Multiview Geometry Lab 3 Report

Delivered by:
Muhammad Faran Akram (u1999088)
Solomon Chibuzo Nwafor (u1999124)

Supervisor:
Professor Nuno Gracias

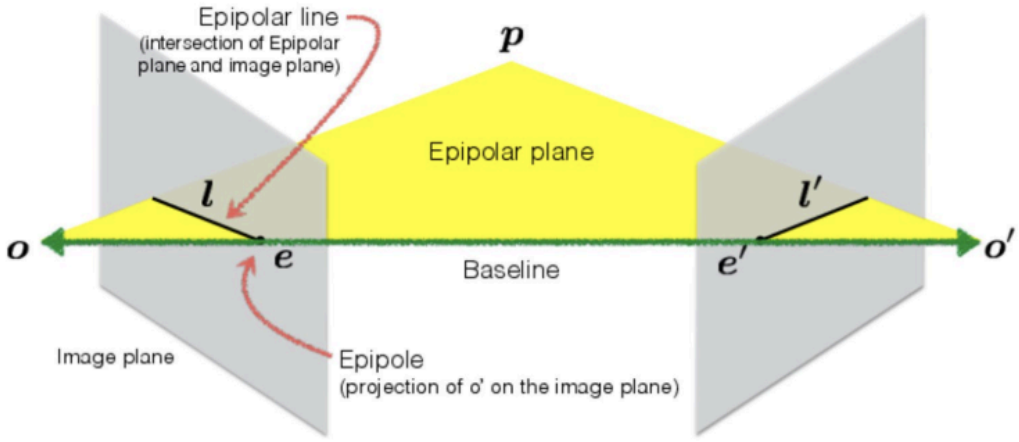
Date of Submission:

28/11/2024

Introduction:

In computer vision, the relationship between features of two views of the same 3D scene is represented by epipolar geometry. Using epipolar geometry, We can not only identify feature correspondences in stereo views of the same 3D scene but can also track them and even reconstruct the 3D scene.

Epipoles, Epipolar-Lines and Epipolar plane are the fundamental elements of epipolar geometry. Epipoles are the points where baseline, line passing through camera origin ***o*** and ***o'*** (as shown in Figure a.), intersect the image plane. Epipolar-Lines are the lines, ***l*** and ***l'***, on an image plane which passes through the epipoles. In other words, We can say It is the segment on the image plane where the epipolar plane intersects the image plane. The Epipolar plane is the plane defined by the 3D point ***p*** and camera centers ***o*** and ***o'***. The epipolar geometry simplifies the problem of finding correspondences.



*[1]**Figure a. Epipolar geometry setup***

The 3D point ***p***, projecting on the right image plane, can be anywhere on line ***p-o'***. This line is projected as the epipolar line ***l***, on the left image plane, passes through epipole ***e.*** Which means, the projection of the 3D point ***p***, must be on the epipolar line ***l***, on the left image plane. This constraint limits the search of correspondence of a 3D point from 2D to 1D. Hence, 3D point ***p***,observed through the right image plane, will be found on the epipolar line ***l*** on the left image plane and vice versa. This constraint is known as ***epipolar constraint.***

Fundamental matrix is a 3x3, rank deficient matrix with 7 degrees of freedom. This matrix ***F*** helps us find the correspondence of a feature between two stereo images.

$$F = \begin{bmatrix} 1 & b & \alpha+\beta \\ c & d & \alpha+\beta \\ e & f & \alpha+\beta \end{bmatrix}$$

$$x'^T F x = 0$$

The fundamental matrix encodes the intrinsic and extrinsic parameters. The rank deficiency property ensures that the fundamental matrix properly encapsulates the epipolar geometry of stereo images, enabling accurate and efficient correspondences. If the fundamental matrix ***F*** is not rank deficient,it won't accurately represent the epipolar geometry, the epipolar lines will not pass through the epipolar centre.Fundamental Matrix F can be estimated using ***8-point algorithm*** and then applying singular value decomposition (***SVD***), where F will be the last column of V and rank constraint can be applied by modifying a diagonal element of D as zero.

