25th International Conference on Pattern Recognition (ICPR2020)

A two-step approach to Lidar-Camera calibration

Yingna Su, Yaqing Ding, Jian Yang and Hui Kong
School of Computer Science and Engineering, Nanjing University of Science and Technology

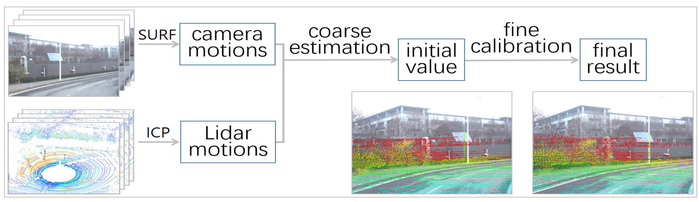
Abstract

Autonomous vehicles and robots are typically equipped with Lidar and camera. Hence, calibrating the Lidar-camera system is of extreme importance for ego-motion estimation and scene understanding. In this paper, we propose a two-step approach (coarse + fine) for the external calibration between a camera and a multiple-line Lidar. First, a new closed-form solution is proposed to obtain the initial calibration parameters. With the initial calibration parameters, the ICP-based calibration framework is used to register the point clouds which extracted from the camera and Lidar coordinate frames, respectively. Experimental results demonstrate that our method achieves promising performance and higher accuracy than two open-source methods.

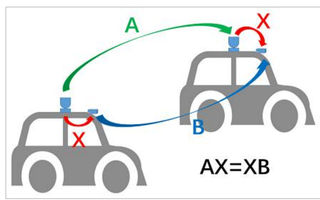
Contributions

- We propose a novel closed-form algorithm that uses at least two motions of camera and Lidar to estimate the initial Lidar-camera transformation matrix.
- With the initialization, the ICP algorithm is used to register two point cloud in the camera coordinate system and the Lidar coordinate system, respectively, to refine the coarse estimation.
- Our method has been evaluated on both synthetic and real data with detailed analyses. Experimental results show that our method gives very promising performance.

Approach



Overview of the proposed Lidar-camera calibration method which consist of two stages: coarse estimation and fine calibration.



Lidar-camera calibration can be formulated as the "hand-eye" calibration problem.

1. Coarse estimation

The rigid-mounted camera and Lidar should satisfy the following constraints:

$$R_c^i R = R R_L^i \quad (1)$$
$$R_c^i T + k_i T_c^i = R T_L^i + T. \quad (2)$$

where $\{R_c^i, T_c^i\}$ and $\{R_L^i, T_L^i\}$ are the camera and Lidar motion. $\{R, T\}$ is the rigid transformation.

Eq. (1) is equal to

$$n_c^i = R n_L^i, \quad (3)$$

where n_c^i, n_L^i are the rotation axes of the rotation matrices R_c^i, R_L^i.

We define the following cost function:

$$\Delta = \sum_{i=1}^{k} ||n_c^i - R n_L^i||^2 = \sum_{i=1}^{k} (n_c^i - R n_L^i)^\top (n_c^i - R n_L^i) = \sum_{i=1}^{k} (n_c^{i\top} n_c^i + n_L^{i\top} n_L^i - 2 n_c^{i\top} R n_L^i) \quad (4)$$

We use the unit quaternion $q = [w, x, y, z]$ to parameterize the rotation matrix R:

$$\begin{bmatrix} w^2+x^2-y^2-z^2 & 2(xy-wz) & 2(xz+wy) \\ 2(xy+wz) & w^2+y^2-x^2-z^2 & 2(yz-wx) \\ 2(xz-wy) & 2(yz+wx) & w^2+z^2-x^2-y^2 \end{bmatrix} \quad (5)$$

with the unit quaternion constraint: $w^2+x^2+y^2+z^2-1=0$.

Then we can use the Lagrange multiplier and define the following function:

$$F = \Delta + \lambda (w^2+x^2+y^2+z^2-1). \quad (6)$$

To find the minimum of F, the partial derivatives of F should be zero.

$$F'_{w,x,y,z} = 0. \quad (7)$$

Then we obtain four polynomial equations:

$$\begin{aligned} \boldsymbol{a_i} \cdot [w\ x\ y\ z] + \lambda w &= 0, \\ \boldsymbol{b_i} \cdot [w\ x\ y\ z] + \lambda x &= 0, \\ \boldsymbol{c_i} \cdot [w\ x\ y\ z] + \lambda y &= 0, \\ \boldsymbol{d_i} \cdot [w\ x\ y\ z] + \lambda z &= 0, \end{aligned} \quad (8)$$

where $\boldsymbol{a_i}, \boldsymbol{b_i}, \boldsymbol{c_i}, \boldsymbol{d_i}$ are known coefficients calculated from the camera and lidar motions.

