3D Reconstruction
Project 03 - Report

Usama Munir Sheikh, Yaser Khalil
ECSE 6650: Computer Vision with Dr. Qiang Ji, RPI
Troy, New York
sheiku@rpi.edu

***Abstract/Goals*— The goal of this project was to perform 3D reconstruction of a human face from two images taken from different viewpoints. We start by calibrating the left and right cameras in the stereo configuration using a set of checkerboard calibration images and move on to find the relative orientation and translation between those cameras. We use this knowledge to construct the fundamental matrix. We also use the famous 8-point method to construct the fundamental matrix and compare the two results. Furthermore, we perform image rectification and construct epipolar lines based on our results. Finally, 3D reconstruction of the face is performed and the SSD correlation technique is used to identify corresponding points between the rectified images and display a disparity map.**

Keywords—Camera Calibration, 3D Reconstruction, eight-point method, fundamental matrix, image rectification, epipolar lines, SSD correlation, Disparity Map

I. Introduction

3D reconstruction is a very important topic in computer vision. It has many applications, ranging from robotics to human computer interaction to exploration. In this project we use passive stereo to 3D reconstruct a human face from two images taken from a left and right camera respectively.

II. Theory and Methods

A. *Camera Calibration*

There are two methods for doing camera calibration. The one used for this project is the conventional method where a single image is required and a calibration pattern that consists of 2D and 3D data is used to compute camera parameters [1]. The second method is called camera self-calibration which uses multiple views but only requires 2D image data[1].

Linear Solution I – Least Squares Method (LSQ):

In order to do camera calibration to determine the intrinsic parameters c_0, r_0, f, s_x, s_y. First, we have to define a prospective projection model that governs the relationship between the 2D image data and 3D position data set. The governing equation is

$$\lambda \begin{bmatrix} c_i \\ r_i \\ 1 \end{bmatrix}_{3x1} = P_{full_{3x4}} \begin{bmatrix} M_i \\ 1 \end{bmatrix}_{4x1} \quad (1)$$

Where i denotes the ith points, c_i and r_i are the column and row coordinates respectively corresponding to the 2D image, $M_{i_{3x1}}$is 3x1 column vector consists of $x\ y\ and\ z$ the 3D position of the point, and P_{full} is the full perspective projection matrix described by

$$P_{full} = \begin{bmatrix} p_1^T & p_{14} \\ p_2^T & p_{24} \\ p_3^T & p_{34} \end{bmatrix}_{3x4} \quad (2)$$

Where p_1, p_2, p_3 are 3x1 column vectors where as p_{14}, p_{24}, p_{34} are scalar values.

Combining equations 1 and 2 and expanding will result in the following three equations

$$\lambda c_i = M_i^T\, p_1 + p_{14} \quad (3)$$

$$\lambda r_i = M_i^T\, p_2 + p_{24} \quad (4)$$

$$\lambda = M_i^T + p_{34} \quad (5)$$

Substituting λ from equation 5 into equations 3 and 4 and simplifying them furthermore will result

$$M_i^T\, p1 + p_{14} - c_i M_i^T\, p_3 - c_i p_{34} = 0 \quad (6)$$

$$M_i^T\, p2 + p_{24} - r_i M_i^T\, p_3 - r_i p_{34} = 0 \quad (7)$$

These two equations can be reorganized for N points into a linear equation of the following form

$$A_{2Nx12}\, V_{12x1} = 0_{2Nx1} \quad (8)$$

Where A is a 2Nx12 matrix that depends on the 3D and 2D coordinates of the N points, and V is 12x1 vector of the projection matrix. A and V are described by

$$A = \begin{bmatrix} M_1^T & 1 & 0_{1x3} & 0 & -c_1 M_1^T & -c_1 \\ 0_{1x3} & 0 & M_1^T & 1 & -r_1 M_1^T & -r_1 \\ & & & \vdots & & \\ M_N^T & 1 & 0_{1x3} & 0 & -c_N M_N^T & -c_N \\ 0_{1x3} & 0 & M_N^T & 1 & -r_N M_N^T & -r_N \end{bmatrix}_{2Nx12} \quad (9)$$

And

$$V = \begin{bmatrix} p_1^T \\ p_{14} \\ p_2^T \\ p_{24} \\ p_3^T \\ p_{34} \end{bmatrix}_{12x1} \quad (10)$$

For this linear solution to work and solve for V from equation 8, the number of points N must be greater than six

