An Adaptive Camera and Lidar Joint Calibration Algorithm

Wen-tao YAO\textsuperscript{a,*}, Wen-sheng DONG\textsuperscript{b}, Chun-feng SHEN\textsuperscript{c}

Shanghai Baosight Software Co. Ltd., Shanghai 201203, China
\textsuperscript{a}yaowentao@baosight.com; \textsuperscript{b}dongwensheng@baosight.com; \textsuperscript{c}shenchunfeng@baosight.com

*Corresponding author

Keywords: Camera, Lidar, Joint calibration, Information fusion, Large mechanism intelligent control system

Abstract. In large mechanism intelligent control system, cameras and lidars are usually mounted to obtain environmental information. In order to fuse the information of camera and lidar, joint calibration is essential to place their data under the same coordinate system. To solve the problem of camera and 3D lidar joint calibration, an adaptive joint calibration method is proposed. Only one planar calibration board is needed for the calibration. In the nonlinear optimization procedure, the camera extrinsic parameters and optimized together with the rotation matrix and translation vector from lidar to camera coordinate. Compared with the method of fixed camera parameters, average calibration error decreases by 22.50%. By using weighed lidar data, calibration error further decreases by 8.80% in average.

Introduction

In industry fields, camera and 3D lidar are often used to obtain environmental information. To fuse the data of them, we need first put their data under one uniform coordinate system. Therefore, we should first jointly calibrate the camera and the lidar.

There already exist some methods for joint calibration of camera and lidar [1]. According to the different types of lidar, the calibration methods can usually divided into three categories: methods based on visible lidar [2][3][4][5]; methods based on planar lidar [6][7][8][9][10]; methods based on multi-line and 3D lidar [11][12][13].

In this paper, we propose an adaptive camera and lidar joint calibration algorithm. Our algorithm is easy to be carried out. Only one planar chessboard is needed and only few parameters are required to be measured for the calibration process. Our algorithm is also with high accuracy and is very useful in various fields.

Coordinate and Geometric Constraints

The joint calibration between camera and lidar is to obtain their relative position, including translation and rotation. The relative position from 3D lidar to camera is usually described using a 3*3 rotation matrix $\mathbf{R}^f$ and a 3*1 translation vector $\mathbf{t}^f$. The purpose of joint calibration is to solve $\mathbf{R}^f$ and $\mathbf{t}^f$.

The definition of coordinate system is shown in Figure 1. The origin of lidar coordinate is the center point of lidar, and $X_l, Y_l, Z_l$ are its axes. The origin of camera coordinate is the camera’s optical center, and $X_c, Y_c, Z_c$ are its axes. The origin of chessboard object coordinate is the top-left corner of the chessboard, and $X, Y, Z$ are its axes. For image coordinate, we define the top-left point of the image as its origin, and
rightward and downward are $u$ and $v$ axes, respectively. This means $u$ and $v$ represent the column and row of pixel $(u,v)$ respectively.

Figure 1. Coordinate system and planar constraint.

We use $[X_l, Y_l, Z_l]^T$ to represent a point in 3D lidar coordinate. $[X_c, Y_c, Z_c]^T$ and $[u, v]^T$ is the corresponding point in camera coordinate and image coordinate, respectively. $R_i$ and $t_i$ are rotation matrix and translation vector from 3D lidar to camera coordinate. $R_i$ and $t_i$ are the rotation matrix and translation vector from chessboard object coordinate to camera coordinate for the $i$-th frame, which are called camera extrinsic parameters.

For a point in space, we assume its coordinate to be $P_i = [X_i, Y_i, Z_i]^T$ in 3D lidar coordinate system, and $P_c = [X_c, Y_c, Z_c]^T$ in camera coordinate system. We have:

$$P_c = R_i P_i + t_i$$  \hspace{1cm} \text{(1)}

The relationship between point in camera coordinate and homogenous image coordinate is as Eq.(2):

$$sp = AP_c$$  \hspace{1cm} \text{(2)}

Where $A$ is intrinsic parameter of the camera, $s$ is scale factor.

Data is needed to be collected before calibration. The calibration algorithm in this paper requires a series of time-synchronized camera image and corresponding 3D lidar data. In this paper, camera image is grayscale whose resolution is 780*572, as shown in Figure 2. Each 3D lidar frame is a set of 3D lidar points $[X_i, Y_i, Z_i]$ for $i = 1, ..., N$. In a typical scene, a frame of 3D lidar contains about 13,000 lidar points. Figure 3 is the same scene captured by 3D lidar in which lidar points are visualized. Calibration can be done after data is collected.

