Project Report

Anirudh Anumalasetty Venkata, Vijay Muthukumaran

December 7, 2016

Abstract

Calibrate camera, create epipolar lines to find depth in a set of 3 images. Use stereo mapping to obtain distance map.

1 Introduction

The camera used in a smart phone can give us high resolution images with the acquisition parameters of our choice that lets us decide the brightness of the image. However, there is no relation to how large an object in an image is in any real world units. One of the purposes of this project is to calibrate the camera, and get the relationship between pixel units and real world units. The image must also be corrected for radial distortion introduced by the lens. The project also focuses on depth estimation and stereo matching algorithms.This report includes the work done for the final project submission. All the parts from camera calibration to multi-view stereo are explained in steps with details of implementation.

2 Report Layout

The report focuses on 6 different sections-

1. Camera calibration with a chessboard.
2. Estimation of epipolar geometry.
3. Retrieving the extrinsic parameters of the cameras.
4. Rescaling the parameters.
5. Plane sweeping algorithm for stereo matching.

3 (Part 1) Camera Calibration

The images were taken using an iPhone 7 camera with a locked optical zoom and fixed exposure. The calibration was done by detecting the inner edges of the black and white squares whose length in real world units was known.

The parameters for calibration were:-

- Square size - 2.35cm
- ISO - 100
- corners along length - 9
- corners along breadth - 6

The camera calibration was done with openCV functions which produced the intrinsic camera matrix and the distortion coefficients which are used in later parts of these projects. The average re-projection error obtained was 1.09 in pixel units.

3.1 Deliverable

- Images

  The images used for calibration are shown below:-

Figure 1: Images used for calibration

- Intrinsic matrix

$$10^3 * \begin{pmatrix} 3.29 & 0 & 1.29 \\ 3.29 & 1.47 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

- Distortion coefficients

$$\begin{pmatrix} 8.48 * 10^{-2} \\ -2.46 * 10^{-1} \\ -6.14 * 10^{-3} \\ -3.07 * 10^{-4} \\ 1.31 * 10^{-1} \end{pmatrix}$$

- The reprojection mean square error is 1.09 pixels.

4 (Part 2) Take Pictures

The final three images chosen were:-

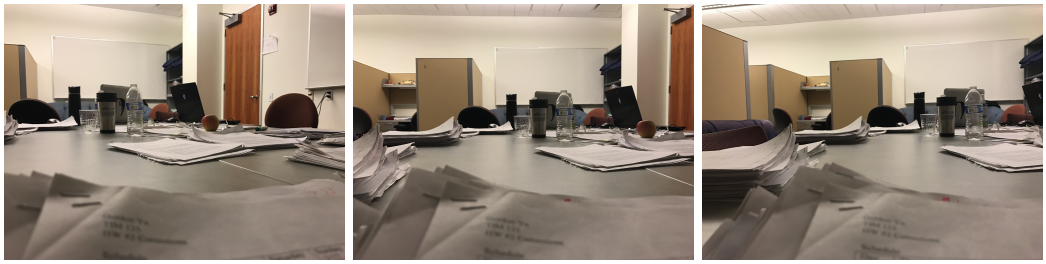

Figure 2: Images from the left and center and right camera

5 (Part 3) Composition of Essential Matrix for Three Camera Pairs

The essential matrix needs to be computed for each image pair (1-2, 2-3, 1-3) following which the epipolar lines need to be drawn with inliers and outliers.

5.1 Process

- The images are first undistorted by obtaining the optimal camera matrix and undistorting the images. The camera matrix and distortion coefficients are obtained from the camera.yml file created from part 1.
- Matching points are selected across each image pair. They are highlighted using red rectangles.

- The fundamental matrix is computed using which the epipolar lines are also computed.
- The function $findFundamentalMat$ gives the mask which contains the details on the number of inliers and outliers.
- The inliers are then plotted on the images in green color.
- The fundamental matrix is then used to compute the conjugate epipolar lines which are then drawn on the images in green colored lines.
- Essential matrix is computed for all images using $findEssentialMat$.