$(N \geq 6)$. At least 6 independent points are required. Examples of dependent points are coplanar or points on circle...etc. In general A matrix has a rank of 11 and the solution V is the null vector of A[1]. However, in most cases A has rank of 12 due to error, noise, or distortion in the data. Then $||AV||^2$must be minimized in order to solve for V and to determine the null vector of A when its rank is 12. First, we will perform singular value decomposition on A

$$A_{mxn} = U_{mxm}\, D_{mxn}\, S_{nxn}^T \tag{11}$$

Denote V' as the null vector of A. It is the last column of the matrix S corresponding to the smallest eigenvalue in the diagonal D matrix. Then V is the null vector of A up to a scale factor.

$$V = \gamma V' \tag{12}$$

This scaling factor γ can be solved for using the fact that $||p_3||^2 = 1$. The vector p_3 corresponds to the 9th, 10th, and 11th elements of V. Then, the scaling factor is described by

$$\gamma = \sqrt{\left(\frac{1}{V'^2(9)+V'^2(10)+V'^2(11)}\right)} \tag{13}$$

Camera Parameters:

Once the vector V has been solved for, then the projection matrix for full perspective projection can be reconstructed using equation 10. The resulting equation is

$$p_{full} = \begin{bmatrix} p_1^T & p_{14} \\ p_2^T & p_{24} \\ p_3^T & p_{34} \end{bmatrix}_{3x4} = \begin{bmatrix} s_x f r_1^T + c_0 r_3^T & s_x f t_x + c_0 t_z \\ s_y f r_2^T + r_0 r_3^T & s_y f t_y + r_0 t_z \\ r_3^T & t_z \end{bmatrix}_{3x4} \tag{19}$$

Where r_1, r_2, and r_3 are the 1st ,2nd, and 3rd rows of the rotation matrix R respectively. t_x, t_y, and t_z are the components of the translation vector T. From equation 19, the intrinsic camera parameters c_0,r_0, $s_x f$,and $s_y f$ can be determined using the fact that r_1, r_2 and r_3 are the rows of the orthogonal rotation matrix R

$$c_0 = p_1^T p_3 \tag{20}$$
$$r_0 = p_2^T p_3 \tag{21}$$
$$s_x f = \sqrt{(p_1^T p_1 - c_o^2)} \tag{22}$$
$$s_y f = \sqrt{(p_2^T p_2 - r_o^2)} \tag{23}$$

The pose can be also estimated using equations 19 through 23, the resulting parameters are

$$p_{34} = t_z \tag{24}$$

$$r_3 = p_3 \tag{25}$$

$$t_x = \frac{p_{14} - c_0 t_z}{s_x f} \tag{26}$$

$$t_y = \frac{p_{24} - r_0 t_z}{s_y f} \tag{27}$$

$$r_1 = \frac{p_1 - c_0 r_3}{s_x f} \tag{28}$$

$$r_2 = \frac{p_2 - r_0 r_3}{s_y f} \tag{29}$$

Remember that $r_1, r_2, and\ r_3$ are row vector of the size 1x3. Similarly $p_1, p_2, and\ p_3$ are row vectors of the size 1x3.

B. *Relative Rotation & Translation*

Now that we have computed the left and right cameras orientations using camera calibration method described above. We can computed the relative orientation between the left and right cameras. The relative rotation of the right camera with respect to the left camera is governed by

$$R = R_l R_r^T \tag{30}$$

where R_l is the rotation matrix for the left camera, and R_r is the rotation matrix for the right camera. The relative translation of the right camera with respect to the left camera is described by

$$T = T_l - RT_r \tag{31}$$

where T_l is the left camera translation, R is the relative rotation in (30), and T_r is the right camera translation. P

C. *Computing Epipolar Lines & Fundamental Matrix*

Let's assumes there exist a point (P) in space captured by both cameras. Where P_l, P_r are the point in the left and right camera frames respectively. Then

$$P_l = RP_r + T \tag{32}$$

and

$$P_r = R^T(P_l - T) \tag{33}$$

where R,T are the relative rotation and translation as described in (30) and (31) respectively. The coplanar constraint will lead to

$$(T\ X\ P_l)^T(P_l - T) = 0 \tag{34}$$

From (33) we know that

$$P_l - T = RP_r \tag{35}$$

this yields

$$(T\ X\ P_l)^T RP_r = 0 \tag{36}$$

the cross operation is defined as a skew matrix, so

$$T^X = S \tag{37}$$

where S is the skew matrix defined as

$$S = \begin{bmatrix} 0 & -T_z & T_y \\ T_z & 0 & -T_x \\ -T_y & T_x & 0 \end{bmatrix} \tag{38}$$

combining equations (36-38) will give

$$P_l^T E P_r = 0 \tag{39}$$

where E is the essential matrix defined as

$$E = S^T R \tag{40}$$

Fundamental Matrix:

Lets denote U_l, U_r to be the pixels corrdinates of the point P in the left and right images respectively. Then the governing equations are

$$\lambda_l U_l = W_l P_l \tag{41}$$
$$\lambda_r U_r = W_r P_r \tag{42}$$

Where W_l, W_r are the left and right cameras intrinsic matrices respectively. Using (41) & (42), and substituting them in (39) yields to

$$U_l^T F U_r = 0 \quad (43)$$

where F is the fundamental matrix defined by

$$F = W_l^{-T} E W_r^{-1} \quad (44)$$

This fundamental matrix is very useful in the determination of the Epipolar lines. For example, given a point in the left image U_l then the equation for the epipolar line on the right image is given by $F^T U_l$. Similarly for a point U_r in the right image, the epipolar line equation on the left image given by is FU_r.