Figure 2. Camera Image.
First of all, the camera is required to be calibrated. Any camera calibration algorithm can be used as long as extrinsic parameters \((\mathbf{R}_i, \mathbf{t}_i)\) for each frame can be obtained. After the camera is calibrated, we can start the joint calibration for camera and 3D lidar. The joint calibration algorithm is two-step: First, we compute closed form of \(\mathbf{R}_i\) and \(\mathbf{t}_i\) using constraints of lidar point on chessboard; Second, we use the closed form solution as initial value, and the projection error of lidar points in image as optimization object to do non-linear optimization on \(\mathbf{R}_i\) and \(\mathbf{t}_i\).

In each frame of data, the chessboard can both be observed by camera and 3D lidar. The chessboard is also called the chessboard plane. Without loss of generality, we assume \(\mathbf{Z} = \mathbf{0}\) for the chessboard plane in chessboard object coordinate, and the rotation matrix and translation vector from chessboard object coordinate to camera coordinate to be \(\mathbf{R}_i\) and \(\mathbf{t}_i\). Assume \(\mathbf{r}_{ij}\) to be the \(j\)-th column of \(\mathbf{R}_i\), \(\mathbf{r}_{ij}\) is the normal vector for chessboard plane in camera coordinate. Note that the origin of chessboard object coordinate is the top-left corner point of the chessboard, and the origin of camera coordinate is the camera’s optical center. Therefore, translation vector \(\mathbf{t}_i\) means the coordinate of the top-left corner point of the chessboard under camera coordinate system. \(\mathbf{P}_c\) is the camera coordinate of lidar points on chessboard plane. Thus, in camera coordinate, \(\mathbf{t}_i\) and \(\mathbf{P}_c\) are both located on the chessboard plane. Vector \(\mathbf{v} = \mathbf{P}_c - \mathbf{t}_i\) is also located on the chessboard plane. Since \(\mathbf{r}_{ij}\) is the normal vector of the chessboard plane, we have:

\[
\mathbf{r}_{ij} \cdot \mathbf{v} = \mathbf{r}_{ij} \cdot (\mathbf{P}_c - \mathbf{t}_i) = 0
\]

The constraints above can be seen in Figure 1.

We substitute Eq. (3) with Eq. (1):

\[
\mathbf{r}_{ij}^T (\mathbf{R}_i^T \mathbf{P}_i + \mathbf{t}_i^T - \mathbf{t}_i) = 0
\]

Write \(\mathbf{P}_i = [X_i, Y_i, Z_i]^T\) as the homogenous coordinate, we have:

\[
\mathbf{r}_{ij}^T \mathbf{P}_i = \mathbf{r}_{ij}^T \mathbf{t}_i.
\]

For each lidar point located on the chessboard plane, Eq. (5) describes a geometric constraint on \((\mathbf{R}_i, \mathbf{t}_i)\). This is a basic geometric constraint between 3D lidar coordinate and camera coordinate.
Closed-form Solution

We use RANSAC algorithm [15] to extract lidar points on the chessboard for each frame, as the red dot shown in Figure 4. Then we can compute the closed-form solution for \( R_i \) and \( t_i \).

![Figure 4. Lidar points on calibration board (red).](image)

We assume \( R_i \) and \( t_i \) to be camera extrinsic parameter for the \( i \)-th frame. We define \( r_{i,j} = [r_{i,1}, r_{i,2}, r_{i,3}]^T \), \( t_i = [t_{i,1}, t_{i,2}, t_{i,3}]^T \), \( R_{ij} \) to be the \( i \)-th row and \( j \)-th column of \( R_i \). \( t_{ij} \) to be the \( i \)-th row of \( t_i \). Assuming there are \( n_i \) lidar points on chessboard for the \( i \)-th frame of 3D lidar data. We use \( P_{i,in} = [X_{i,in}, Y_{i,in}, Z_{i,in}]^T \) for each 3D lidar point. Therefore, for lidar points \( P_{ij} \) on chessboard, we rewrite Eq. (5) as:

\[
\begin{bmatrix} r_{i,11} & r_{i,12} & r_{i,13} & t_{i,1} \\ r_{i,21} & r_{i,22} & r_{i,23} & t_{i,2} \\ r_{i,31} & r_{i,32} & r_{i,33} & t_{i,3} \end{bmatrix} \begin{bmatrix} X_{ij} \\ Y_{ij} \\ Z_{ij} \end{bmatrix} = \begin{bmatrix} r_{i,31} \cdot r_{i,32} \cdot r_{i,33} \\ t_{i,2} \\ t_{i,3} \end{bmatrix} \quad (6)
\]

For all \( n_i \) lidar points on chessboard in one frame, we consider elements in \( R_i \) and \( t_i \) as unknown variables, Eq.(6) can further be rewrite as:

\[
A_i x = b_i \quad (7)
\]

For each frame, we stack Eq.(7). Assuming there are \( N \) frames, we can obtain a linear equation system:

\[
AX = b \quad (8)
\]

Where

\[
A = \begin{bmatrix} w_1 A_1 \\ w_2 A_2 \\ \vdots \\ w_N A_N \end{bmatrix}_{N \times 12}
\]

is a \( \sum_{i=1}^{N} n_i \times 12 \) matrix, and \( w_i \) is the weight for the \( i \)-th frame.
is a $\sum_{i=1}^{N} n_i \times 1$ vector.

