Namrata Vivek Raghavan
nrgahav@purdue.edu

Projective Scene Reconstruction

In the following assignment we are required to create a projective reconstruction of a scene using two stereo images taken using an uncalibrated camera.

Procedure for scene reconstruction from two stereo images

I. Image Rectification

Extraction of Manual Correspondences

Manually extract a minimum of 8 sets of corresponding points between the two stereo images. For this experiment I have used a total of 10 corresponding points between the two images. I have extracted the correspondence points using IrfanView.

Estimation of Fundamental Matrix

Once the correspondence points have been manually extracted between the two points, the following are the steps to calculate the fundamental matrix:

1. The correspondence points from each image needs to be normalized using a normalizing matrix T. Given that x' is the corresponding coordinate from the right image and x is the corresponding coordinate from the left image. The normalized points are given by
$$\hat{x} = Tx \;\; \hat{x}' = T'x'$$
Here T and T' are calculated such that all the correspondence points have a mean of 0 and the are at a distance of $\sqrt{2}$ from the center.

2. Let x' be the (not normalized) correspondence points from the right image while x (not normalized) are the ones from the left image. The fundamental matrix F is given by the following formula
$$x_i' \, F x_i = 0$$
Where, $x' = [x' \; y' \; 1]^T$ and $x = [x \; y \; 1]^T$
Thus, from N correspondence points we get
$$A.f = \begin{bmatrix} x_1'x_1 & x_1'y_1 & x_1' & y_1'x_1 & y_1'y_1 & y_1' & x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_N'x_N & x_N'y_N & x_N' & y_N'x_N & y_N'y_N & y_N' & x_N & y_N & 1 \end{bmatrix} f = 0$$
$$f = [f_{11}f_{12}f_{13}f_{21}f_{22}f_{23}f_{31}f_{32}f_{33}]$$
Thus, f can be solved by conducting the Single value decomposition of A and found using SVD as the eigen vector corresponding to the smallest eigen value.

3. F needs to rank deficient and needs to have a rank of 2. The rank of F can be conditioned to 2 by using SVD of F. $F = U \, D \, V^T$, to set the rank of F to 2, the smallest singular value in D is set to 0. Then $F_{cond} = U \, D_{cond} \, V^T$, D_{cond} has the smallest singular value is set to 0.

Namrata Vivek Raghavan
nrgahav@purdue.edu

4. Since F was calculate using normalized points, we need to denormalize the matrix. Denormalizing F can be doing using the following formula.

$$F = T'^{T} F_{cond} T$$

Estimation of projection matrices in the canonical form from Fundamental matrix

Once the fundamental matrix F is calculated the projection matrices can be calculated using the following formulas,

$$F.e = 0 \ , e'^{T}.F = 0$$
$$P_1 = [I \,|0], \quad P_2 = [[e']_x F \,|e']$$

Refining the fundamental matrix using the Levenberg Marquardt algorithm

Once the fundamental matrix and the projection matrices in the canonical form are calculated they can be refined using the LM algorithm. In the following experiment I have done that by refining the matrix P_2.
The LM algorithm combines both the gradient descent and gauss newton methods. In the LM algorithm the step size in each iteration is given by the formula

$$\vec{\delta}_p = \left(J_{\vec{f}}^T J_{\vec{f}} + \mu I\right)^{-1} J_{\vec{f}}^T \vec{\epsilon}(\overrightarrow{p_k})$$

, $\vec{\epsilon}(\overrightarrow{p_k}) = \|X'_{actual} - F(P_k)\|$ is the error function, where F(P(k)) is $PX_{world\ coordinate}$.
The 3D world coordinates are calculated by triangulating the 2D coordinates x and x'. For each x and x' correspondence pair the following matrix is created

$$A = \begin{bmatrix} xP_3^T - P_1^T \\ yP_3^T - P_2^T \\ x'P_3'^T - P_1'^T \\ y'P_3'^T - P_2'^T \end{bmatrix}$$

3D world coordinate is given by the null vector of A. The null vector can be calculated conducting the SVD of A and taken the eigenvector corresponding to the smallest eigen value.