D. *Eight-Point Algorithm for Computing F*

Suppose that the fundamental matrix F is unknown and it cannot be determined because the camera parameters are not given. However, 2D image points in the left image and their corresponding points in the right image are given. These points are defined by $U_l = (c_{l_i}, r_{l_i}, 1)$ for the left image points, and $U_{r_i} = (c_{r_i}, r_{r_i}, 1)$ for the right image points.

Using the fundamental equation (44), we can rewrite it in the following form

$$A^{NX9} f^{9X1} = 0^{NX1} \quad (45)$$

Where N is the number of 2D image points, A is defined as

$$A = \begin{bmatrix} c_{l1}c_{r1} & c_{l1}r_{r1} & c_{l1} & r_{l1}c_{r1} & r_{l1}r_{r1} & r_{l1} & c_{r1} & r_{r1} & 1 \\ c_{l2}c_{r2} & c_{l2}r_{r2} & c_{l2} & r_{l2}c_{r2} & r_{l2}r_{r2} & r_{l2} & c_{r2} & r_{r2} & 1 \\ & & & & \vdots & & & & \\ c_{lN}c_{rN} & c_{lN}r_{rN} & c_{lN} & r_{lN}c_{rN} & r_{lN}r_{rN} & r_{lN} & c_{rN} & r_{rN} & 1 \end{bmatrix} \quad (46)$$

$$f = [f_{11} \; f_{12} \; f_{13} \; f_{21} \; f_{22} \; f_{23} \; f_{31} \; f_{32} \; f_{33}]^T \quad (47)$$

Using at least 8 non-coplanar points, the fundamental matrix can be solved for up to scale factor. It the rank of matrix A is 8 then the solution for f is the null vector of A. If rank(A)<8 then there exists many solutions equal to linear combination of all the null vectors of A. However, due to noise and error in the data, the *A* matrix might be a full rank (9). Since F is singular, then singular value decomposition can be preformed on F matrix which yields to

$$F = UDV^T \quad (48)$$

Next is to set the smallest element of the matrix D to zero which yields to D'. Using D' instead of D in equation (48) will yield to

$$F' = UD'V^T \quad (49)$$

Where F' is the closest singular matrix to F.

E. *Image Rectification*

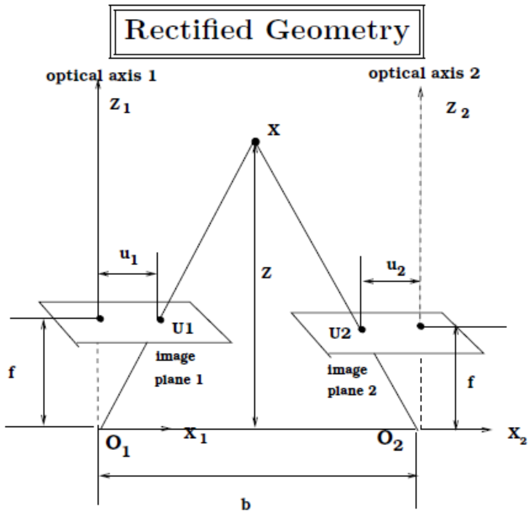

Figure 1: Rectified Geometry [1]

A rectified geometry is when the camera frames are parallel and coplanar. This makes the epipolar lines parallel. For this project, the cameras frames are neither parallel nor coplanar. Therefore, rectification process was used to make the epipolar lines parallel. Rectification is basically transforming and rotating the camera frames from the original pose to a new pose where they are parallel and coplanar. This will limit the search for the correspondence of 2D image points to a line search along the same row. So given, U_l, an Image point in the left image, the search will be along the same row of U_l But different columns in the right image. The process for rectification is

First, let's define a rotation matrix, R_l, that rotates the left image camera frame to be parallel to the baseline. This leads to the following

$$R_l \frac{T}{\|T\|} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad (50)$$

Where

$$\frac{T}{\|T\|} = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} \quad (51)$$

