

# Homework 1 - AutoCalib

## Report

Using 1 late day

Sigurthor Bjorgvinsson  
Department of Computer Science  
University of Maryland  
College Park, Maryland 20740

### I. INTRODUCTION

The aim of this project was to implement the algorithm presented by Zhengyou Zhang of Microsoft. Zhang's method as it is usually called is an automatic way to perform camera calibration. In this project, I attempted at implementing his algorithm to estimate  $f_x$ ,  $f_y$ ,  $c_x$  and  $c_y$  for the  $K$  matrix, Rotation and translation matrices for each homography, and the distortion  $k_1$  and  $k_2$ .

#### II. 4.1. SOLVING FOR APPROXIMATE $K$ OR CAMERA INTRINSIC MATRIX

To find out the camera intrinsic matrix I first had to get all of the corners from the images. The images had a grid of  $8x11$  squares and I extracted  $6x9$  squares from them. I generated a reference corner array which had  $21.5px$  spacing as per the width of each square. Next I generated homography matrices for each image which was the homography to translate the reference corners to the image corners. Instead of using cv2 I wanted to try to expand on my failed attempt to at generating a homography in the panorama project to using least\_squares. With a little help from the internet I was able to generate a accurate and optimized homography. [1]

With the homographies, I created a array of  $26x6$  values to solve the 6 values in the  $b$  array which was used to construct the  $K$  matrix, the  $K$  matrix is constructed of 5 values calculated using the values from the  $b$  array,  $f_x(\alpha)$ ,  $\gamma$ ,  $c_x$ ,  $f_y(\beta)$  and  $c_y$ . The  $K$  matrix is shown below:

$$\begin{bmatrix} 2062.502297 & -0.693072 & 754.068056 \\ 0.0 & 2051.710108 & 1362.499319 \\ 0.0 & 0.0 & 1.0 \end{bmatrix}$$

#### III. 4.2. ESTIMATE APPROXIMATE $R$ AND $t$ OR CAMERA EXTRINSICS

To estimate rotation and translation matrices I used the  $K$  matrix and the homographies. With the test images, I got 13 rotation matrices and 13 translation matrices. One anomaly which did not fit the paper was that  $\lambda r1$  and  $\lambda r2$  were not equal. Later in this project I experienced great issues with translation which I believe was because of this inequality. To get around that I averaged  $\lambda r1$  and  $\lambda r2$  for  $\lambda r3$ .

### IV. 4.3 APPROXIMATE DISTORTION $k_c$

In the project description, it was said that a good estimate for the distortion was to start with  $k_c = [0, 0]^T$ . I was unable to use this approach to optimize the distortion and ended up using the numpy least square function and solve for equation 11 and 12 from the Zhang paper[2]:

$$\begin{bmatrix} (u - u_0)(x^2 + y^2) & (u - u_0)(x^2 + y^2)^2 \\ (v - v_0)(x^2 + y^2) & (v - v_0)(x^2 + y^2)^2 \end{bmatrix} \begin{bmatrix} k_1 \\ k_2 \end{bmatrix} = \begin{bmatrix} \check{u} - u \\ \check{v} - v \end{bmatrix}$$

The values of  $k_1$  and  $k_2$  after solving the equation was:

$$[5.12447186e^{-11} \quad -2.79156402e^{-18}]$$

### V. 5. NON-LINEAR GEOMETRIC ERROR MINIMIZATION

This caused a great headache for me to understand and I was unfortunately not able to figure this out. Understanding what parameters I was supposed to optimize and how to do it was difficult. I tried to use the minimize function from scipy.optimize and tried to apply it to all the inputs,  $K$ ,  $R_i$ ,  $t_i$  and  $k_s$  separately but none of them was successful. I left my attempts to optimize the parameters in the code so that you can see that I really tried. I hope that will be taken into consideration.

### VI. RESULTS

The images after rectification and reprojection of the reference corners on the image looked fantastic in my opinion. What I don't understand is the large euclidean error that I experienced. I calculated a euclidean error of 334261.336762 which I do not feel reflects the accuracy shown on the images below. The images were created by applying the undistort function from cv2 with the camera matrix  $K$  and the distortion and then I translated each point using the rotation and translation matrix and the  $K$  matrix.

### VII. CONCLUSION

After spending way to much time on this project, I have felt that I learned quite a lot. The knowledge that I gained will not be reflected in this report given the midnight deadline I am trying to make. I hope the results speak for them selfs but unfortunately I won't be able to show the sweat, tears and frustration that I put into this project. I hope that the results are enough to show my work.