Epipolar geometry and the fundamental matrix plays a crucial role in various computer vision tasks. They are widely used in camera calibration to determine intrinsic and extrinsic camera parameters, and in stereo depth estimation to calculate the 3D structure of a scene from two images.

Implementation:

Results can be regenerated by running the ***"lab3_empty.m"*** file.

Step 1 & 2 : Parameter Initialization

Defined camera parameters:

Camera1	Camera2

au1 = 100; av1 = 120; uo1 = 128;
vo1 = 128;
imageSize = [256 256];
	au2 = 90; av2 = 110; uo2 = 128; vo2 = 128;
	ax = 0.1; by = pi/4; cz = 0.2;
	tx = -1000; ty = 190; tz = 230;

Step 3: Intrinsic parameter

We computed the intrinsic and projection matrices for camera 1 and camera 2. The intrinsic matrix of camera 1 was given while we computed the intrinsic parameter for camera 2, as shown in the code below:

code:

```
%The intrinsic matrix for camera 2
K2 = [au2 0 uo2; 0 av2 vo2; 0 0 1];% Intrinsics matrix for camera 2
eul = [ax by cz]; %euler angles for X, Y and Z coordinates
wR2c = eul2rotm(eul, 'XYZ' ); %matlab function for computing the euler angles
wt2c = [tx ty tz]'; %translatiion of camera 2, from the camera to the world
coordinate frame
```

Step 4: Analytical Fundamental Matrix

After computing the rotation and translation matrices, we computed the fundamental matrix by using the following formula:

$$F = K_2^{-T} . R^T . [t]_X . K_1^{-1}$$

where $[t]_X$ is the skew-symmetric matrix and K_2 are the intrinsic matrices of camera 1 and camera 2, respectively.

$$\begin{bmatrix} u_2 & v_2 & 1 \end{bmatrix} F \begin{bmatrix} u_1 \\ v_1 \\ 1 \end{bmatrix} = 0$$

Code:

```
%Buildinf the skew symmetric matrix
skew_tx =[0 ,-tz,ty;
   tz,0,-tx;
   -ty,tx,0];
%Computing the fundamental matrix
F = inv(K2)' * wR2c' * skew_tx * inv(K1);
%Normalization of the fundamental matrix
F = F./F(3,3);
```

Results:

Analytically obtained F:

```
[0.0000   0.0001   -0.0066
 0.0000  -0.0000    0.0130
-0.0096  -0.0173    1.0000]
```

Steps 5 & 6: Camera Points

The two camera points were provided in step 5, and we used this function *mvg_projectPointToImagePlane()* in step 6 to compute the projections on the image planes as ***cam1_p2d*** and ***cam2_p2d*** for camera 1 and camera 2, respectively.

Code:

```
cam1_p2d = mvg_projectPointToImagePlane(V,P1); %u1,v1
cam2_p2d = mvg_projectPointToImagePlane(V,P2);%u2,v2
```

Step 7: Points drawn on image planes

mvg_show_projected_points() draws the 2D projections on image planes, while mvg_compute_epipolar_geom_modif() function draws the epipolar lines

Code:

```
% Draw 2D projections on image planes
cam1_fig = mvg_show_projected_points(cam1_p2d(1:2,:),imageSize,'Projected
points on image plane 1');
cam2_fig = mvg_show_projected_points(cam2_p2d(1:2,:),imageSize,'Projected
points on image plane 2');
% % Draw epipolar lines
[~,~,c1_l_coeff,c2_l_coeff] =
mvg_compute_epipolar_geom_modif(cam1_p2d,cam2_p2d,F);
[cam1_fig,cam2_fig] = mvg_show_epipolar_lines(cam1_fig, cam2_fig,
c1_l_coeff,c2_l_coeff, [-400,1;300,400],'b');
```