And

$$R = \begin{bmatrix} r_{l1} \\ r_{l2} \\ r_{l3} \end{bmatrix} \quad (52)$$

Where r_{li} Is a 1x3 row vectors of the rotation matrix. Now using 50-52, and the fact the rows of the rotation matrix are orthogonal will give

$$r_{l1} = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} \quad (53)$$

$$r_{l2} = \frac{1}{\sqrt{(t_x^2 + t_y^2)}} \begin{bmatrix} t_y \\ -t_x \\ 0 \end{bmatrix} \quad (54)$$

And

$$r_{l3} = r_{l1}\, X\, r_{l2} \quad (55)$$

Second, is to define the relationship that governs the correspondence between an image point U_l before rectification and $U_l{}'$ after rectification. The governing equation is

$$\lambda \begin{bmatrix} c_l' \\ r_l' \\ 1 \end{bmatrix} = W_l R_l W_l^{-1} \begin{bmatrix} c_l \\ r_l \\ 1 \end{bmatrix} \quad (56)$$

Similar analysis can be performed for the right image, the resulting relation for an image point before and after rectification is

$$\lambda \begin{bmatrix} c_r' \\ r_r' \\ 1 \end{bmatrix} = W_r R_r W_r^{-1} \begin{bmatrix} c_r \\ r_r \\ 1 \end{bmatrix} \quad (57)$$

Where R_r Is the right camera rotation and it is defined by

$$R_r = R_l R \quad (58)$$

The following step is to make the rectified image fits the original image by adjusting the scaling factors$(f s_x, f s_y, r_0, c_0)$ in the intrinsic matrix W. Note, usually the intrinsic camera parameters are the same. But if they are not the same, then an average of them must be used for the first W matrix in both equations (56) & (57), since this was the assumption made for the rectification process. This yields to

$$W = \frac{1}{2}(W_l + W_r) \quad (59)$$

And the rectification equations for left and right

$$\lambda \begin{bmatrix} c_l' \\ r_l' \\ 1 \end{bmatrix} = W R_l W_l^{-1} \begin{bmatrix} c_l \\ r_l \\ 1 \end{bmatrix} \quad (60)$$

$$\lambda \begin{bmatrix} c_r' \\ r_r' \\ 1 \end{bmatrix} = W R_r W_r^{-1} \begin{bmatrix} c_r \\ r_r \\ 1 \end{bmatrix} \quad (61)$$

Equations (60) & (61) describes the forward mapping which maps (c, r) to(c', r'). However, this mapping might create some holes and discontinuity in the new rectified image. Where some places in the rectified image will be black. The solution to this problem is to perform a backward mapping which maps (c', r') to (c, r). This yields to

$$\lambda \begin{bmatrix} c_l \\ r_l \\ 1 \end{bmatrix} = (W R_l W_l^{-1})^{-1} \begin{bmatrix} c_l' \\ r_l' \\ 1 \end{bmatrix} \quad (62)$$

$$\lambda \begin{bmatrix} c_r \\ r_r \\ 1 \end{bmatrix} = (W R_r W_r^{-1})^{-1} \begin{bmatrix} c_r' \\ r_r' \\ 1 \end{bmatrix} \quad (63)$$

Equations (62) & (63) can be used to preform backward mapping. In this process, where are looking for a point (c, r) from the original image that corresponds to the rectified point (c', r'). The intensity of the original points is then stored in the rectified (c', r'). Also, using the equations above for the fundamental matrix and equations (50-63), we can derive a formula for the rectification fundamental matrix to be $F_{rec} = W^{-1} R_l^{-T} E R_r^{-1} W^{-1}$.

F. Establishing Correspondence – Sum of Squared Difference (SSD)

The underlying idea for correlation methods is to establish correspondence between points based on intensity distribution of their neighborhood. These correlation methods have few assumptions, the assumptions are:

1- The points are visible from both viewpoints
2- Single light source
3- Corresponding image regions look similar

One of the correlation methods is the sum of squared differences (SSD). This method creates two windows. The first window in the left image and the second window in the right image. Then, we compute the squared difference between the elements of the windows. Finally, we sum all the elements. The formula for SSD is

$$SSD = (W_1 - W_2)^T (W_1 - W_2) \quad (64)$$

Where W_1 is a window about a point in the left image, and W_2 is a window about a point in the right image. For correspondence search, we will look for points in the right image that corresponds to points in the left image, vice versa also holds. The process correspondence search is described by the following steps:

1- Pick a point in the left image
2- Create a window W_1
3- Search along the epipolar line in the right image corresponding to the point in the left image. Denote the search window W_{2i}, where i means that the windows is moving.
   a) For rectified images the epipolar lines are parallel, therefore the search will be along the same row in the left image.
   b) For non-rectified images the epipolar lines are not parallel and they change by changing the point.
4- Compute the SSD using (64) for each window in the right image.
5- Find the minimum SSD value. This corresponds to the matching right image point.
6- Repeat the process to find correspondence for other points.

