane $P_{\text{new}}$ not in the database, we estimate its covariance $\Sigma_{P_{\text{new}}}$, which is set to the covariance of the most closest virtual plane in the database. Thus,

$$\Sigma_{P_{\text{new}}} = \Sigma_{\bar{N}_{\text{closest}}} \quad (24)$$

2) Estimating error distribution of LIDAR measurements: LIDAR measurements compose ranges $r_i$ and scan angles $\theta_i$ with additive noises: $r_i = R_i + n_r$, $\theta_i = \Theta_i + n_{\theta}$, where $R_i$ and $\Theta_i$ are the "true" measurements, $n_r$ and $n_{\theta}$ are independent additive zero-mean Gaussian noises with variance $\delta_r$ and $\delta_{\theta}$. Hence, we have,

$$\begin{bmatrix} X_i \\ Z_i \end{bmatrix} = \begin{bmatrix} R_i + n_r \\ \sin(\Theta_i + n_{\theta}) \end{bmatrix} = \bar{P}_{\text{lidar}}^i + n_p \quad (25)$$

where $\bar{P}_{\text{lidar}}^i$ is the "true" result in Cartesian coordinate, $n_p$ is the variance vector. When assuming $n_{\theta} \ll 1$ (which is the fact in most LIDAR equipments), after expanding the $\sin(\cdot)$ and $\cos(\cdot)$, Eq. (25) yields:

$$n_p = (R_i + n_r) n_{\theta} \begin{bmatrix} -\sin \Theta_i \\ \cos \Theta_i \end{bmatrix} + n_r \begin{bmatrix} \cos \Theta_i \\ \sin \Theta_i \end{bmatrix} \quad (26)$$
Hence, the covariance matrix for LIDAR measurements is:

\[
\mathbf{Q}_{i,\text{lidar}} = \frac{(R_i \delta \theta_j)^2}{2} \begin{bmatrix}
2 \sin^2 \theta_i & - \sin 2 \theta_i \\
- \sin 2 \theta_i & 2 \cos^2 \theta_i \\
\end{bmatrix}
\]

(27)

where \( \mathbf{Q}_{i,\text{lidar}} \) is the covariance matrix of the \( i \)th LIDAR measurement \( \mathbf{P}_{i,\text{lidar}} \). \( \delta \theta \) and \( \theta_0 \) are the variances of independent additive zero-mean Gaussian noises \( n_r \) and \( n_\theta \). In practice, \( \theta_i \) and \( R_i \) are approximated by \( \hat{\theta}_i \) and \( r_i \), respectively. Thus, in homogeneous coordinates, the covariance matrix of \( \mathbf{P}_{i,\text{lidar}} \) can be expressed as:

\[
\Sigma_{\mathbf{P}_{i,\text{lidar}}} = \begin{bmatrix}
Q_{i11} & 0 & Q_{i12} \\
0 & 0 & 0 \\
Q_{i21} & 0 & Q_{i22} \\
\end{bmatrix}
\]

(28)

where \( Q_{kj} \) corresponds to the \((k,j)\)th element in Eq. (27). Given the estimated covariance matrices \( \Sigma_{\mathbf{Q}} \) and \( \Sigma_{\mathbf{P}_{i,\text{lidar}}} \), the covariance matrix is achieved by Eq. (23). The classic nonlinear optimization of Eq. (21) is then solved by Levenberg-Marquardt (LM) algorithm [31], since it has a quadratic rate of convergence and has the advantage of preventing the trial step from being too large. This is very practical in real applications. An initial guess of the rigid transformation, \( \mathbf{\Phi}_0 \) and \( \Delta_0 \), is given at first. The solution of extrinsic calibration is represented by \( \mathbf{\Phi}_r, \Delta_r \). In a similar way, the transformation \( \mathbf{\Phi}_r, \Delta_r \) is calculated.

E. Summary of the Calibration Procedure

The proposed calibration procedure is summarized below in the form of Algorithm 1.