The cost function for the LM algorithm is given by, $D_2 = \Sigma||x - PX_{world\ coordinate}||^2 + ||x' - PX'_{world\ coordinate}||^2$

Once the refined P_2 matrix is calculated the refined Fundamental Matrix F can be calculated using as $[e']_x$ times the first three columns of P_2.

In the results section the LM algorithm is seen to optimize the left and right rectified image.

In the following experiment I have implemented the LM algorithm to optimize the projection matrix using the scipy.optimize.root library. In order to implement the library, I had to create a function that would calculate the error as well as the Jacobian matrix which was then used as an input to the library.

Namrata Vivek Raghavan
nrgahav@purdue.edu

Estimation of Homography matrices to generate the rectified image

Next the homography matrices H and H' need to be calculated to send e and e' to infinity.

1. *The following is the procedure to calculate the homography matrix for the right image*
   i. First, the angle to the epipole with respect to the positive x axis is calculated using the following formula

$$\theta = tan^{-1}(-1 \frac{\left(\frac{height}{2}\right) - e'(y)}{\left(\frac{width}{2}\right) - e'(x)})$$

   ii. Next the matrix G is calculated

$$G = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -\frac{1}{f} & 0 & 1 \end{bmatrix}$$

$$where\ f = \cos\theta\left(\left(\frac{width}{2}\right) - e'(x)\right) - sin\theta(\left(\frac{height}{2}\right) - e'(y))$$

   iii. The rotation matrix R is calculated

$$R = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

   iv. The translation matrix T is calculated

$$T = \begin{bmatrix} 0 & 0 & -\left(\frac{width}{2}\right) \\ 0 & 1 & -\left(\frac{height}{2}\right) \\ 1 & 0 & 1 \end{bmatrix}$$

   v. Next the image center is translated with respect to H2 and next translation matrix T2 is calculated. The final H' matrix is calculated using the following formula

$$H' = T_2GRT$$

The right epipole is given by $e' = [f\ 0\ 1]^T$

   vi. The homography H' can be applied to the right image to get the right rectified image

2. *The following is the procedure to calculate the homography matrix for the left image*
   i. First calculate the matrix M, given by $P'P^+$
   ii. Next calculate H_0, given by H_2M
   iii. Next, the matrix HA by minimizing the following sum of square

$$\Sigma(ax_i + by_i + c - x'_i)^2$$

$$H_A = \begin{bmatrix} a & b & c \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

   iv. The final H matrix is calculated using the following formula

$$H = H_AH_0$$

   v. The homography H can be applied to the left image to get the left rectified image

Namrata Vivek Raghavan
nrgahav@purdue.edu

II. Interest Point Detection

SIFT feature detector

Once the rectified images have been obtained then the SIFT is used to extract the features and the corresponding descriptors from the rectified images. In this task I have used the OpenCV inbuilt SIFT function to calculate the key points and their respective descriptor values. The correspondences between the two key points are then established by calculating the minimum sum of squared differences

SSD : Sum of Squared Differences

SSD is one method for establishing correspondence between the interest points detected in the left and right rectified image. In this method we create a (M+1) x (M+1) matrix around the selected corner points. Once the matrix is created SSD is calculate as,

$$SSD = \Sigma_i \Sigma_j |f_1(i,j) - f_2(i,j)|^2$$

, where f_1 is the window around the point in the 1st image anf_2 is the window around the point in the 2nd image. For each point in the first image find the corresponding minimum SSD value in the second image. Once the minimum SSD values have been calculated for each of the key points the we then sort the correspondences based on the ascending order of their SSD values. I have chosen the 40 best interest points between the two rectified images to calculate their corresponding world coordinate values.

Selection on interest points on the object of interest

The following is the algorithm to select only those interest points that lie on the object of the interest:

1. Manually detect the center of the object of interest and the corners of the object of interest in the rectified images.
2. Calculate the maximum distance between the center and the corners of the object of interest.
3. Once the maximum distance is calculated, only those correspondence that lie within the maximum distance from the center of the object of interest are selected.