Disparity Map:

The image distance between the corresponding matched points found between the left and the right rectified images, using correlation, is called the disparity. Since they are present in the same row, the disparity is just the distance between the column coordinates of the matched points within the two images. By viewing the disparity between all corresponding points together, within a particular region, we can get a disparity map. For us, the correlation was contained within the facial area and therefore we have a found of a disparity map of this region only.

G. 3D Reconstruction

Figure Now that we have established correspondence between image points. The next step is to do a 3D-reconstruction of the object. For this project, since we know the camera parameters. Then, we will perform a full 3D reconstruction. Let's denote a point in the left image by (c_l, r_l), the corresponding point in the right image is denoted my (c_r, r_r). Then we can solve for the 3D coordinates (x, y, z) of the point relative to the left camera frame by the following

$$\lambda_l \begin{bmatrix} c_l \\ r_l \\ 1 \end{bmatrix} = \begin{bmatrix} w_{l1} \\ w_{l2} \\ w_{l3} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{65}$$

$$\lambda_l \begin{bmatrix} c_l \\ r_l \\ 1 \end{bmatrix} = \begin{bmatrix} p_{r1} & p_{14} \\ p_{r2} & p_{24} \\ p_{r3} & p_{34} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{66}$$

Where w_{li} is a 1x3 corresponds to the i^{th} row of the left camera intrinsic matrix, and p_{ri} is a 1x3 corresponds to the first 3 elements of the i^{th} row of the projection matrix P defind by

$$P = W_r[R \ \ T] \tag{67}$$

Where R, T are the relative rotation and translation of the right camera with respect to the left camera respectively. In equations (65) and (66), we notice that we have 5 unknowns $(\lambda_l, \lambda_r, x, y, z)$ with 6 equations. This means that we can create a least square problem to solve for the unknowns. Expanding (65) and (66) yields the following

$$\begin{gathered} \lambda_l c_l = w_{l1}[x \ y \ z]^T \\ \lambda_l r_l = w_{l2}[x \ y \ z]^T \\ \lambda_l = w_{l3}[x \ y \ z]^T \\ \lambda_r c_r = p_{r1}[x \ y \ z]^T + p_{14} \\ \lambda_r r_r = p_{r2}[x \ y \ z]^T + p_{24} \\ \lambda_r = p_{r3}[x \ y \ z]^T + p_{34} \end{gathered} \tag{68}$$

Rearranging these equations yields

$$\begin{bmatrix} w_{l1} & -c_l & 0 \\ w_{l2} & -r_l & 0 \\ w_{l3} & -1 & 0 \\ p_{r1} & 0 & -c_r \\ p_{r2} & 0 & -r_r \\ p_{r3} & 0 & -1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ \lambda_l \\ \lambda_r \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -p_{14} \\ -p_{24} \\ -p_{34} \end{bmatrix} \tag{69}$$

This forms a least square problem

$$AX = b \tag{70}$$

Where

$$A = \begin{bmatrix} w_{l1} & -c_l & 0 \\ w_{l2} & -r_l & 0 \\ w_{l3} & -1 & 0 \\ p_{r1} & 0 & -c_r \\ p_{r2} & 0 & -r_r \\ p_{r3} & 0 & -1 \end{bmatrix} \tag{71}$$

And

$$b = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -p_{14} \\ -p_{24} \\ -p_{34} \end{bmatrix} \tag{72}$$

And

$$X = \begin{bmatrix} x \\ y \\ z \\ \lambda_l \\ \lambda_r \end{bmatrix} \tag{73}$$

X can be solved for by the following least square formula,

$$X = (A^T A)^{-1} A^T b \tag{74}$$

III. Experimental Procedure & Results

A. *Camera Calibration*

Figure 2 shows the left and right checkerboard images and the corresponding 2D points that were used for camera calibration. The 3D coordinates and the 2D points were already measured and provided to us as data for calibration.

