Calibration of mobile robot odometry and camera based on hand-eye calibration

Xuwu Geng\textsuperscript{1,2} and Yueli Hu\textsuperscript{1,2,3}

\textsuperscript{1} Shanghai Key Laboratory of Power Station Automation Technology, Shanghai 200444, China
\textsuperscript{2} School of Mechatronic Engineering and Automation, Shanghai University, Shanghai 200444, China
\textsuperscript{3} E-mail: huyueli@shu.edu.cn

Abstract. The relative pose transformation among several sensors in mobile robots has a certain impact on the pose estimation of the robot. The traditional hand-eye calibration works out the matrix transformation relationship between the camera coordinate system and the robot end effector or its base coordinate system. As for the household sweeping robot, which use monocular camera for localization and mapping. A novel method is proposed calibrating the chassis odometry and the top-view camera. First, we get chassis poses from wheel encoder and camera poses from Perspective-n-Point(PnP). Then we use two types of poses to solve the formula we build which is similar to hand-eye calibration. We make some tests to ensure this method is feasible.

1. Introduction

Simultaneous localization and mapping (SLAM) is widely used in mobile robots for their localization with different sensors, such as lasers and cameras. Compared with laser SLAM, monocular SLAM has a lower cost and rich information. And monocular SLAM has the function of relocalization which pure odometry localization does not own [1, 2]. Therefore, some household sweeping robots begin to use monocular cameras for localization and mapping. The camera is installed on top of the chassis, making the camera realizes its location and builds a point-cloud map by capturing the pictures of ceiling. When the camera is installed, there is a relative pose transformation matrix between the camera and the odometry. When it needs to use the camera pose calculated by the visual SLAM as the true pose of the sweeping robot in the moving process, it has to find the pose transformation matrix between the odometry and the camera.

In order to get the accurate calibration, the poses of chassis and camera should be as accurate as possible. The poses of chassis obtained from wheel encoder while poses of camera can obtain from many ways. Some researchers have done some work. In [3], G.Antonelli, etc. use a box landmark with 135 makers to certain the poses of camera. In [4], Shusheng B, etc. use multi-composite-targets around the robot to solve the problem which needs a big ground and a lot of mark boards.

This paper proposes a novel algorithm uses only one board with square makers and use PnP solver [5] to obtain the poses of camera. The board is fixed on the ceiling. The sweeper obtains the odometry and camera data during the period of movement and rotation. We calculate the pose transforms relative to the first frame for solving the formula we get in section 2. Finally, we take some tests to ensure the novel method is feasible.
2. Model construction

The chassis of a sweeping robot uses a spider gear wheel structure. The odometry is installed at the center of the circular chassis. After our reference in [6] A mobile model of the robot is built (see Figure 1). The initial odometry position is settled as \( P_o \) and the initial camera position as \( P_c \). If the position is presented by \( SE(3) \), \( T^w_o \) indicates the position of the camera relative to the world coordinate system. \( T^w_r \) indicates the position of the robot(odometry) relative to the world coordinate system. The world coordinate system coincides with the first frame of the camera coordinate system.

After a period of movement, the odometry poses and camera's poses are \( T^w_{ik} \) and \( T^w_{ck} \) in the world coordinate system at the \( k \)th frame of the camera. The pose at the \( k \)th frame compared with the first frame expresses as:

\[
T^w_{ik} = T^w_{ic} \cdot T^w_{ic}
\]

(1)

\[
T^w_{ik} = T^w_c \cdot T^c_{ik}
\]

(2)

The pose transformation of camera and odometry:

\[
T^w_{ic} = T^w_{ic} \cdot T^c_{ic}
\]

(3)

\[
T^w_{ic} = T^w_c \cdot T^c_{ic}
\]

(4)

\( T^c_{ic} \) is the transformation matrix between the camera and the odometry. The following formula is inferred by the Formula (1),(2),(3),(4).That is:

\[
T^w_{ic} \cdot T^c_{ic} = T^c_{ic} \cdot T^w_{ic}
\]

(5)

In this way, solving the problem of relative pose transformation is simplified to work out the Formula (5). \( T^c_{ic} \) can be figured out by the two parameters \( T^w_{ic}, T^w_{ic} \). Next section will show how to get and process the two parameters.