III. Projective Reconstruction

The following is the procedure for 3D projective reconstruction of the 2D (x, x') correspondence found between the left and right image:

1. Using the correspondence detected using the SIFT feature detectors, estimate the fundamental matrix and canonical forms of the projection matrices.
2. Refine the P_2 matrix using the LM algorithm refined above and calculate the refined fundamental matrix and canonical forms of the refined projection matrices.
3. The 3D world coordinates are calculated by triangulating the 2D coordinates x and x'. For each x and x' correspondence pair the following matrix is created

Namrata Vivek Raghavan
nrgahav@purdue.edu

$$A = \begin{bmatrix} xP_3^T - P_1^T \\ yP_3^T - P_2^T \\ x'P_3'^T - P_1'^T \\ y'P_3'^T - P_2'^T \end{bmatrix}$$

3D world coordinate is given by the null vector of A. The null vector can be calculated conducting the SVD of A and taken the eigenvector corresponding to the smallest eigen value.

IV. 3D Visual Inspection

Once the world coordinates of the interest point between the images are calculates. The world coordinates of the corners points of the object of interest are calculated using the procedure above. These points are then plotted using scatter plot feature in matplotlib in python. The interest points are plotted in blue and the corner points are plotted in green. The corners are connected to create the outline of the 3D reconstruction using the plot feature of matplotlib in python

V. Results

Figure 1: Input left and right input image

Namrata Vivek Raghavan
nrgahav@purdue.edu

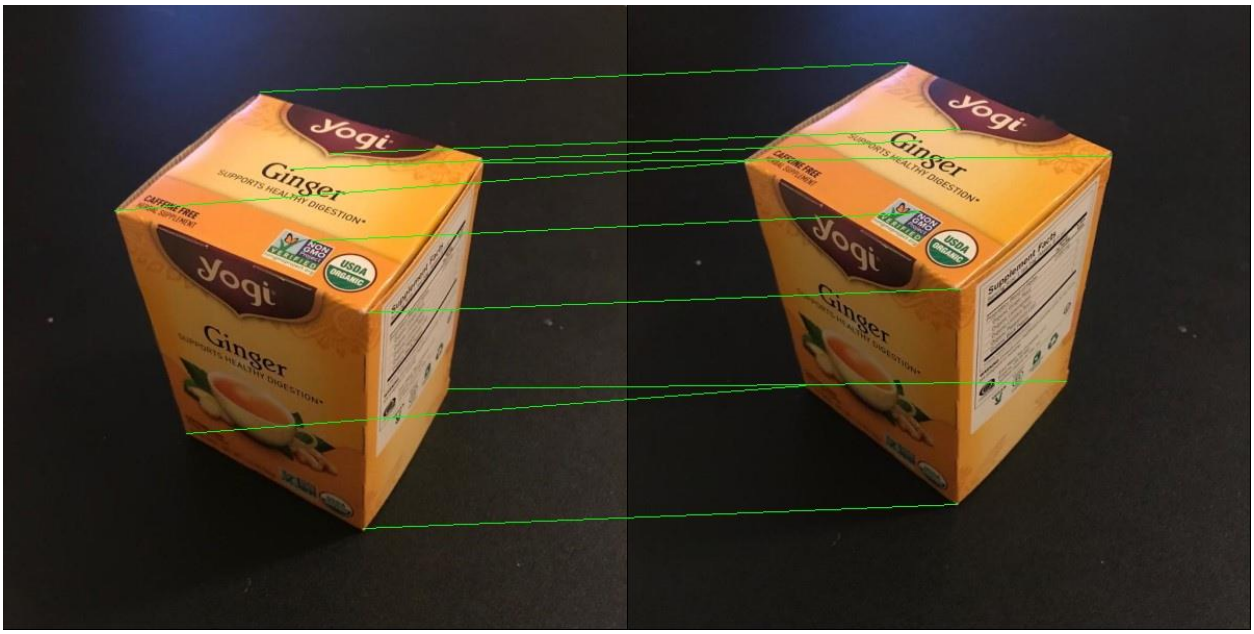

Figure 2: Manually selected initial correspondence points between the two input images