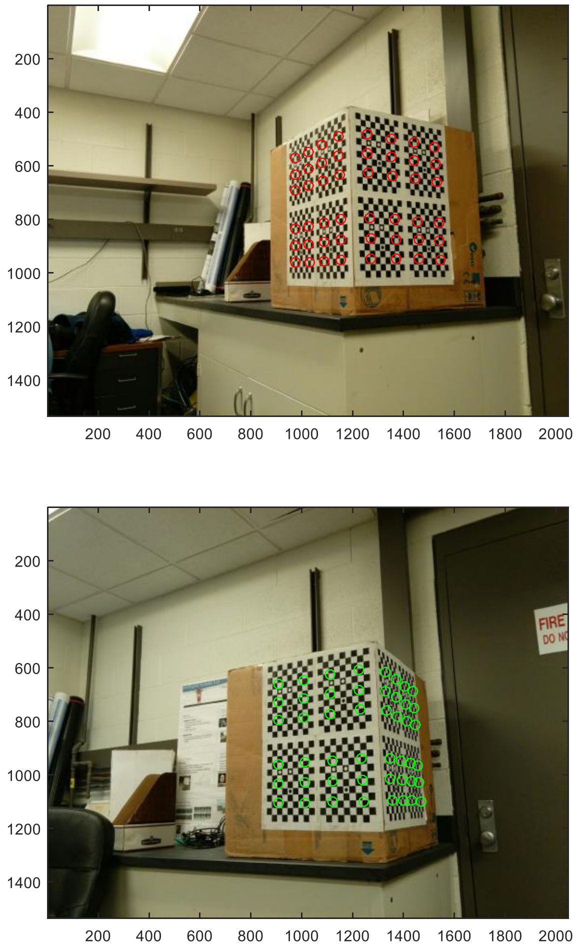

Figure 2. (Top) Left Calibration Image
(Bottom) Right Calibration Image

Using the procedure discussed earlier in Section II A, we get the following calibration results for the left and right cameras, respectively.

$$W_l = \begin{bmatrix} 1634.7 & 0 & 1055.4 \\ 0 & 1600.2 & 753.0 \\ 0 & 0 & 1 \end{bmatrix} \tag{75}$$

$$R_l = \begin{bmatrix} 0.5131 & -0.8582 & -0.0131 \\ -0.1366 & -0.0620 & -0.9887 \\ -0.8477 & -0.5091 & 0.1490 \end{bmatrix} \quad T_l = \begin{bmatrix} -9.9023 \\ 28.9974 \\ -122.61 \end{bmatrix} \tag{76}$$

$$W_r = \begin{bmatrix} 1656.1 & 0 & 1007.4 \\ 0 & 1637.2 & 719.0 \\ & 0 \quad 0 & 1 \end{bmatrix} \tag{77}$$

$$R_r = \begin{bmatrix} 0.8565 & -0.5160 & -0.0121 \\ -0.1289 & -0.1884 & -0. \\ -0.5001 & -0.8354 & 0.2279 \end{bmatrix} \quad T_r = \begin{bmatrix} -20.9003 \\ 16.4988 \\ -122.3548 \end{bmatrix} \tag{78}$$

Where W_l/W_r represent the intrinsic camera parameters and R_l/R_r and T_l/T_r provide the relative rotation and orientations of the cameras with respect to the 3D object (checkerboard).

Although it's the same camera being used for both the left and the right images, we see that the intrinsic parameters found are different for the right and left camera. This is purely due to noise in the corresponding points.

B. *Relative Rotation & Translation*

Using equations 30 and 31, we get the following relative rotation and translation between the two cameras:

$$R = \begin{bmatrix} 0.8825 & -0.1084 & -0.4574 \\ -0.0730 & 0.9919 & -0.1052 \\ -0.451 & 0.0601 & 0.8832 \end{bmatrix} \tag{79}$$

$$T_r = \begin{bmatrix} 62.7161 \\ -1.7623 \\ -25.2535 \end{bmatrix} \tag{80}$$

C. *Fundamental Matrix*

Using our results for the left and right intrinsic parameters and the relative rotation and translation between the two cameras, we compute the fundamental matrix through the method highlighted in section II C. We also compute the norm of the resulting fundamental matrix as this will be a useful comparison to our result of the same matrix using a different approach.

$$F = \begin{bmatrix} 0.0000 & 0.0000 & -0.0086 \\ 0.0000 & 0.0000 & -0.0434 \\ -0.0036 & 0.0306 & 9.9637 \end{bmatrix} \tag{81}$$

$$F_{norm} = \begin{bmatrix} 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0419 \\ -0.0004 & 0.0031 & 1.000 \end{bmatrix} \tag{82}$$

EIGHT-POINT Algorithm Results

We also compute the fundamental matrix using the eight-point algorithm described in section II D. The results are the following:

$$F_8 = \begin{bmatrix} 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0010 & -0.0007 \\ -0.0004 & -0.0028 & -1 \end{bmatrix} \tag{83}$$

Comparing this result, equation 82, to the result presented in equation 83, we see that the results are quite similar and the fundamental matrix found using the eight point method is very close to, however, not the same as the norm of the fundamental matrix we found using our calibration results earlier. Note, however, that if we don't normalize our earlier result, the two fundamental matrices found are quite different to one another.