5.2 Deliverable

- Images for the three pairs chosen

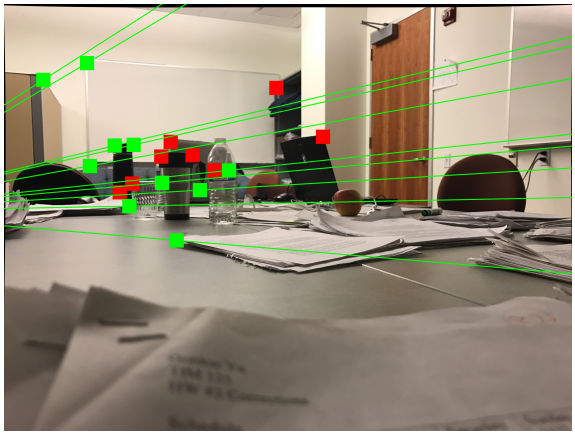

Figure 3: Image from the left camera

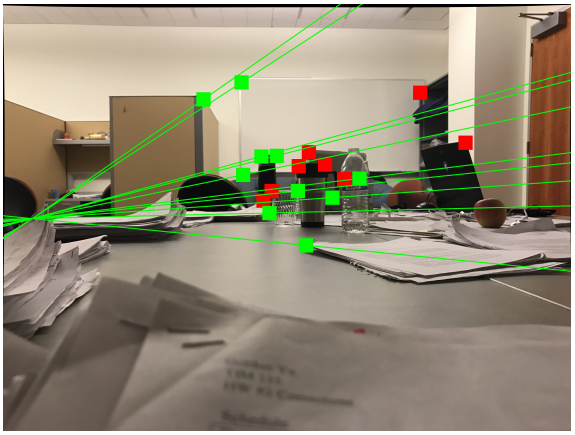

Figure 4: Image from the center camera

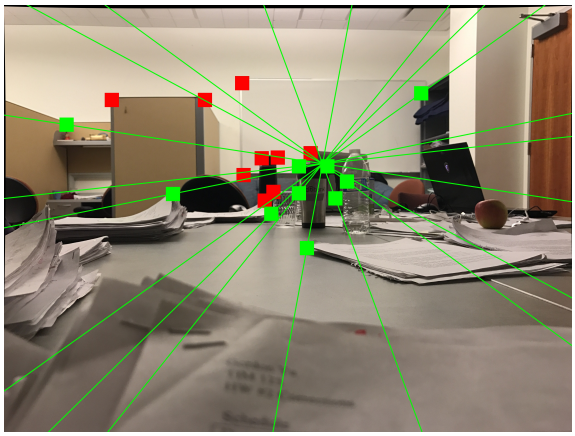

Figure 5: Image from the center camera

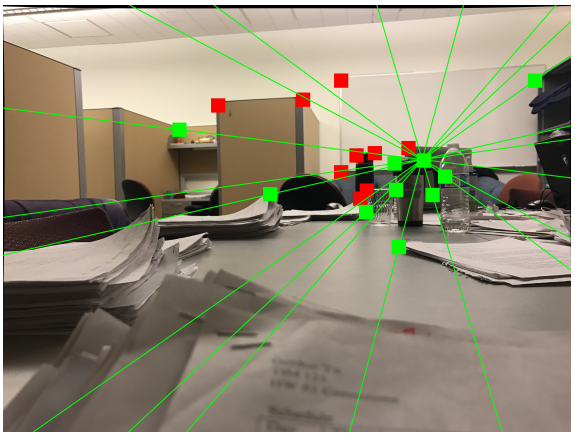

Figure 6: Image from the right camera

- The inliers are shown in red and the outliers are shown in green. The conjugate epipolar lines are also shown in both images.