Figure 3: Right rectified input image before applying the LM algorithm

Namrata Vivek Raghavan
nrgahav@purdue.edu

Figure 4: Left rectified input image before applying the LM algorithm

Figure 5: Right rectified input image after applying the LM algorithm

Namrata Vivek Raghavan
nrgahav@purdue.edu

Figure 6: Left rectified input image after applying the LM algorithm

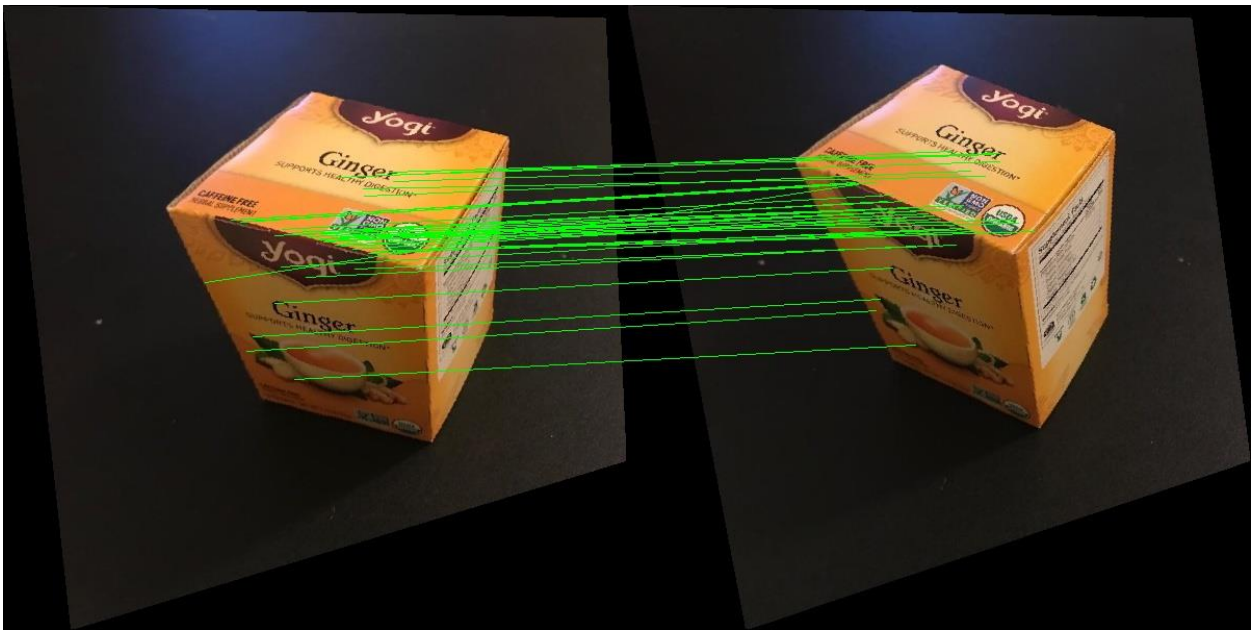

Figure 7: SIFT correspondence points between right and left rectified images

Namrata Vivek Raghavan
nrgahav@purdue.edu

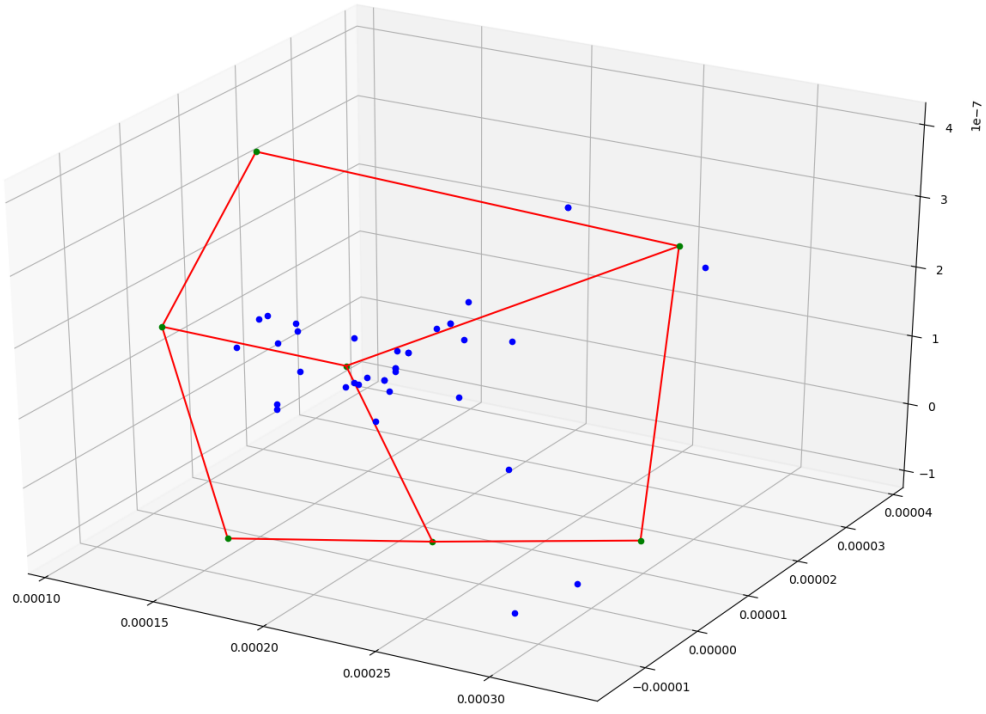

Figure 8: 3D projection of object of interest

Namrata Vivek Raghavan
nrgahav@purdue.edu

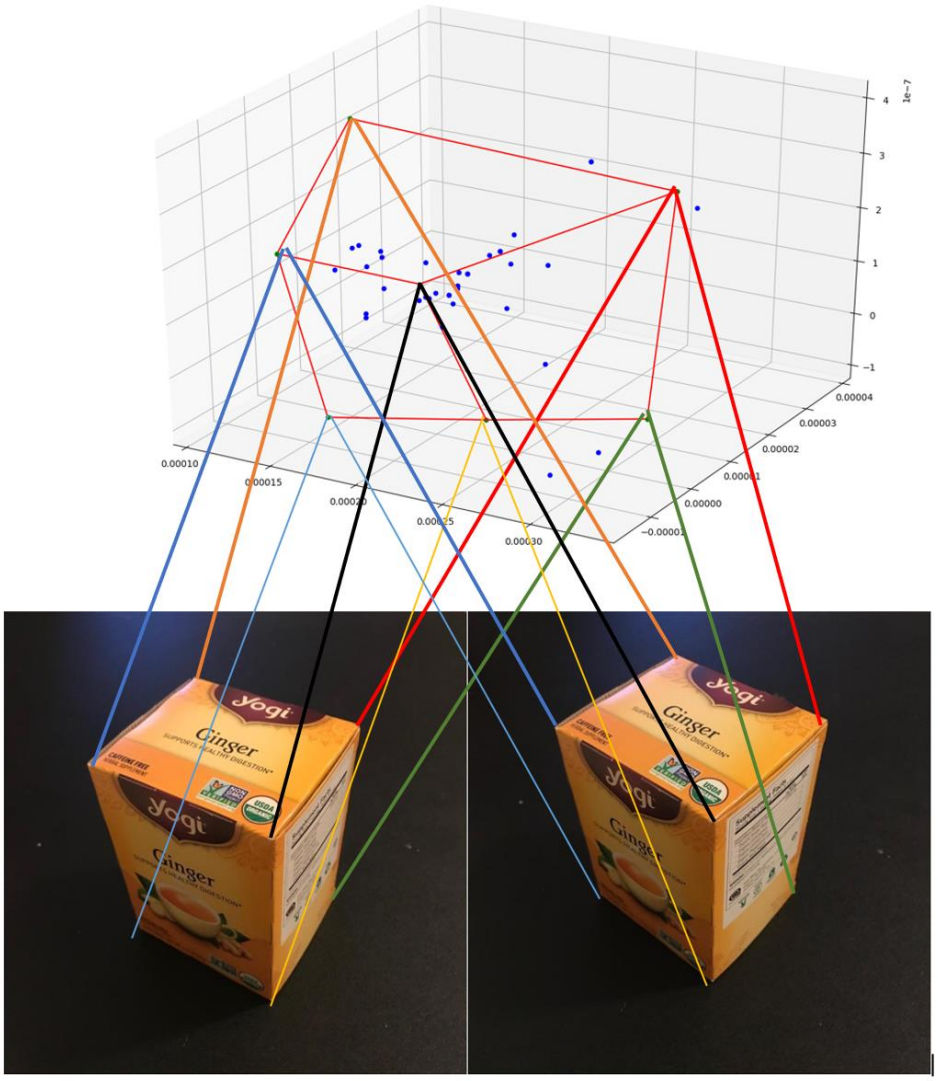

Figure 9: 3D projection of object of interest with markers to input images

Namrata Vivek Raghavan
nrgahav@purdue.edu

```
F before LM optimization
[[-1.02987628e-21 -9.50042125e-21  5.28237754e-19]
 [-1.24275736e-06 -1.32790559e-07  7.17729433e-03]
 [-6.76139882e-04 -6.59838095e-03  3.25510503e-01]]