D. *Image Rectification*

Following the procedure described in section II C, we perform image rectification on the two facial images. The fundamental matrix used for rectification is given below.

$$F_{rect} = \begin{bmatrix} 0 & 0 & 0.0089 \\ 0 & 0 & 0.0427 \\ 0.0034 & -0.0301 & -10.1609 \end{bmatrix} \tag{84}$$

The original images and the rectified left and right images are also shown below. The epipolar lines for both images have also been drawn to verify the results of our rectification.

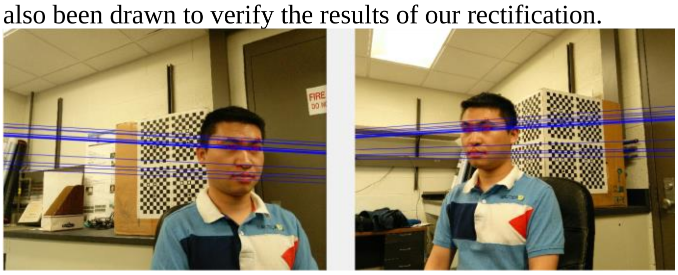

Figure 3. Original Images with Epipolar Lines

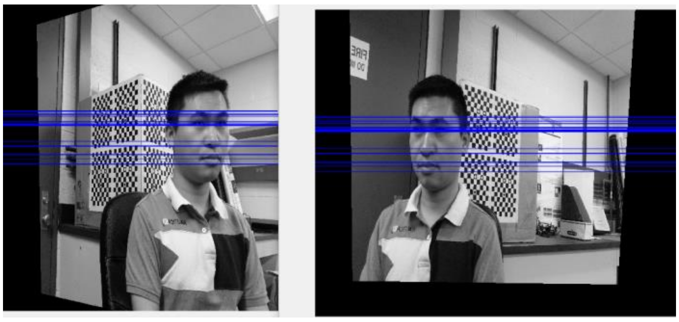

Figure 4. Rectified Images with Parallel Epipolar Lines

Note here that the rectified images are shifted here, by adding a small offset to the intrinsic parameters so that the image would fit into the same image size as the original image. If you want to look at the entirety of the images, the complete rectified images are also shown below.

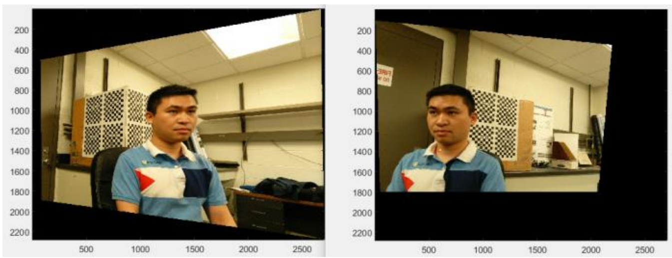

Figure 5. Complete Rectified Images

From figures 3 and 4, we can clearly see that all the epipolar lines drawn (passing through prominent facial points) are parallel to each other in the rectified images. And not only that, the rows of the left rectified image match up with the rows of right rectified image.

E. Establishing Correspondences & Disparity Map

The SSD correlation method described in section II F was used to establish correspondences in a small window of 7-by-7 pixels. For one window in the left rectified image (for a particular row), at each iteration of the method, the window was slid forward along the column (with the same row/scan line) on the right rectified image and the sum of squared differences with the corresponding window on the left image found. The point (pixel) that result in the lowest difference and therefore the greatest match was then established as the corresponding point. Then, the disparity for this matched pair was found and the window in the left image was moved forward to repeat the process for the next point. Some of the corresponding points in the left and right rectified image are shown below,

Figure 6. Correlated Corresponding Points

We can see that all of the correspondences are pretty close, however, they are not all completely precise. This is because the illumination direction for each of the two cameras is different and this affects the intensity values in both the right and the left image and therefore, the same 3D point (e.g. tip of the nose) has a different illumination (in rgb or grayscale) for the left image and a different one for the right image. Occlusions also affect the correlation process by hindering it. To counter act the effects of occlusion we can add a threshold for correlation. If the sum of squared difference is greater than this threshold, that corresponding point can be discarded. The result of partial occlusion on correlation can be seen from the green point at the side of the nose. But even still the points are quite close to each other.

That said, however, the points really are quite close, as you can see from the yellow and blue points at the side of the mouth and the lower lip respectively which are clearly well corresponded.

We've also created a disparity map for the correlation. After passing it through a median filter, the map is given by the following images. The second image is the zoomed-in version of the first.

Figure 7. Disparity Map. Shows a face-like image

The disparity maps show the disparity between the correlated points and as expected the shape of the map resembles the face.

F. 3D Reconstruction

The last part of the project was to use the face 2D image points provided to perform 3D reconstruction. The 3D reconstructed points in the frame of the left camera are shown below.