3. Measurement

The data obtained from the chassis odometry after the basic calculation includes the distance \( x_n \) and \( y_n \), which is relative to the initial position of the x-axis and the y-axis, as well as the angle of rotation relative to the initial position. The frequency of camera is different from that of odometry. The frequency of odometry is about 20 Hz while that of the key frame from the camera is about 2 Hz. In order to ensure the error of odometry and camera pose data in the \( k \)th frame is as small as possible, we use high-frequency odometry data to align with the camera timestamp in approximate linear method [8].
As is shown in Figure 2, the upper half part of the timeline is the camera's timestamp in which $t_0^c$ is the initial frame time and $t_k^c$ is the time of $k_{th}$ frame. Since the timestamp of odometry cannot be exactly equal to that of camera, there are two odometry data before and after the time of the $k_{th}$ frame and the corresponding timestamps are $t_{i}^r$, $t_{i+1}^r$. The statistics of movement on the x-axis are $x_i, x_{i+1}$, then the movement on the x-axis at the time $t_k^c$ is shown as:

$$x_k = x_i + (x_{i+1} - x_i) \times \frac{t_k^c - t_i^r}{t_{i+1}^r - t_i^r}$$

(6)

Similarly, the approximate movement on the y-axis and the angle of rotation can be obtained. Since the sweeping robot moves only on the ground, only the yaw angle needs to be considered. With using these three parameters, the Euclidean transformation matrix is shown as [7]:

$$T_{c_k}^0 = \begin{pmatrix} R_k & t_k \\ 0 & 1 \end{pmatrix}$$

(7)

with:

$$R_k = \begin{pmatrix} \cos \theta_k & -\sin \theta_k & 0 \\ \sin \theta_k & \cos \theta_k & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad t_k = \begin{pmatrix} x_k \\ y_k \\ 0 \end{pmatrix}$$

(8)

The camera captures a key frame every half second. We place a checkerboard with black and white squares above the position under which the robot moves, and establish a coordinate system on the checkerboard to confirm the 3D coordinates (x, y, z) of the four feature points on the black and white grid [9], as well as the coordinates(u,v) of the $k_{th}$ frame correspond to point of the features in the pixel. Then we use the direct linear transformation (DLT) in PnP to solve the $k_{th}$ frame camera’s coordinates and rotation angle in the checkerboard coordinate system[10, 11].The pose is represented as $T_{c_k}^{w_0}$ and that of the initial frame solved by PnP is $T_{c_0}^{w_0}$, then the pose transformation matrix of the $k_{th}$ frame relative to the initial frame is represented as:

$$T_{c_k}^{c_0} = (T_{c_0}^{w_0})^{-1} \cdot T_{c_k}^{w_0}$$

(9)

The two necessary parameters $T_{c_k}^{c_0}$ and $T_{c_0}^{c_k}$ are worked out. Next, the paper will introduce how to solve the Formula (5).

4. Solve the formula

The process of solving this formula refers to classical hand-eye method, namely Tsai-Lenz [12]. The Formula (5) is expanded as following:
\[
\begin{pmatrix}
R_{c_k} \\
t_{c_k}
\end{pmatrix}
\begin{pmatrix}
R_{cr} & t_{cr} \\
0 & 1
\end{pmatrix}
\begin{pmatrix}
R_{cr} \\
t_{cr}
\end{pmatrix}
= 
\begin{pmatrix}
R_{cr} & t_{cr} \\
0 & 1
\end{pmatrix}
\begin{pmatrix}
R_{cr} & t_{cr} \\
0 & 1
\end{pmatrix}
\]

We will get the basic equation of hand-eye calibration:

\[
\begin{cases}
R_{c_k} R_{cr} = R_{cr} R_{c_k} \\
(R_{c_k} - 1)t_{cr} = R_{cr} t_{c_k} - t_{c_k}
\end{cases}
\]

The paper [12] uses the two-step method to solve the equation (11), which means to get \( R_{cr} \) first from the basic equation and then put it into the following equation to get \( t_{cr} \). The rotary axis-rotation angle system describes the rotational motion to solve the system of equations. The Rodriguez transformation changes rotation matrix to rotation vector [13].

5. Experiment procedure and result analysis

An ordinary household sweeping robot is used in the experiment and the upward camera is at the center of the robot. The checkerboard is a black and white square with 20-cm side, which is placed about 70 cm above from the ground. The chassis is from our partner, Lefant home robot company in Shenzhen and the size is T700, the camera is from Sunny optical technology and the size is OV7725 (see Figure 3).