Results:

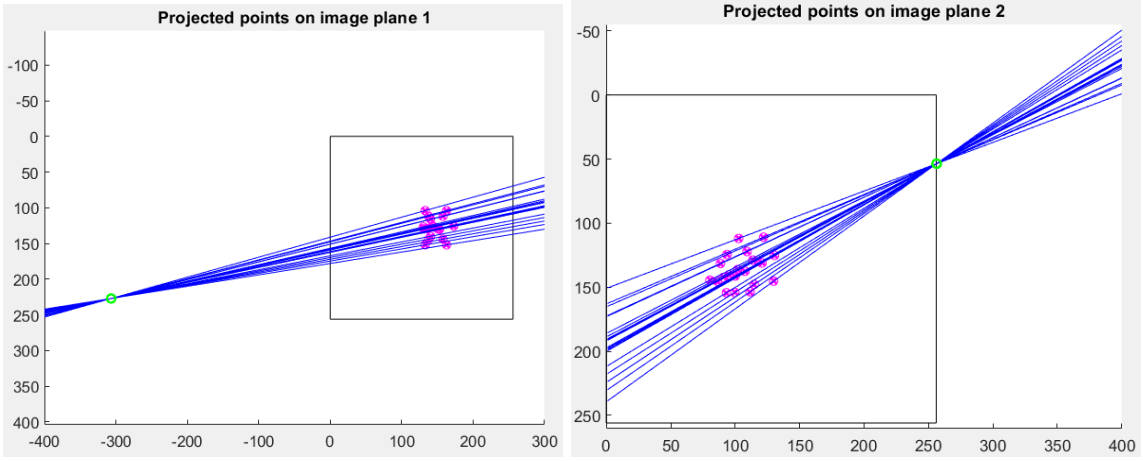


Figure 1: Projected points on image planes I1 &I2

Step 8: Estimated Fundamental Matrix (8-Point Algorithm)

Implemented the *u_matrix(x1, x2)* function to find the fundamental matrix using the 8-point method using 2D points found in step 6.

Code:

```
%% Making function to calculate the 8-point method
function u_n = u_matrix(x1, x2)
num_1 = length(x1);
u_n = [x2(1,:)'.*x1(1,:)' x2(1,:)'.*x1(2,:)' x2(1,:)' x2(2,:)'.*x1(1,:)'
x2(2,:)'.*x1(2,:)' x2(2,:)' x1(1,:)' x1(2,:)' ones(num_1,1) ];
end
%8-Point method
U_n = u_matrix(cam1_p2d, cam2_p2d);%8-point computation for the two cameras
[U, S, V] = svd(U_n);
F_8 = reshape(V(:,9),3,3)'; %the last column of the V matrix which forms the
```

```
8-point fundamental matrix
% Enforce rank2 constraint
[U,D,V] = svd(F_8);
F_8 = U*diag([D(1,1) D(2,2) 0])*V'; %Enforcing the rank 2 constraint by
setting D(3,3) = 0
F_8 = F_8./F_8(3,3); %Normalizing the F_8 matrix
```

Results:

Estimated F-8 point:

$$\begin{bmatrix} 0.0000 & 0.0001 & -0.0066 \\ 0.0000 & -0.0000 & 0.0130 \\ -0.0096 & -0.0173 & 1.0000 \end{bmatrix}$$

Step 9: Sum of Absolute Differences

In this step, we computed the sum of the absolute difference between F (analytical fundamental matrix) and F-8 (estimated 8-point fundamental matrix), which is very small (3.8761e-15).