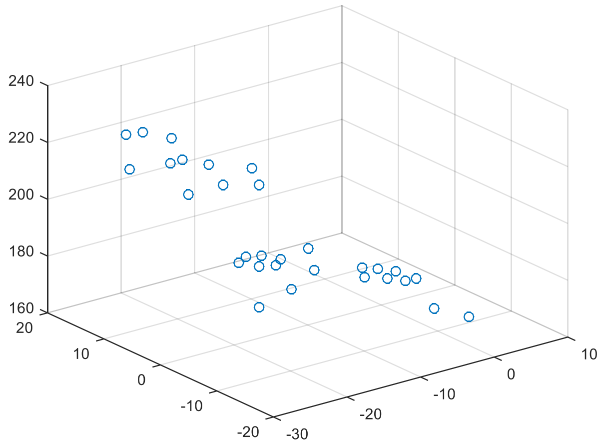

Figure 8. 3D Reconstructed Points

It is hard to make out a face from the above scatter plot but if we only look at the XY plane and do not consider the depth, we can make our the shape of the face as described by the 2D points provided:

Figure 9. Set of 2D points on left image

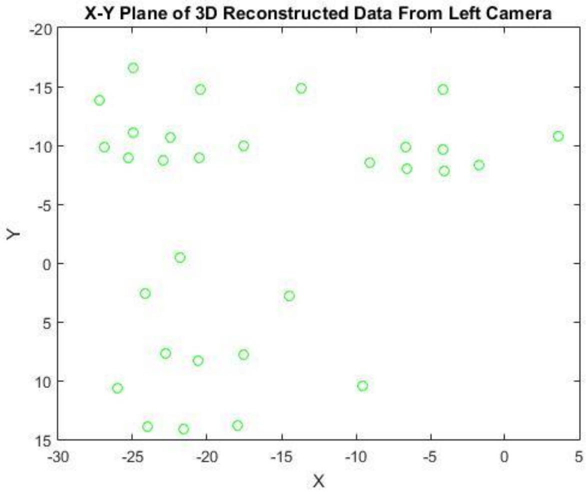

Figure 10. XY Plane of 3D points as seen from the left camera

From the results of the 3D reconstructed facial points, we can determine the width of the left and right eye and the mouth in mm. Points A and B are at the two corners of the eyes/mouth in each case and DistanceAB gives the width of the facial feature.

Left Eye:

$$Pt\ A = \begin{bmatrix} -9.037 \\ -8.50 \\ 187.20 \end{bmatrix} \qquad Pt\ B = \begin{bmatrix} -1.746 \\ -8.351 \\ 178.10 \end{bmatrix}$$

Distance AB = 11.6615 mm

Right Eye:

$$Pt\ A = \begin{bmatrix} -17.52 \\ -9.973 \\ 201.10 \end{bmatrix} \qquad Pt\ B = \begin{bmatrix} -26.90 \\ -9.867 \\ 202.90 \end{bmatrix}$$

Distance AB = 9.5517 mm

Mouth:

$$Pt\ A = \begin{bmatrix} -9.53 \\ 10.48 \\ 205.2 \end{bmatrix} \ Pt\ B = \begin{bmatrix} -25.97 \\ 10.63 \\ 216.30 \end{bmatrix}$$

Distance AB = 19.837 mm

We can see that the end-to-end distance or width of the left and the right eye are almost the same which should be the case for a symmetrical face. However, we do note that there is some error, this can be because the points provided aren't at the exact eye edges of either or one of the eyes.

IV. Summary & Conclusions

We achieved all the goals (undergraduate and graduate) as laid out by the project guidelines. We have achieved good results for camera calibration, finding relative orientation and translation, rectifying images and creating epipolar lines, using the eight-point algorithm, finding correspondences and performing 3D reconstruction.

Further work can be done to improve said algorithms and we can still play around with different parameters like change in window size for the SSD correlation to further analyze the effects of these parameters on our results.

One of the main problems encountered in this project was the great length of time it takes to run the code. This was reduced by using a faster computer but we could have also run our code in parallel to do the same or optimize the code further using some image processing techniques.

As far as contribution goes, both group mates worked on all parts of the project and consolidated their results with each other. We worked separately on each part and compared our results before moving on to the next part. A lot of bugs were found and solved because of this. Also, this way, we had a way to validate our results and stay on the right track.

Furthermore, to validate our results for rectification, we first, performed rectification on the calibration images. This helped us determine that there was something wrong with the initial data provided for the face. This issue was brought up in class by our group and a new set of data was provided.

References

[1] [1] Ji, Qiang. "ECSE 6650 Lecture Notes." N.p., n.d. Web. 19 Mar. 2016.

<https://www.ecse.rpi.edu/Homepages/qji/CV/ecse6650lecture notes.html>