**Algorithm 1 3D Plane Reconstruction Based Extrinsic Calibration**

1. for \( j = 1 \) to \( M \) (number of poses) do
2. Reconstruct their 3D position in the stereoscopic coordinate system by optimal triangulation, as presented in section IV-A
3. Estimate the best fitting plane \( \mathbf{N}_j, d_j \), as stated in section IV-B
4. Automatically select the points \( \mathbf{P}_{j,\text{lidar}} := \{P_{i,\text{lidar}} | k = 1, 2, ..., L_j \} \) on the calibration board from LIDAR measurements
5. For every selected LIDAR point and the estimated plane, compute their covariance matrices according to Eq. (28) and Eq. (24), respectively
6. end for
7. Given a first guess, \( \mathbf{\Phi}_0, \Delta_0 \), use LM-algorithm to reach a convergence with \( \mathbf{\Phi}, \Delta \) minimizing the objective function of Eq. (21)
8. return \( \mathbf{\Phi}, \Delta \)

V. Experimental Results

The proposed algorithm has been implemented with Matlab and tested using both computer simulations and real data sets. Moreover, a comparison between our approach and a popular camera/LIDAR calibration method [16] is given.

A. Computer Simulations

Ground truth of the 3D rigid transformation \( (\mathbf{\Phi}_r, \Delta_r) \) is set by rotation vector \( [0.5^\circ, 2^\circ, -3^\circ]^T \), and the translation vector \( \Delta_l = [0.5, 1.2, -0.3]^T \) (in meters). The intrinsic parameters of the two cameras are the same: \( f_x = 1040, f_y = 1050 \) (focal lengths in pixels), principal point is \( (600, 450) \) (in pixels) and skewness coefficient \( \gamma = 0 \). The extrinsic parameters within the stereoscopic system are set by the rotation vector \( [0.1^\circ, -0.1^\circ, 0.2^\circ]^T \), and the translation vector \( \mathbf{T}_r = [0.28, 0.04, -0.02]^T \) (in meters). We set several virtual 3D planes and select some points from each plane, project them into the images of the left and right cameras. The 3D points whose projections are within the left and right images are collected. Virtual LIDAR measurements on calibration board planes are acquired at the intersection of the chessboard plane and the scan plane of the LIDAR.

1) Performance w.r.t the image noise: Gaussian noise with zero mean and standard deviation \( \sigma \) (from 0.5 to 6 pixels) is added to the virtual corner points. Based on the "noised" corner points, the proposed method is compared with Zhang’s method [16]. The errors are computed as:

\[
\varepsilon_{\Phi} = \| \mathbf{\Phi}_r - \mathbf{\Phi}_0 \|, \quad \varepsilon_{\Delta} = \| \Delta_r - \Delta_0 \|
\]

(29)

where \( \mathbf{\Phi}_r \) and \( \Delta_r \) are the ground truth values, \( \mathbf{\Phi}_0 \) and \( \Delta_0 \) are the estimated values. For short, we only measure the error of the left camera. The results are shown in Fig. 7 (a), (b). It’s clear to see that our calibration method performs the best and tolerance of image noises is around 3 pixels.

2) Performance w.r.t the distance: An experiment is conducted to explore an available range for the position of the calibration board. We set 6 different calibration board poses within a range of 1m. This is repeated at different distances within the range \( \left[ 2n, 12n \right] \) from the stereoscopic system. The calibration results are shown in Fig. 8 (a), (b), which reveal that the best range is within 8 meters.

3) Performance w.r.t LIDAR noise: The LIDAR points are "polluted" by zero-mean Gaussian noises of range and angle measurements \( n_r \) and \( n_\theta \) with variances \( \delta_r \) and \( \delta_\theta \), respectively. \( \delta_0 = 0.0001 \) degree while \( \delta_r \) varies from 0 to 50 (mm). The simulation results are shown in Fig. 8 (c), (d). One can see that our method makes the calibration results more robust, when compared with Zhang’s method [16].

4) Performance w.r.t the number of calibration board poses: We randomly generate 22 different 3D calibration board planes 10 times. By gradually increasing the number of various poses of the chessboard from 2 to 22 in each run, we record the results and calculate the errors. We also test Zhang’s method [16] for comparison. We run the tests 10 times with various sets of chessboard poses. The results are averaged, and shown in Fig. 8 (e), (f). The error bars represent the variance of 10 different tests. Good results can be obtained when the number of poses is larger than 6.