We use the hidden variable technique to solve Eq. (8) and obtain the rotation matrix. Once the rotation matrix is known, the translation matrix can be calculated from Eq. (2).

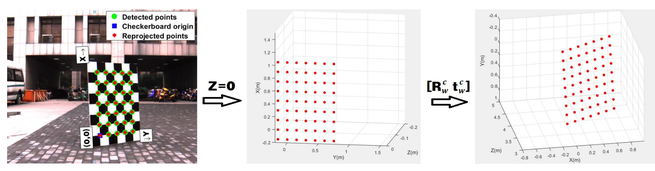
Algorithm 1 Coarse estimation

Require: camera motions $\{R_c^i, T_c^i\}$ and Lidar motions $\{R_L^i, T_L^i\}$
Ensure: the transformation matrix $\{R, T\}$ between the camera and Lidar
1: use the unit quaternion $q = [w, x, y, z]$ to parameterize R
2: convert R_c^i, R_L^i into the unit rotation axes n_c^i, n_L^i
Steps 3-4 are performed within LO-RANSAC
3: compute the matrix C in Eq. 8
4: find the null space of the matrix C, and build the rotation matrix. Once the loop is finished, we obtain the best rotation matrix R.
5: calculate T based on Eq. 2 and R

2. Fine calibration

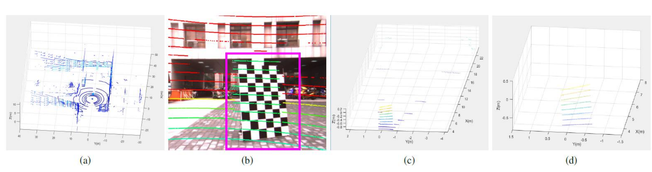
With the initial calibration parameters, the second step is to utilize ICP algorithm to register two point clouds in the camera and Lidar coordinate system, respectively, to refine the coarse estimation.

- The generation of the point cloud in the camera coordinate system:



The point cloud is generated based on Zhang's camera calibration algorithm. We first assemble all the corner points to generate a point cloud. Then we transform the point cloud from the world coordinate system to the camera coordinate system with the extrinsic parameters obtained from zhang's method.

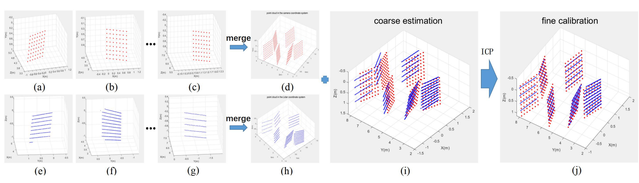
- The generation of the point cloud in the Lidar coordinate system:



Correspondingly, the point cloud in the Lidar coordinate system can be extracted from raw Lidar data. As the figure shows, we filter the raw data (a), keeping only the points that are re-projected onto the checkerboard and the surrounding area (shown within the purple box in (b)). Then a plane fitting method is used to obtain the final point cloud (d) from the filtered point cloud (c).

In this way, we collect 6 pairs of point clouds of the checkerboard pattern at different positions and in different orientations in the camera and Lidar coordinate systems, respectively.

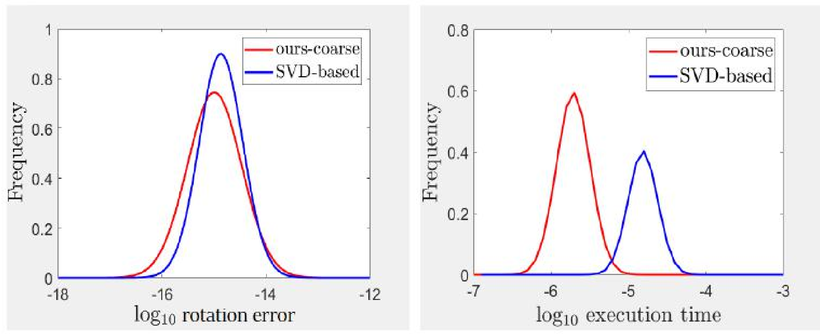
- Use the ICP algorithm to refine the calibration result:



Finally, we use the ICP algorithm with the initialization obtained from coarse estimation step, to register the point clouds and then get the fine calibration result.
(a)(b)(c) and (e)(f)(g) are the point cloud of the checkerboard pattern with different positions in the camera and the Lidar coordinate system, respectively. (d) and (h) are the point cloud originated from the fusion of (a)(b)(c) and (e)(f)(g), respectively. With the coarse calibration result (i), we use ICP algorithm to register the two merged point cloud (d) and (h) to get the fine calibration result (j).