Code:

```
difference1 = sum(abs(F_8-F),"all"); %calculating the sum of absoute
difference between F-8 and F
fprintf('Step 9:\n\tComparing F-8 point and F:\n');
disp(difference1);
```

Results:

Comparing F-8 (Estimated) matrix and F (Analytical) matrix:

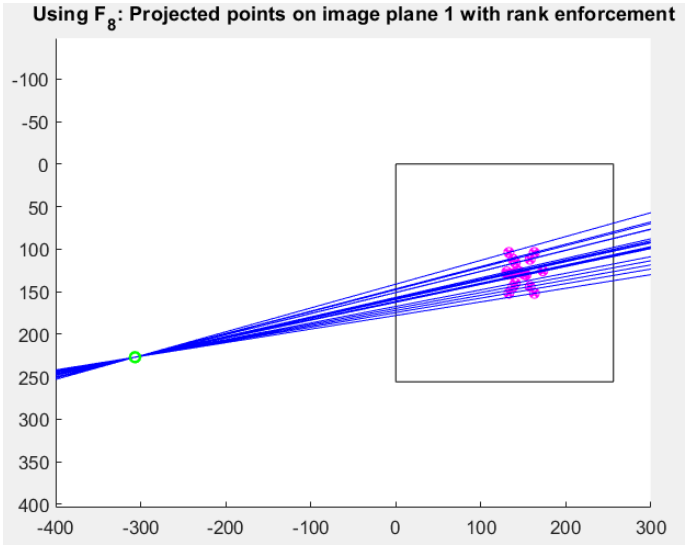
3.8761e-15

Step 10: Projection based on Estimated Fundamental Matrix

We implemented the fundamental matrix of the 8-point algorithm with rank 2 constraint enforcement as shown in Figure 2 and without rank 2 enforcement as shown in Figure 3, respectively. Since the sum of the absolute difference we computed in step 9 was too small, the performance of the 8-point algorithm and that of the analytical fundmental matrixx were the same. As the left epipole ***e*** lies on all epipolar lines in the left image, this means for any point in the right image, the epipolar line in the left image passes through ***e***. Mathematically, this relationship can be described as

$$p^T F e = 0.$$

Hence, ***e*** is the null space of ***F***. Similarly, ***e'*** is also the null space of ***F***.



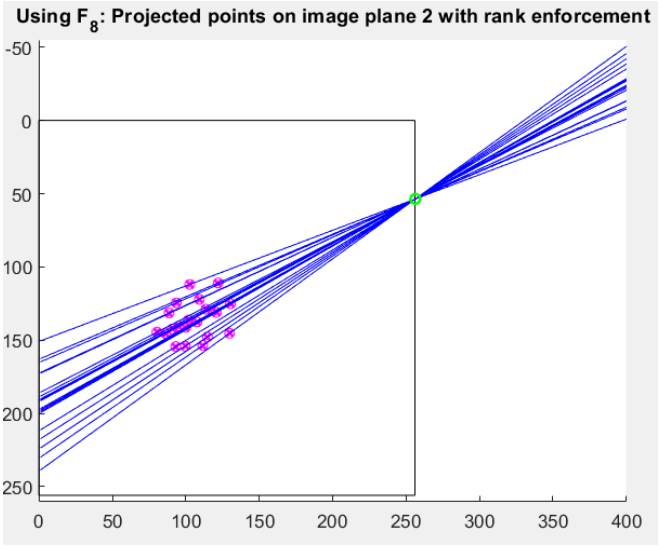
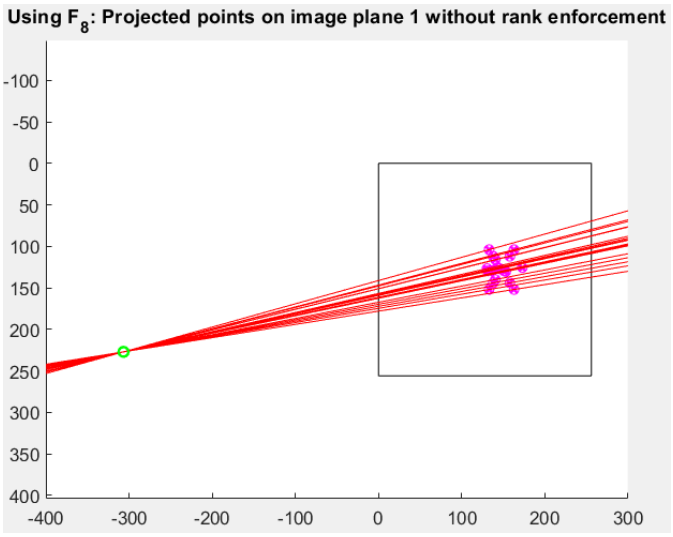