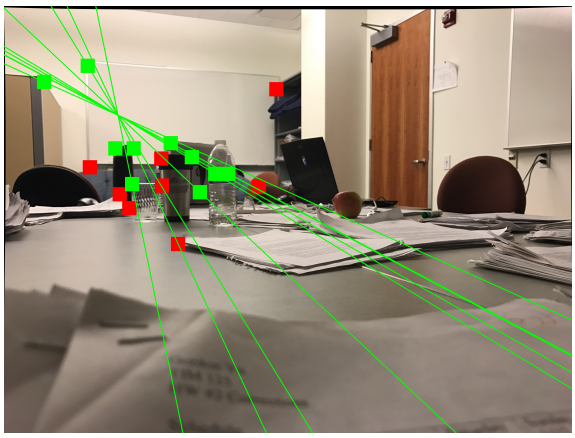

Figure 7: Image from the left camera

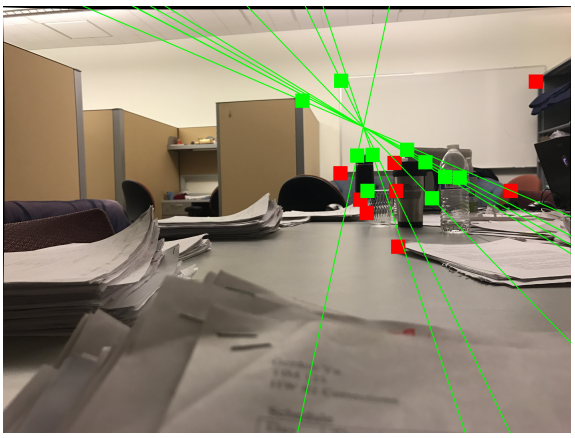

Figure 8: Image from the right camera

- The essential matrices computed for the 3 image pairs are:-
  - Image Pair 1-2

$$\begin{pmatrix} 0.02 & 0.689 & 0.027 \\ -0.705 & 0.030 & -0.04 \\ -0.017 & -0.15 & -0.006 \end{pmatrix}$$

  - Image Pair 1-3

$$\begin{pmatrix} -0.04 & -0.01 & 0.16 \\ 0.19 & 0.006 & -0.658 \\ -0.15 & 0.687 & -0.03 \end{pmatrix}$$

  - Image Pair 2-3

$$\begin{pmatrix} 0.23 & -0.26 & -0.39 \\ -0.02 & -0.06 & 0.54 \\ 0.332 & -0.513 & 0.191 \end{pmatrix}$$

6 (Part 4) Find Extrinsic Parameters

The essential matrix E and the Fundamental matrix F was now available. Using this , the extrinsic parameters a.k.a the relative pose of the camera (Rotation and Translation) can be estimated. The following part of the section describes how the parameters were estimated from the essential matrix.

- The svd of the essential matrix E was computed to and the eigen vectors of E viz. U and V^T was computed along with the singular value matrix Σ such that E $= U * \Sigma * V^T$. The third Singular value is expected to be zero, since E is a rank 2 matrix. Oftentimes in the real world the essential matrix obtained is full rank. This is because of noise present in the real data which acts independently on the images, thus making it impossible for one of the columns to be linearly dependent on the other two. We then force the last singular value to be zero and recompute the essential matrix.

- There are two candidates for the rotation matrix. These two candidates are represented by R_1 & R_2. Where $R_1 = U * W * V^T$ and $R_2 = U * W^T * V^T$ since R_1 & R_2 are the only matrices that satisfy the essential matrix decomposition, namely $E = R_L^R[r^L]_X$. The translation vector retrieved from the essential matrix is defined only upto a scale factor and a sign factor. r^L is the translation vector which is the eigen vector of E with a null eigen value $= -U_3$. Hence there are 4 candidate pairs for the extrinsic parameters. They are R_1, U_3, $R_1, -U_3$, R_2, U_3, $R_2, -U_3$.

- Only one of those 4 possible pairs give rise to a positive depth p_z of the matching pixels, and hence becoming the desired solution.

- The depth is computed as

$$p_z^L = -f_L \frac{f_r r_x^R - x_r^R r_Z^R}{(f_r (R_L^R)^1 - x_r^R (R_L^R)^3) x_l^L}$$

- The matrix W = $\begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$

- Then the corresponding pair that gives a positive p_z is chosen as the solution to the extrinsic parameters.