5) Performance w.r.t the first guess in LM algorithm: Although Levenberg-Marquardt (LM) algorithm is very robust, it is required to evaluate the influence of the first guess on the final results. In this experiment, since it is impossible to exhaustively try all the random first guesses, we test all
the factors in a first guess independently. For the initial rotation guesses, we start from a $3 \times 3$ identity matrix, then independently change the yaw, pitch and roll angles $[32]$ from $0^\circ$ to $90^\circ$ with an interval of $5^\circ$ as:

$$R_{\text{guess}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot R_z(\gamma) \cdot R_y(\alpha) \cdot R_z(\beta)$$

(30)

where $\gamma, \alpha, \beta$ are the yaw, pitch and roll angles. For the initial translation guesses, we start from $[0, 0, 0]^T$ (in meter), then gradually increase the $X, Y, Z$ values independently to $[10, 10, 10]^T$ (in meter) with an interval of 0.5 meter. Results of errors according to various first guess values are shown in Fig. 9. It is shown that LM algorithm is quite robust. Indeed, even for a first guess that is far from the expected value, it always produces solutions close to the ground truth.

**B. Real Data Test**

In real experiments, the calibration pattern is $19 \times 12$ squares of $50\text{mm} \times 50\text{mm}$. The chessboard is detected by the two sensors meanwhile with 20 various poses. We both apply our method and Zhang’s approach [16] for comparison. The projection of the LIDAR measurements into the images of the stereoscopic system is shown in Fig. 10 (a) - (d). It seems that both of the results are quite reasonable. While from the partial enlargement views, our method gets more stable and precise results. Indeed, blue points (calculated by our method) in the pair of images have almost the same height, while the red ones (calculated by Zhang’s method [16]) have obvious deviation between the two images. The estimated extrinsic parameters using our method and Zhang’s method are shown in Tab. I. It shows that our algorithm performs more precisely
Fig. 10: Comparison of the proposed method with Zhang’s method \[16\]. In (c) and (d), the blue points (by the proposed method) are almost at the same height, while the red points (by Zhang’s method \[16\]) are not.

Table I: Extrinsic Parameters and Precision Measure Calculated in Real Data

<table>
<thead>
<tr>
<th></th>
<th>The proposed method</th>
<th>Zhang’s method [16]</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \Phi_1 )</td>
<td>([0.1474, -0.0251, 0.0083])</td>
<td>([0.1520, -0.0284, 0.0045])</td>
</tr>
<tr>
<td>( \Delta_1 )</td>
<td>([0.1195, -1.4543, -1.2326])</td>
<td>([0.1245, -1.4566, -1.2297])</td>
</tr>
<tr>
<td>( \Phi_r )</td>
<td>([0.1509, -0.0210, 0.0067])</td>
<td>([0.1409, -0.0240, -0.0101])</td>
</tr>
<tr>
<td>( \Delta_r )</td>
<td>([0.3516, -1.4533, -1.2398])</td>
<td>([0.3637, -1.4583, -1.2427])</td>
</tr>
<tr>
<td>( \varepsilon )</td>
<td>0.5405 mm</td>
<td>3.6 mm</td>
</tr>
</tbody>
</table>

More experimental results in outdoor environments are shown in Fig. 11. Here, we also apply the same algorithm for the 2D LIDAR mounted on the top of our platform. Fig. 11 (b),(c) show LIDAR measurements from the bottom and top. Fig. 11 (a) shows the projections of LIDAR points into the left camera’s field of view by the output of the proposed algorithm. It has to be noticed that not all the LIDAR points are within the camera’s field of view. The LIDAR points within the camera’s field of view are marked by red color in Fig. 11 (b),(c). Fig. 11 (d) (e) and (f) show another test in outdoor environment. We can see that the projections are quite reasonable, since the geometric characteristics of the scenes detected by the camera coincide closely with LIDAR data.

VI. Conclusion and Future Work

In this paper, we presented a novel extrinsic calibration method for integrating a LIDAR and a stereoscopic system composed with two cameras. Since the geometric relationship between the two cameras in the stereoscopic system is considered, real data experiments show that our method is more exact and precise than Zhang’s method which is widely used. Meanwhile, the introduction of sensor noise models and non-linear Mahalanobis distance optimization makes our method much more robust than Zhang’s method. This is verified by computer simulation results. The proposed method can be applied for any multi-sensor fusion system consisting of multiple cameras and multiple LIDARs. Future work will be mainly focused on the calibration of a stereo vision system with multi-layer LIDARs. Also, we plan to develop an improved self-calibration method.

REFERENCES

Fig. 11: More experimental results in outdoor environments