Figure 2: Results with rank enforced F matrix



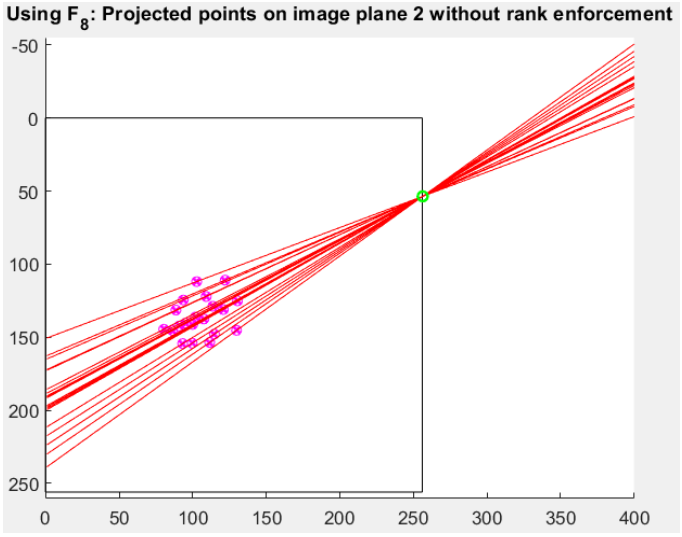


Figure 3: Results without rank enforcement

Step 11: Noisy Camera Points Generation

Since we are going to implement 95% noise to the points in steps 12 and 13, we created a function *add_gaussian_noise(projection_param, range)* to add gaussian noise to points as shown in the code below:

code:

```
function noisy_cam_p2d = add_gaussian_noise(projection_param, range)
std_dev = range/2;  % This ensures ±1 range for 95% of the points
gaussian = std_dev*randn(size(projection_param));
noisy_cam_p2d = projection_param + gaussian;
end
```

Step 12: [-1, 1] Noise Addition

In this step, we repeated step 8 to step 10 with the noisy 95% camera points we obtained in step 11. Figure 4 shows the projection of the noisy points on images 1 and 2 with rank 2 enforcement of the fundamental matrix. As we observed, due to the noise on the camera points, the epipolar lines did not pass through those noisy points, but they passed through the epipoles.

We also implemented the same noisy camera points with a fundamental matrix without rank 2 enforcement. As shown in figure 6, the epipolar lines did not pass through those noisy points and also did not pass through the epipoles. We show these epipolar lines not passing through the epipole in Figure 5.

code:

```
noisy_cam1_p2d= add_gaussian_noise(cam1_p2d, 1);
noisy_cam2_p2d= add_gaussian_noise(cam2_p2d, 1);
U_noise = u_matrix(noisy_cam1_p2d, noisy_cam2_p2d);
[U, S, noisy_V] = svd(U_noise);
F_8_noise = reshape(noisy_V(:,9),3,3)';
```

Results:

