Camera to Robotic Arm Calibration

Zachary Taylor, ACFR, University of Sydney

z.taylor@acfr.usyd.edu.au

December 10, 2014

This technical report gives a method for calibrating a camera to work with a robotic arm so that the location of the arms end effector can be projected into the cameras output. This report complements a Matlab toolbox that was developed to perform the calibration. The method presented here was inspired by but different to that given by [1].

1 Projection of end effector

The position of a point relative to the arms end effector can be projected into the cameras image using equation 1.

$$\boldsymbol{p_{cam} = P_{cam} T^{cam}_{base} T^{base}_{end} p_{end}} \tag{1}$$

Where

$\boldsymbol{p_{end}}$ is the points position relative to the end effector
$\boldsymbol{T^{base}_{end}}$ is the transformation from the robotic arms base to its end effector
$\boldsymbol{T^{cam}_{base}}$ is the transformation from the camera to the robotic arms base
$\boldsymbol{P_{cam}}$ is the cameras projection matrix
$\boldsymbol{p_{cam}}$ is the position of the point in the cameras image

The above equation however assumes a perfect pin hole camera model. Real cameras have lens distortions however that must be account for by equation [2] 2

$$\begin{aligned} \boldsymbol{r} &= \boldsymbol{\sqrt{x^2 + y^2}} \\ \boldsymbol{x_d} &= \boldsymbol{x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + 2p_1 xy + p_2 * r^2 + 2x^2} \\ \boldsymbol{y_d} &= \boldsymbol{y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + p_1 (r^2 + 2y^2) + 2p_2 xy} \end{aligned} \tag{2}$$

Where
$\boldsymbol{k}$ are the radial distortion coefficients
$\boldsymbol{p}$ are the tangential distortion coefficients
$\boldsymbol{x}$ is the x position of the points in normalised camera coordinates
$\boldsymbol{y}$ is the y position of the points in normalised camera coordinates
$\boldsymbol{x_d}$ is the distorted x position of the points in normalised camera coordinates
$\boldsymbol{y_d}$ is the distorted y position of the points in normalised camera coordinates

2 Calibration Methodology

A checkerboard is mounted on the end of the end effector. The corner points of this checkerboard can be robustly detected in the image. The location of its points can also be projected into the image using

$$\boldsymbol{p_{cam}} = \boldsymbol{P_{cam}} \boldsymbol{T_{checker}^{end}} \boldsymbol{T_{base}^{cam}} \boldsymbol{T_{end}^{base}} \boldsymbol{p_{checker}} \tag{3}$$

Where
$\boldsymbol{p_{checker}}$ is a corner point on the checkerboard
$\boldsymbol{T_{checker}^{end}}$ is the transformation from the robotic arms end effector to the checkerboard

Thus the difference in the position of the detected checkerboard and the board projected into the cameras image will be minimised when we have found the correct value of $\boldsymbol{T_{base}^{cam}}$.

3 Camera Calibration

If the camera matrix and distortion parameters are known these values are used. However if the values are unknown then they are estimated using standard checkerboard camera calibration techniques provided by Matlab [3].

4 Optimisation

As equation 3 contains two unknown transformations $\boldsymbol{T_{checker}^{end}}$ and $\boldsymbol{T_{end}^{base}}$ no simple analytical method for solving it exists. Instead we make use of the interior point optimisation method to find the parameters that minimise the projection error of the points. To do this we form a 13 element vector of the unknowns, 6 elements represent each transformation and the last element is the size of the checkerboard squares. The square size is included as measuring the squares with sufficient accuracy can be challenging.

4.1 Transformation vectors

When optimising a transformation matrix, the problem is greatly simplified if it is represented as a minimal 6 element vector. In our implementation the first 3 elements of this vector are simply the translation $[\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}]$. The final 3 elements represent an angle-axis rotation vector $[\boldsymbol{ax}, \boldsymbol{ay}, \boldsymbol{az}]$ where the elements give the axis to rotate about and their magnitude $||[\boldsymbol{ax}, \boldsymbol{ay}, \boldsymbol{az}]||_{\mathbf{2}}$ gives the angle to rotate by.

4.2 Trimmed means

The error given by our optimisation is the trimmed means representation of the error in the projection. This is formed by discarding the worst n points in the matches before taking the mean to give the overall error. In our default implementation we discard the worst 20 percent of the points. This is done so that effects such as large blur in an image or a single board being matched to the background do not prevent an accurate solution being found. Trimmed means was used over a simple median as we found the median was more likely to cause convergence to local minima and typically gave less accurate results.

5 Error Estimation

The first question asked when providing a calibration is generally how accurate is it? To provide some indication of this we bootstrap the data and rerun the optimisation 100 times. The results of these optimisations are used to estimate the standard deviation of the estimate. This is converted from the 6 element transformation vector into a transformation matrix using a Monty Carlo sampling method.

6 Code

A Matlab toolbox to perform the calibration and an example dataset can be found at `https://github.com/ZacharyTaylor/Camera-to-Arm-Calibration`.

References

[1] O. Nielsen and E. Pedersen, “Calibration of a robotic arm,” 2010. [Online]. Available: http://robolabwiki.sdu.dk/mediawiki/index.php/Calibration_of_a_robotic_arm

[2] J. Weng, P. Cohen, and M. Herniou, “Camera calibration with distortion models and accuracy evaluation,” *Pattern Analysis and Machine* . . . , 1992. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=159901

[3] Z. Zhang, “A flexible new technique for camera calibration,” *Pattern Analysis and Machine Intelligence, IEEE* . . . , vol. 1998, no. 11, pp. 1330–1334, 2000. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=888718