Experimental Results

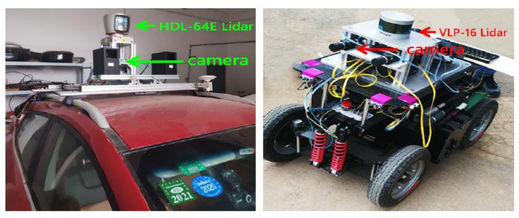
- **Coarse calibration**



(a) rotation error (b) execution time

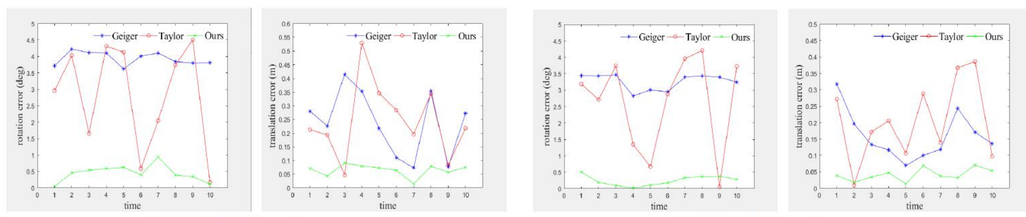
we compare our closed-form solution against the state-of-the-art SVD-based algorithm with synthetic data. The figures show kernel smoothed histograms of log10 rotation error and execution time, respectively. As we can see, our method can effectively reduce the run time while possessing similar accuracy as the SVD-based algorithm.

- **Fine calibration**



(a) (b)

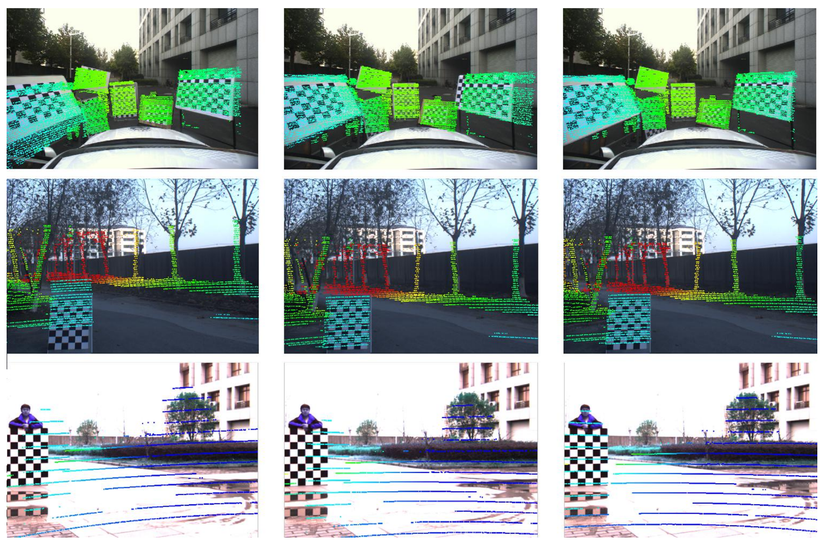
Experimental Setup:
(a) An unmanned car equipped with the HDL-64E Lidar and the PointGrey Flea3 camera.
(b) (b) A mobile robot equipped with the VLP-16 Lidar and the PointGrey Flea3 camera.



(a) HDL-64E Lidar and camera (b) VPL-16 Lidar and camera

We compare our method with two state of the art algorithms denoted as 'Geiger' and 'Taylor'. The comparison results of the rotation error (the 1st and the 3rd subfigure) and the translation error (the 2nd and the 4th subfigure) by our method and the other two methods are shown in the figure. Our method is more robust than comparable methods.

- **qualitative calibration results**



The qualitative calibration result is displayed to present the comparison intuitively. The calibration results from the left to right column are from Geiger, Taylor and our method, respectively. The first and second row show the result captured by the HDL-64E Lidar. The third row shows the VPL-16 Lidar results. As shown, our calibration method achieves the best results.

Paper ID: 1895

Contact email: suyingna@njust.edu.cn