CIS II Paper Review

Mengze Xu

March 2017

1 Introduction

1.1 Project Background

Our project is DVRK Stereo Camera Calibration and Model Registration. First we want to do a hand-eye calibration between da vinci PSM and Intel RealSense Stereo Camera. Then we'll try to register known and unknown surface to the robot system through the stereo camera.

To complete hand-eye calibration, we need to know the pose of PSM which is provided by DVRK API and the pose of the camera which will be provided by ArUco Library. Then it becomes a "AX=YB" problem which is solvable.

1.2 Paper Selected

I pick two papers, one for ArUco Library and one for solving "AX=YB". [1] is the main paper related to ArUco Library used in our project and [2] is one of the most updated paper with open sourced codes.

[1] Garrido-Jurado S, Muñoz-Salinas R, Madrid-Cuevas F J, et al. Automatic generation and detection of highly reliable fiducial markers under occlusion[J]. Pattern Recognition, 2014, 47(6): 2280-2292.

[2] M. Shah, Solving the Robot-World/Hand-Eye Calibration Problem Using the Kronecker Product, ASME Journal of Mechanisms and Robotics, Volume 5, Issue 3 (2013).

2 ArUco Library

2.1 Summary

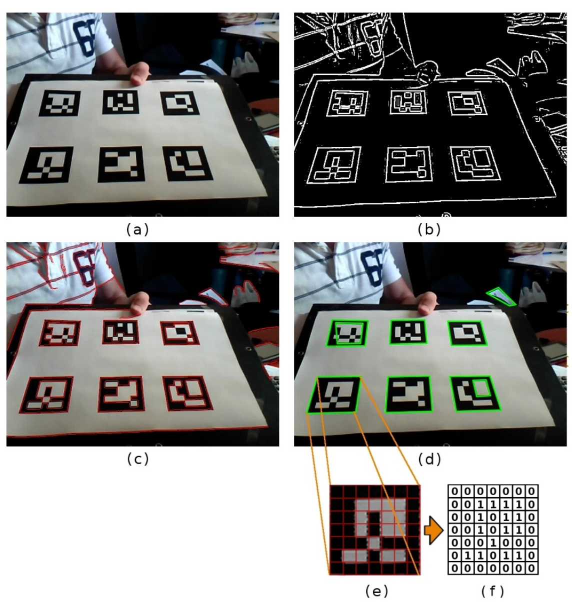
This paper provides methods and codes to estimate the pose of camera to multi-objects using fiducial markers. The method is easy to complete and convenient to use, providing a useful tool to do hand-eye calibration. It's also more accurate than similar codes.

This paper mainly solves three detailed problem: how to generate markers, how to detect markers from image and how to solve occulsion problem. Since only the second problem is relative to our project, we mainly talks about this part.(Section 4 and 6)

2.2 Method

I'll shortly talks about the method to estimate the pose of the camera from image.Just as the figure below shows, the algorithm run as

- Detect edge points in (a) to get (b) based on local gradient intensity.
- Extract contours from edge points to get (c) using Suzuki and Abe algorithm.
- Filter and only keep related rectangular contours to get (d) using Douglas and Peucker algorithm.
- With edge points and prior known size of the marker, use least square to estimate the pose of the camera.
- Project the marker using estimated pose to get binary codes and decode it to get the ID of the marker (e & f)
- It should be noticed that the detection of the marker ID also works as a check that we detect the right figure.



2.3 Assessment

This paper provides useful tools to application in AR an robotics. For our project, it provides a very convenient tools compared to calibration objects or touching fiducial points. This paper also clearly describe their method and has some discussion about error estimation.

For our purpose, this paper may talks too little about error estimation. It only gives average error in images by pixel and is lack of error estimation of the pose. Also this paper doesn't talk about those factors which influence the error. So we can't have any error estimation of our hand-eye calibration before we complete it and compared to ground truth.

3 Solving AX=YB Problem

3.1 Summary

This paper provide a closed form solution to AX=YB problem and test result of simulation and real data. Closed form solution is easy to complete and very fast to compute. Their method works a little better than former methods. This paper provides a solver for our codes.

3.2 Method

To solve $A_i X = Y B_i$ problem, we need at least $n >= 3$ pair of A and B and the algorithm works as

- Calculate Kronecker Product $= \sum_1^n R_{A_i} \otimes R_{B_i}$
- Singular Value Decomposition $K = U \Sigma V$
- n is proofed to be the largest singular value of K, set u and v the correspond singular vector in U and V
- Transform the twist u and v into rotation matrix R_B and R_A.
- Solving t_A and t_B using least square.

3.3 Simulation and Real Data

This paper does simulation as these: first generate ground truth of X and Y, and then generate correspond A and B. Random noise was added to A and B and then did the calculation.

η is relative error intensity. According to their result, we need to make sure the relative error of camera pose and robotic pose is less than 2% to get desired accuracy.

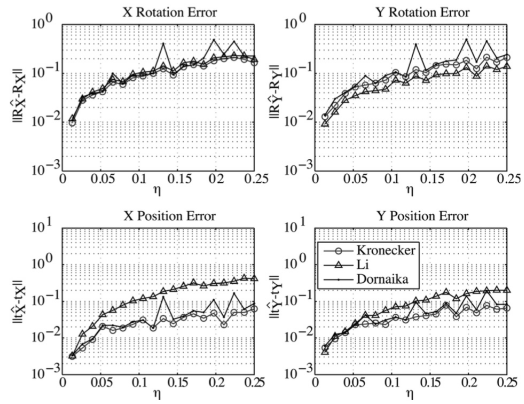


Fig. 2 Comparison of the Kronecker product method described in this paper (circles), the Li et al. Kronecker product method described in Ref. [9] (triangles), and the Dornaika and Horaud closed-form quaternion method described in Ref. [4] (solid line) on simulated data.

The real data comes from a system whose Y is measured directly by another camera as ground truth. Since our project have totally different setup, I do not talk about it.

3.4 Assessment

This paper provides a easy-to-complete closed form solution to $AX = YB$ problem in our project. However, it works only a little better than former method and doesn't provide more advice for hand-eye calibration such as how to choose appropriate A and B.

4 Things Helpful

Here are several open sourced codes. It will be helpful if you also want to complete this.

http://wiki.ros.org/aruco
http://terpconnect.umd.edu/ jwelsh12/enes100/markergen.html
http://math.loyola.edu/ mili/Calibration/AXYB/shah.m