![Figure 3](attachment:image.png)

(a) (b)

**Figure 3.** The experiment equipments, T700 robot chassis and OV7725 camera(a) and the environment of the experiment(b).

| Timestamp/s | X-axis movements/m | Y-axis movements/m | Rotation/rad |
|-------------|--------------------|--------------------|--------------|
| 94.364      | 0.000000           | 0.000000           | 0            |
| 95.596      | 0.150000           | 0.000000           | -0.003491    |
| 96.228      | 0.292400           | 0.000000           | -0.005236    |
| 96.728      | 0.370000           | 0.000000           | -0.005236    |
| 97.894      | 0.380000           | 0.000000           | -0.708150    |
| 98.827      | 0.380000           | 0.000000           | -1.400924    |
| 99.293      | 0.380000           | 0.000000           | -1.404990    |
| 101.892     | 0.380000           | 0.000000           | -2.113594    |
| 102.359     | 0.360000           | -0.037619          | -2.118830    |
| 102.825     | 0.330000           | -0.090000          | -2.125287    |
| 103.791     | 0.330000           | -0.090000          | -2.815216    |
To ensure translation and rotation, we make the robot move forth and back and turn. The experiment uses 11 frames of data. The camera takes 11 pictures at different times and the odometry data at every time is shown in Table 1. Figure 4 shows 11 pictures that are labeled from (a) to (k) in the order of time stamp.

Figure 4. Camera data of the robot.

With Figure 4(a) being taken as an example, coordinate system established on the checkerboard is as shown in the Figure 5, and the z-axis direction is outward. The coordinates of the four selected feature points are A(0,0), B(400,0), C(400,400), D(0,400) (the unit is mm).

Figure 5. Key points selected on the checkerboard frame.

In order to solve the PnP problem, the four feature points ABCD need to be mapped one by one in the pixel coordinates corresponding to 11 certain pictures. The corresponding results are shown in Table 2.

The pose in the checkerboard coordinate system calculated by PnP can figure out transformation matrix $T_{ci}^{ck}$ of the $k_{th}$ frame (relative to the initial frame) by calculating Formula (9).

For example, the pose transformation corresponding to two frames of Figure 4(a) and Figure 4(b) is:
\[
R_{cr}^{10} = \begin{pmatrix}
0.9999823 & 0.0035486 & -0.0047793 & 0.0029733 \\
-0.0035208 & 0.999976 & 0.0057989 & -0.1460874 \\
0.0047998 & -0.0057820 & 0.9999713 & -0.0061724 \\
0 & 0 & 0 & 1
\end{pmatrix}
\] (12)

The transformation matrix of the corresponding odometry from time 94.364s to 95.296s is expressed as:
\[
R_{cr}^{6} = \begin{pmatrix}
0.999994 & 0.0034910 & 0 & -0.0015708 \\
-0.0034910 & 0.999994 & 0 & -0.1499918 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{pmatrix}
\] (13)

Since we have 11 sets of data at different times, we can obtain 10 sets of relative pose data relative to the initial frame. The 10 sets of data into equation (10), (11) are put into the equation and the final result is got:
\[
R_{cr} = \begin{pmatrix}
0.9999921 & 0.0277637 & -0.0028423 \\
-0.0026370 & 0.9988479 & 0.0479156 \\
0.0297203 & -0.0479077 & 0.9988473
\end{pmatrix},
\quad
t_{cr} = \begin{pmatrix}
-0.0225586 \\
-0.0101597 \\
-0.0388593
\end{pmatrix}
\] (14)

Table 2. Pixel correspondence of 4 feature points and 11 photos.

| picture | Time/s  | A_u | A_v | B_u | B_v | C_u | C_v | D_u | D_v |
|---------|---------|-----|-----|-----|-----|-----|-----|-----|-----|
| a       | 94.364  | 203 | 195 | 406 | 201 | 400 | 29  | 223 | 24  |
| b       | 95.296  | 203 | 267 | 406 | 273 | 407 | 81  | 216 | 75  |
| c       | 96.228  | 206 | 338 | 401 | 344 | 411 | 147 | 209 | 140 |
| d       | 96.728  | 209 | 373 | 399 | 378 | 413 | 187 | 207 | 180 |
| e       | 97.894  | 136 | 273 | 277 | 399 | 409 | 262 | 256 | 124 |
| f       | 98.827  | 147 | 149 | 173 | 335 | 361 | 315 | 332 | 111 |
| g       | 99.293  | 147 | 148 | 173 | 334 | 361 | 314 | 332 | 111 |
| h       | 101.892 | 236 | 57  | 136 | 215 | 288 | 322 | 400 | 149 |
| i       | 102.359 | 235 | 71  | 135 | 232 | 288 | 341 | 401 | 169 |
| j       | 102.825 | 233 | 101 | 136 | 264 | 288 | 373 | 402 | 206 |
| k       | 103.791 | 328 | 76  | 150 | 141 | 199 | 321 | 391 | 265 |