6.1 Deliverable

1. The Rotation matrix and translation vectors are listed below.
   - Pair 1-2:-

$$Rotation = \begin{pmatrix} 0.441 & -0.853 & 0.277 \\ 0.893 & 0.446 & -0.485 \\ -0.824 & 0.269 & 0.959 \end{pmatrix}$$

$$Translation = \begin{pmatrix} 0.215 \\ -0.016 \\ 0.976 \end{pmatrix}$$

   - Pair 2-3:-

$$Rotation = \begin{pmatrix} 0.963 & -0.0097 & 0.265 \\ 0.006 & 0.999 & -0.012 \\ -0.265 & 0.010 & 0.963 \end{pmatrix}$$

$$Translation = \begin{pmatrix} 0.970 \\ -0.240 \\ 0.016 \end{pmatrix}$$

   - Pair 1-3:-

$$Rotation = \begin{pmatrix} -0.11 & -0.833 & 0.540 \\ -0.956 & -0.241 & -0.163 \\ 0.266 & -0.497 & 0.825 \end{pmatrix}$$

$$Translation = \begin{pmatrix} 0.660 \\ 0.621 \\ -0.42 \end{pmatrix}$$

2. The Depths computed for the Inliers are
   - Pair 1-2

$$Depths = \begin{pmatrix} 16.9 \\ 12.9 \\ 9.36 \\ 12.12 \\ 13.22 \\ 15.4 \\ 19.22 \\ 17.93 \\ 126.8 \\ 66.375 \end{pmatrix}$$

– Pair 2-3

$$Depths = \begin{pmatrix} 2.17 \\ 32 \\ 2 \\ 1.8 \\ -1.6 \\ 1.9 \\ 1.47 \\ -1.41 \\ -1.63 \\ 1.08 \end{pmatrix}$$

– Pair 1-3

$$Depths = \begin{pmatrix} 0.883 \\ 1.0 \\ 1.4 \\ 1.5 \\ 1.822 \\ 1.9 \\ 2.47 \\ 3.51 \\ 3.47 \\ 4.17 \end{pmatrix}$$

- The Reprojection of the points for the different image pairs are shown below.

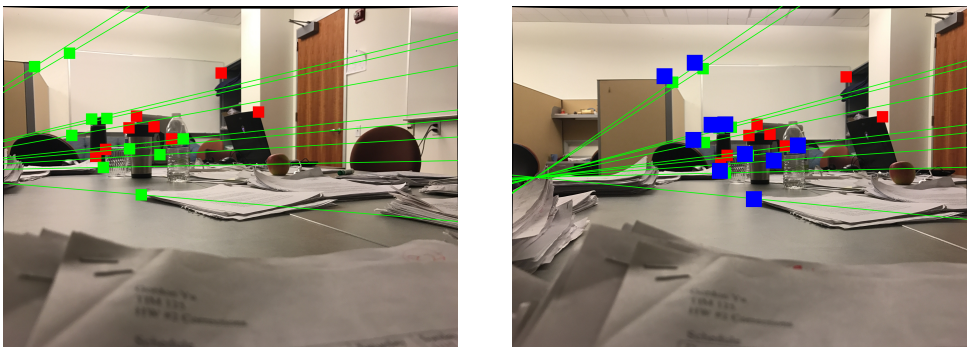

Figure 9: Left-center camera image pair , inliers in green, reprojected points in blue

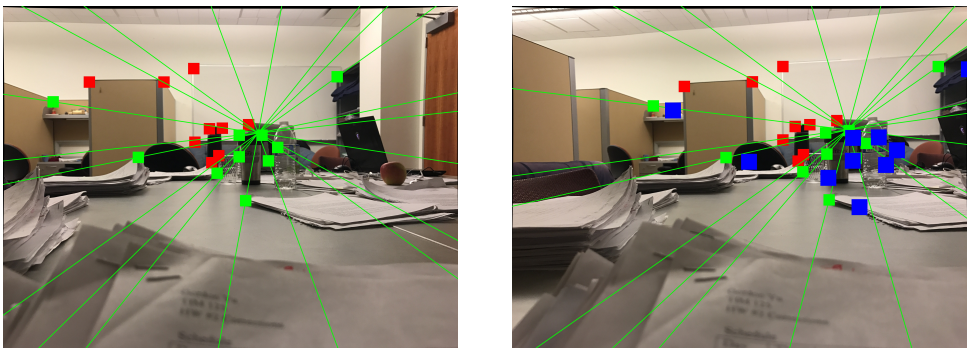

Figure 10: Center-Right camera image pair, inliers in green , reprojected points in blue

- The Reprojection in the third image pair is an example of an error in estimation in the essential matrix.The reprojected points do not coincide with the inliers This could be because of noise, lack of texture, and coplanar points.