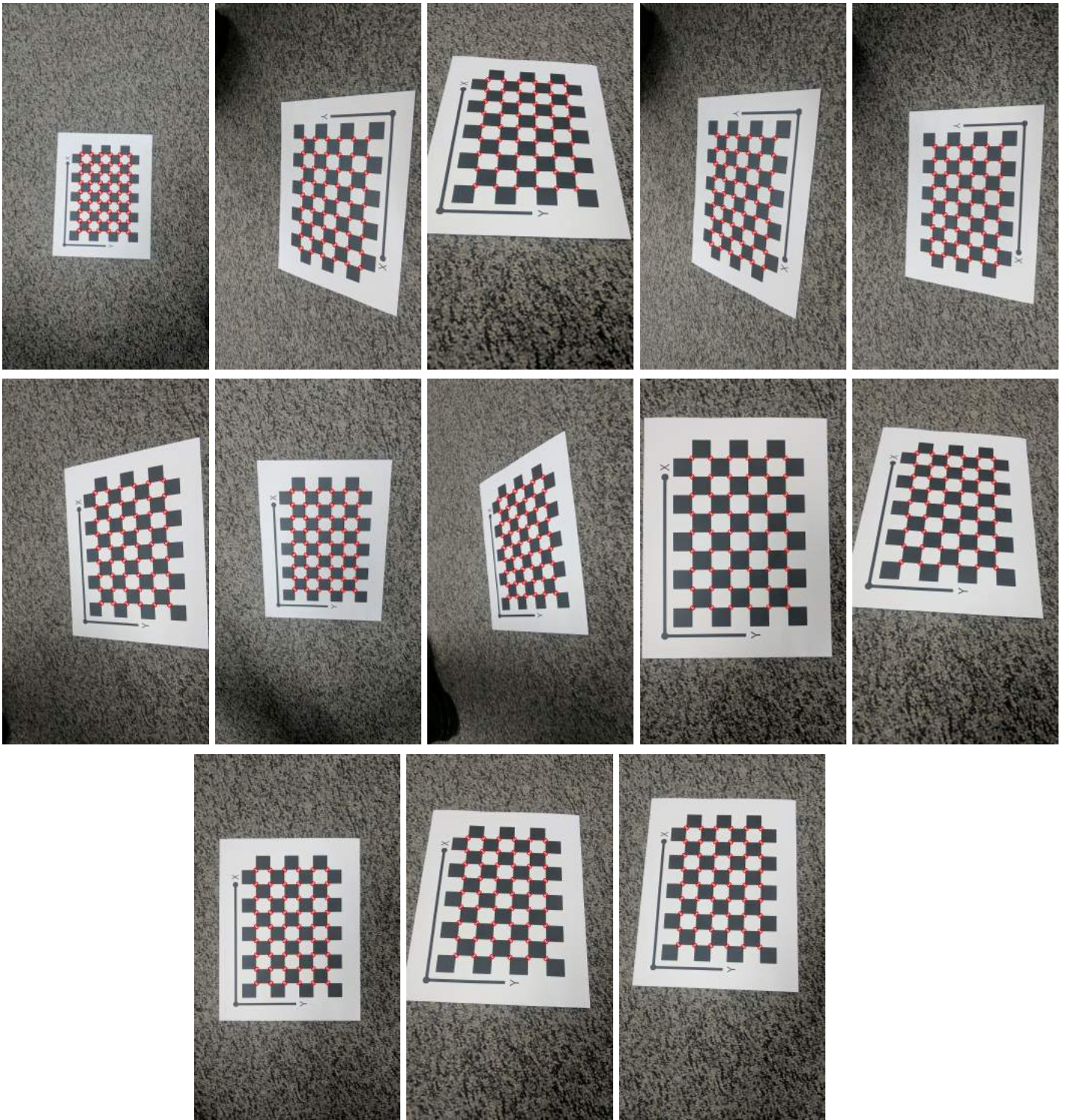


Fig. 1. Checkerboards after rectification and reprojection of corners on rectified image

#### REFERENCES

- [1] Demystifying Geometric Camera Calibration for Intrinsic Matrix - <https://kushalvyas.github.io/calib.html>
- [2] A Flexible New Technique for Camera Calibration by Zhengyou Zhang <https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/tr98-71.pdf>