After matrix $A$ and $b$ are obtained, the least square solution for $X$ can be directly computed. $X$ is a 12*1 vector.

$$X = (A^T A)^{-1} A^T b$$  \hspace{1cm} (1)

We assume $X_i$ to be the $i$-th element for $X$, the closed-form least square solution for $R_i^f$ and $t_i^f$ is:

$$R_i^f = \begin{bmatrix} X_1 & X_2 & X_3 \\ X_4 & X_5 & X_6 \\ X_7 & X_8 & X_9 \end{bmatrix}, \quad t_i^f = \begin{bmatrix} X_{10} \\ X_{11} \\ X_{12} \end{bmatrix}$$  \hspace{1cm} (12)

Because of noise and measurement error, the $R_i^f$ is not necessarily be an orthogonal matrix. To make $R_i^f$ orthogonal, we do SVD decomposition for $R_i^f$, which makes $R_i^f$ to be $R_i^f = UDV^T$, where $U$ and $V$ are orthogonal matrices, and $D$ is a diagonal matrix. If $R_i^f$ is an orthogonal matrix, we have $D=I$. Thus, we consider $R_i^f = UDV^T$ to be the best linear estimation for $R_i^f$.

**Non-linear Optimization**

In last section, we get the closed-form solution for $R_i^f$ and $t_i^f$. But this least square solution is not of physical meaning. Therefore, we will do non-linear optimization for $R_i^f$ and $t_i^f$, which is more meaningful and more accuracy. During the camera calibration procedure, the sum of projection error for chessboard corners in images is often used as the optimization function for camera intrinsic and extrinsic parameters. In this section, we use similar method to do non-linear optimization for $R_i^f$ and $t_i^f$.

Eq.(4) can be rewrite as follows:

$$r_i^T (R_i^f P_i + t_i^f) = r_i^T t_i.$$

(13)

To observe both sides of Eq.(13), it is easy to find that $R_i^f P_i + t_i^f$ and $t_i$ are both the camera coordinate for points on chessboard. Thus, both sides are distance from chessboard to origin of camera coordinate. If the left side is not equal to the right side, the difference is the projection error for the lidar points in camera coordinate. Since the number of lidar points in each frame differs from each other, we choose the object function as the sum of average projection error of lidar points in camera coordinate in each frame. We use this object function to do non-linear optimization for $R_i^f$ and $t_i^f$.

The object function is as Eq.(14):

$$\arg\min_{R_i^f, t_i^f} \sum_{i=1}^{N} \frac{1}{n_i} \sum_{j=1}^{n_i} (r_{i,j}^T (R_i^f P_{ij} + t_i^f) - t_{i,j})^2$$

(14)

Where argmin means that we want to find the parameters that can minimize the object function. By using Rodrigues transform[16], the closed-form solution for $R_i^f$ can be
expressed as vector \( \mathbf{r}_f \). The closed-form solution for translation vector can be expressed as \( \mathbf{r}_f = [x_f, y_f, z_f] \). Thus, 6 variables need to be optimized. We use the closed-form solution as initial value, together with LM optimization algorithm, to do the non-linear optimization.

During the non-linear optimization, after we get the non-linear estimation for \( R_f \) and \( t_f \), we can further optimize the camera extrinsic parameters. In camera extrinsic parameters optimization stage, the object function for the \( i \)-th calibration image is as follows:

\[
\arg\min_{R_i, t_i} \sum_{j=1}^{m_i} \| p_{ij} - \hat{m}(A, R_i, t_i, P_j) \|^2 + \alpha \sum_{j=1}^{m_i} \left( t_i^T (R_i^T p_{ij} + t_i) - t_j \right)^2
\]

where \( \alpha \) is a constant to control the error ratio for two parts.

In non-linear optimization, we alternatively optimize \( R_i \) and \( t_i \), together with camera extrinsic parameter \( R_i \) for each frame.

### Experimental Results

Table 1 shows average projection error of lidar points on chessboard in different stages of joint calibration. We alternatively optimize for \( R_i, t_i \) and \( R_f \) for 20 times. The unit of error is cm.