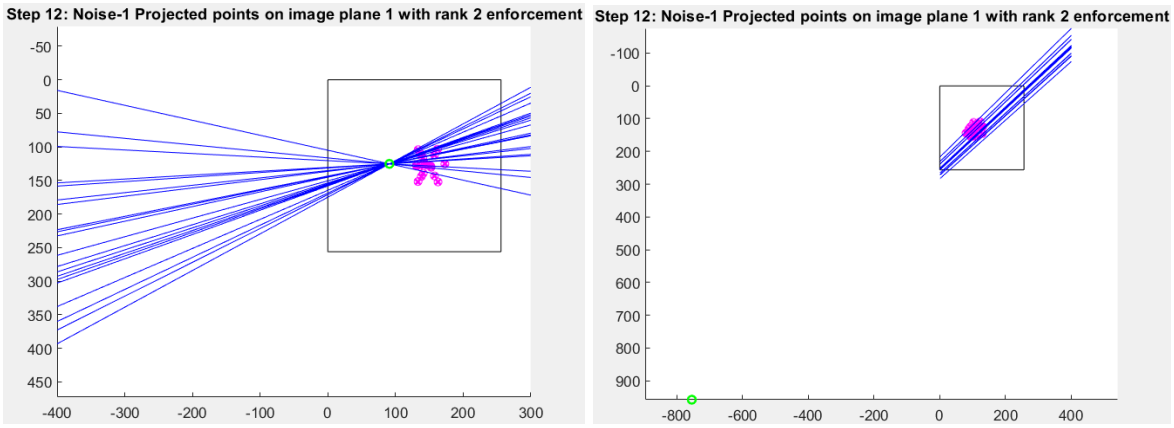


Figure 4: Noise-1 projected on image planes 1 and 2 with rank enforcement.

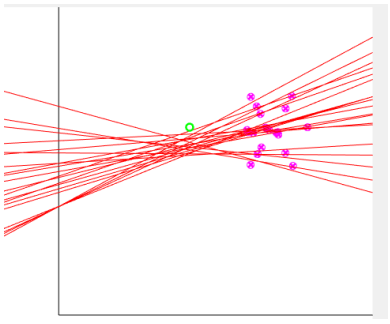


Figure 5: Zoomed-in version of Fig.6(a)

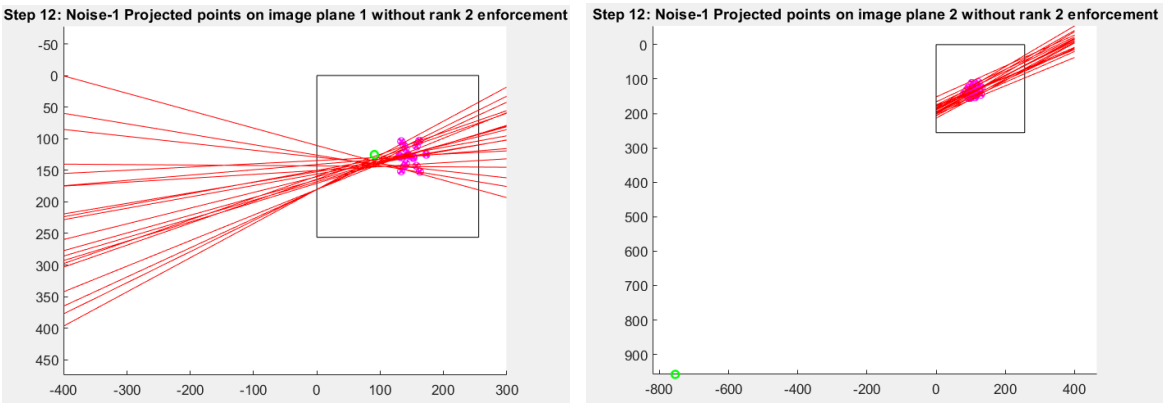


Figure 6: a) Noise-1 projected points on I1 without rank enforcement. b) Noise-2 projected points on I2 without rank enforcement.

Questions

- Are points on the epipolar lines?

Ans: No, as we have added noise in points, So these points are not exactly on the epipolar lines.

- Are the epipolar lines all converging on a single point?

Ans: When we enforce rank deficiency, the epipolar lines converge on a single point called the epipole. But without rank enforcement, the epipolar lines do not converge at epipoles.