P1 before LM optimization
[[1. 0. 0. 0.]
 [0. 1. 0. 0.]
 [0. 0. 1. 0.]]
P2 before LM optimization
[[-3.88433723e-03 -3.79188506e-02  1.86343725e+00 -6.94301620e+17]
 [-4.69445015e+14 -4.58126658e+15  2.26002469e+17  5.74671024e+00]
 [ 8.62848449e+11  9.21967003e+10 -4.98320708e+15  1.00000000e+00]]
left epipole before LM optimization
[ 5.83390322e+03 -5.48471535e+02  1.00000000e+00]
right epipole before LM optimization
[-6.94301620e+17  5.74671024e+00  1.00000000e+00]
left homography before LM optimization
[[-9.15037977e-01 -1.35160081e+00 -1.82366464e+03]
 [-9.21206947e-01 -8.37062519e+00  2.83893362e+02]
 [-3.04642935e-03 -2.98630153e-04 -7.34200308e+00]]
right homography before LM optimization
[[ 1.15169102e+00  1.75544208e-01 -9.81705697e+01]
 [ 1.24261728e-02  1.01344372e+00 -7.76096885e+00]
 [ 5.43696137e-04  8.28717995e-05  8.12029619e-01]]
```

Figure 10: Parameter values before LM optimization

Namrata Vivek Raghavan
nrgahav@purdue.edu

```
F after LM optimization
[[ 7.21203368e-15  7.30234796e-14 -2.00953467e-12]