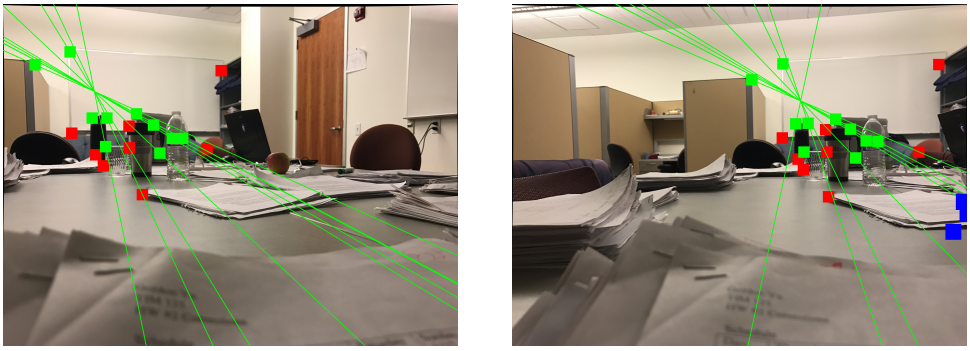

Figure 11: Left-Right camera image pair, inliers in green , reprojected points in blue

7 (Part 5) Rescale the Translation Vector

This part ensures that the sum of the three translation vectors with respect to one camera frame is equal to zero. But in real world, these conditions are not possible as the value obtained for each vector has noise in them which leads to the sum not being equal to zero. By normalizing the vectors with respect to one camera frame one can ensure that the sum of the vectors will be as low as possible.

7.1 Process

- The three translation vectors r_{12}^2, r_{23}^3 and r_{13}^3 and rotation vectors R_1^2, R_2^3 and R_1^3 are obtained from part 4.
- These vectors need to be transformed to the camera coordinates of the first camera as r_{12}^1, r_{23}^1 and r_{13}^1
- These transformations are done as - $r_{12}^1 = R_2^1 r_{12}^2, r_{23}^1 = R_2^1 r_{23}^3$ and $r_{13}^1 = -R_1^3 r_{13}^3$
- The vectors are verified to have unit norm.
- The sum of the three vectors as $r_{12}^1 + r_{23}^1 + r_{13}^1$ is computed. The norm of this sum is computed and it is verified that the value is not equal to zero and is = 2.36.
- We must now find β and γ such that we can minimize the function $F = ||r_{12}^1 + \beta r_{23}^1 + \gamma r_{13}^1||^2$ which is done by taking the gradient of the function with respect to β and γ.
- On taking the derivative with respect to β we get $2\beta(r_{23}^1)^T r_{23}^1 = 0$. This can be substituted into F in order to compute for γ as $\gamma = -\frac{(r_{12}^1)^T r_{23}^1}{(r_{13}^1)^T r_{23}^1}$
- On taking the derivative with respect to γ we get $2\gamma(r_{13}^1)^T r_{13}^1 = 0$. This can be substituted into F in order to compute for β as $\beta = -\frac{(r_{12}^1)^T r_{13}^1}{(r_{13}^1)^T r_{23}^1}$
- The sum of the vectors are re-scaled as $r_{12}^1 + \beta r_{23}^1 + \gamma r_{13}^1$.
- The norm of this sum is computed and found to be = 1.15.

7.2 Deliverables

The values of $\beta = -0.611$ and $\gamma = -0.047$ were found and it can be seen that the value of $||r_{12}^1 + r_{23}^1 + r_{13}^1|| = 2.36$ is greater than $||r_{12}^1 + \beta r_{23}^1 + \gamma r_{13}^1|| = 1.15$

8 Plane sweeping stereo

Plain sweeping stereo is used to compute the distance map. This is done by finding a plane parallel to the left camera focal plane at every depth and computing the SAD for the image pair and finding the minimum SAD value for every pixel. The methodology was-