Table 1. Average projection error of lidar points on chessboard (w/o weights).

<table>
<thead>
<tr>
<th>Dataset</th>
<th>#Frames</th>
<th>Closed-form</th>
<th>Optimize ( R_i ) and ( t_i )</th>
<th>Optimize ( R_f, t_i ) and ( R_f )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dataset 1</td>
<td>26</td>
<td>2.052</td>
<td>2.031</td>
<td>1.820</td>
</tr>
<tr>
<td>Dataset 2</td>
<td>24</td>
<td>2.292</td>
<td>2.390</td>
<td>1.955</td>
</tr>
<tr>
<td>Dataset 3</td>
<td>28</td>
<td>3.576</td>
<td>3.470</td>
<td>2.736</td>
</tr>
<tr>
<td>Dataset 4</td>
<td>48</td>
<td>3.967</td>
<td>4.408</td>
<td>3.150</td>
</tr>
<tr>
<td>Dataset 5</td>
<td>63</td>
<td>4.498</td>
<td>4.181</td>
<td>3.650</td>
</tr>
<tr>
<td>Dataset 6</td>
<td>68</td>
<td>9.840</td>
<td>11.183</td>
<td>9.192</td>
</tr>
</tbody>
</table>

It is easy to be observed that after the non-linear optimization for \( R_i \) and \( t_i \), the average error always decreases. After alternative optimization for \( R_i, t_i \) and \( R_f \), the average error further decreases by 22.50%, thus proves the effectiveness of our method. In addition, projection error for dataset 6 is obviously larger than other five datasets. That is because no lens distortion rectification is done for dataset 6.

Since the distance from the chessboard to lidar is different in each frame, the lidar points on the chessboard also differs. There are more lidar points on the chessboard when the chessboard is closer to lidar. To utilize the distance information, we define the weight for each frame proportional to the number of lidar points on the chessboard. That means \( w_{ij} \) is proportional to \( n_r \). Table 2 shows calibration result with weights. The data is with the same meaning in Table 1. The optimization algorithm and number of iterations are also same with Table 1. The unit of error is cm.

Table 2. Average projection error of lidar points on chessboard (w/ weights).

<table>
<thead>
<tr>
<th>Dataset</th>
<th>#Frames</th>
<th>Closed-form</th>
<th>Optimize ( R_i ) and ( t_i )</th>
<th>Optimize ( R_f, t_i ) and ( R_f )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dataset 1</td>
<td>26</td>
<td>2.052</td>
<td>2.032</td>
<td>1.819</td>
</tr>
<tr>
<td>Dataset 2</td>
<td>24</td>
<td>2.292</td>
<td>2.211</td>
<td>1.946</td>
</tr>
<tr>
<td>Dataset 3</td>
<td>28</td>
<td>3.576</td>
<td>3.377</td>
<td>2.564</td>
</tr>
<tr>
<td>Dataset 4</td>
<td>48</td>
<td>3.967</td>
<td>3.841</td>
<td>2.913</td>
</tr>
<tr>
<td>Dataset 5</td>
<td>63</td>
<td>4.498</td>
<td>4.151</td>
<td>3.618</td>
</tr>
<tr>
<td>Dataset 6</td>
<td>68</td>
<td>9.840</td>
<td>9.709</td>
<td>7.663</td>
</tr>
</tbody>
</table>
To compare the 5-th column for Table 1 and Table 2, we can find that the projection error is obviously decreased when the number of lidar points on the chessboard is used as weight for each frame. The average error decreases by 8.80% with weights, which proves the effectiveness of weighing each frame.

Figure 5 shows the 3D reconstruction of joint calibration result. It shows the camera coordinate (blue), the lidar coordinate (green) and their relative position with chessboard in each frame.

![Figure 5. 3D reconstruction of joint calibration result.](image)

**Conclusions**

In this paper, we propose an adaptive camera and lidar joint calibration algorithm. By using our algorithm, the relationship between camera coordinate and lidar coordinate can be established. Our method has many advantages compared to the existing methods. Firstly, the calibration object is very simple and easy to make. Only one plane with chessboard pattern is required. Secondly, only few parameters need to be measured. We need only measure the width and height for the chessboard square and no other parameter is required to be measured.

The joint calibration algorithm in this paper has a high accuracy. By alternatively optimizing the rotation matrix and translation vector, together with the camera extrinsic parameters, the average error decreases by 22.50%. By weighing each frame, the calibration error further decreases. Compared to the result without weighing, the average error decreases by 8.80%.

**Acknowledgement**

This work is supported by Shanghai Science and Technology Committee project 14JC1402203.

**References**