 [ 2.64474482e+00  2.66221500e-01 -1.55418168e+04]
 [ 1.48094760e+03  1.43197391e+04 -7.10412828e+05]]
P1 after LM optimization
[[1. 0. 0. 0.]
 [0. 1. 0. 0.]
 [0. 0. 1. 0.]]
P2 after LM optimization
[[-3.01576699e+04 -2.91578505e+05  1.44809535e+07 -1.96040742e+17]
 [ 2.90326066e+20  2.80725228e+21 -1.39269858e+23 -2.03619798e+01]
 [-5.18477737e+17 -5.21902606e+16  3.04682931e+21  1.00000000e+00]]
left epipole after LM optimization
[ 5.93326344e+03 -5.64007442e+02  1.00000000e+00]
right epipole after LM optimization
[-1.96040742e+17 -2.03619798e+01  1.00000000e+00]
left homography after LM optimization
[[ 1.99342713e+06  2.75627857e+06  3.49730512e+09]
 [ 1.95276109e+06  1.65097801e+07 -6.16545539e+08]
 [ 6.66972954e+03  7.41769886e+02  1.41839396e+07]]
right homography after LM optimization
[[ 1.16254032e+00  1.81825620e-01 -1.03309780e+02]
 [ 2.00263318e-02  1.01528935e+00 -1.05947060e+01]
 [ 5.81838187e-04  9.10016516e-05  7.98148049e-01]]
```

Figure 11: Parameter values after LM optimization