- The homography matrix is computed using $H = R_1^2 - (r_{12}^1 n^T)/d_i$ where $n = (0, 0, -1)$ and the distance of the plane is d_i. This is similarly done for image pair 1,3 with its rotation matrix $-R_1^3$ and translation vector γr_{13}^1.
- The images 2, 3 are then warped using the homography matrix and *warpPerspective* function in OpenCV.
- For every plane distance, the images are warped and then the absolute difference between the new warped image and left image is computed.
- A box filter with kernel size of 31 is run along this to compute the convolution among the difference between the warped image and left image. This is similar to computing the SAD and finding the minimum block.
- This is done for different plane's at depths that go from range : 1 to f. Therefore, for every new plane, the homography matrix is computed, each image is warped. Difference between the warped and left image is computed and SAD for each pixel is undertaken. This is stored into a array of matrices called SAD_{1-2} and SAD_{1-3}.
- The minimum value for each pixel from this is then found and stored into a separate matrix. This matrix now contains the distance map for each pixel.

8.1 Deliverables

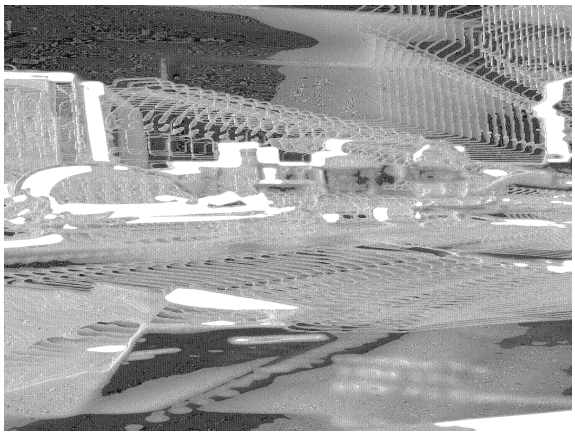

Figure 12: Distance Map for left and center camera

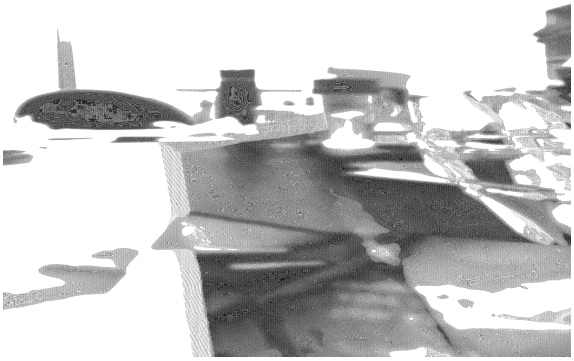

Figure 13: Distance Map for the right and center camera

There is a lot of noise and errors. Which could be due to a lot of points being co-planar and a dearth of unique points.

9 Multi-View Stereo

The multi-view stereo can be applied as an extension of the plane-sweeping stereo described in the previous section.

- The SAD_{1-2-3} can be computed as :-

$$SAD_{1-2-3} = SAD_{1-2} + SAD_{1-3}$$

- The new SAD is used to compute a distance map where the brightness of each pixel in the distance map is indicative of the depth at which the disparity that minimizes the SAD is found.

9.1 Deliverables

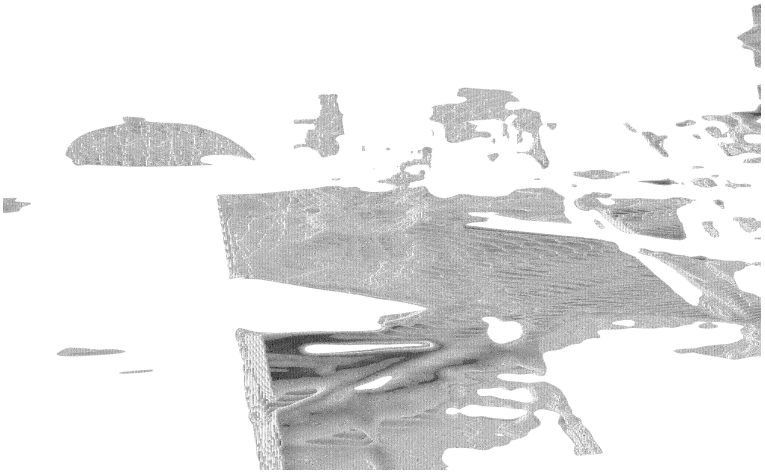

Figure 14: Distance map for the sum of SADs computed from Part 6

10 Result

Thus after the above mentioned steps were performed, the cameras were calibrated. The pose was retrieved and stereo matching was performed as stated.