The result \(R_{cr}\), rotation matrix, is converted to Euler angle and the roll angle in the x-axis direction is 0.15 degrees, the pitch angle in the y-axis direction is 0.17 degrees and the yaw angle in the z-axis direction is 2.75. In the visual odometry (VO), the result is left multiplied with an external parameter matrix to obtain the true pose state estimation of the sweeping robot.

In fact, we do a lot of experiments with data from 3 sets to 11 sets and we get different results as Figure 6.

Finally, we have a calibration accuracy test as Figure 7.

From the Figure 7, it can be seen that the ground truth (taken by VICON) and the trajectory of chassis odometry are very close. The red line is the visual odometry whose frequency is about 0.5Hz, so the line is not continuous enough and the max error is about 5cm. The work on improving visual odometry is our next plan. However, it indicates that the method of getting transformation matrix of odometry and camera proposed is reliable.
Figure 6. The value calculated from different sets of data, roll(a), pitch(b) and yaw(c).

Figure 7. The trajectories of chassis odometry (green), visual odometry (red) and the ground truth (blue).
6. Conclusions
After the experiment, we find that the accuracy of result is related to the selection of data, which means that the movement of chassis has to include both translation and rotation. Meanwhile, the larger the amount of data is, the more accurate the estimation of results is (As shown in Figure 6). This novel method can figure out transformation matrix of odometry and camera using only one mark board. Besides, it combines PnP with hand-eye calibration to make a new method of calibration between chassis odometry and camera.

References
[1] Taketomi T, Uchiyama H and Ikeda S 2017 Visual slam algorithms: a survey from 2010 to 2016 IPSJ Transactions on Computer Vision and Applications 9(1) 16
[2] Feng Y, Wu Y and Fan L 2017 Real-time slam relocalization with online learning of binary feature indexing Machine Vision & Applications 28(8) p 1-11
[3] G Antonelli, F Caccavale, F Grossi and A Marino 2010 Simultaneous calibration of odometry and camera for a differential drive mobile robot IEEE International Conference on Robotics and Automation Anchorage AK p 5417-5422
[4] Mahmood R A and Jafar K 2017 Point stabilization of nonholonomic spherical mobile robot using nonlinear model predictive control Robotics and Autonomous Systems p 347-359
[5] Deng F, Wu Y, Hu Y and Cui H 2016 Position and pose estimation of spherical panoramic image with improved epnp algorithm Acta Geodaetica et Cartographica Sinica 45(6) p 677-684
[6] Bi S, Yang D and Cai Y 2018 Automatic calibration of odometry and robot extrinsic parameters using multi-composite-targets for a differential-drive robot with a camera Sensors 18(9) 3097
[7] Hall B C 2015 Lie groups, lie algebras, and representations Springer Berlin 659(5) xiv 351
[8] Sun W, Zhang X, Zhang Z and Zhu R 2017 Data fusion of near-infrared and mid-infrared spectra for identification of rhubarb Spectrochimica Acta Part A Molecular & Biomolecular Spectroscopy 171(Complete) p 72-79
[9] Y Dong, X Ye and X He 2016 A novel camera calibration method combined with calibration toolbox and genetic algorithm Conference on Industrial Electronics and Applications (Hefei) p 1416-1420
[10] Marchand E, Uchiyama H and Spindler F 2016 Pose estimation for augmented reality: a hands-on survey IEEE Transactions on Visualization & Computer Graphics 22(12) 2633-2651
[11] Hamel T and Samson C 2017 Riccati observers for the non-stationary pnp problem IEEE Transactions on Automatic Control p 726-741
[12] Malti and Abed 2013 Hand-eye calibration with epipolar constraints: application to endoscopy Robotics and Autonomous Systems 61(2) p 161-169
[13] Yang F, Dai H and Xing H 2015 Least squares based on Rodrigues Matrix and its application in similar material model of mining IEEE International Conference on Mechatronics and Automation (Beijing) p 1686-1690