- What condition should the F matrix fulfill so that all epipolar lines cross at the same point?

Ans: The F matrix should be enforced with a rank 2 constraint.

- Verify if all epipolar lines now cross at single same point on each image.

Ans: After imposing the rank 2 constraint on the F matrix, the epipolar lines crossed at the same point.

- Investigate how the location of both epipoles can be computed directly from the F matrix, using the SVD, and describe the Procedure.

Ans: To compute the locations of the epipoles using the fundamental matrix F:

1. Compute Singular Value Decomposition (SVD) and decompose F into three matrices UDV.
2. The epipole in the second image can be found as the last column of the matrix V, As this column corresponds to the smallest singular value and represents the null space of F.
3. Epipole in the first image is the last column of the matrix U, as this column represents the null space of F.
4. Normalize F to ensure that it is a homogeneous matrix.

Step 13: [-2, 2] Noise Addition

In this step, we increased 95% noise to the camera points in step 12 to (-2, 2) and implemented the same steps in step 12 without rank 2 constraints and with rank 2 enforcement. As shown in Figure 7, we enforced the rank 2 constraint, and the epipolar lines became more noisy and did not pass through the noisy camera points when compared to the previous noisy epipolar lines and still passed through the epipoles.

When we did not enforce the rank 2 constraint on the fundamental matrix, the epipolar lines became more noisy and hence did not pass through the epipoles and noisy camera points as shown in Figure 9. Figure 8 shows the zoomed-in version of the epipolar lines not passing through the noisy camera points and the epipoles when the rank 2 constraint was not enforced on the fundamental matrix.

Results:

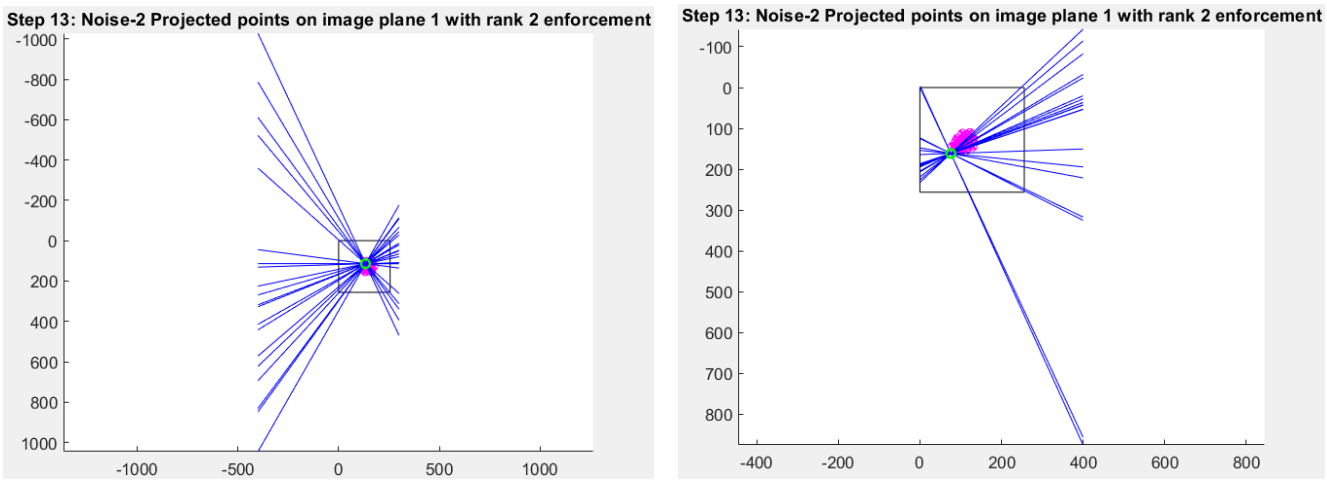


Figure 7: Noise-2 projected points on I1 & I2 with rank enforcement.

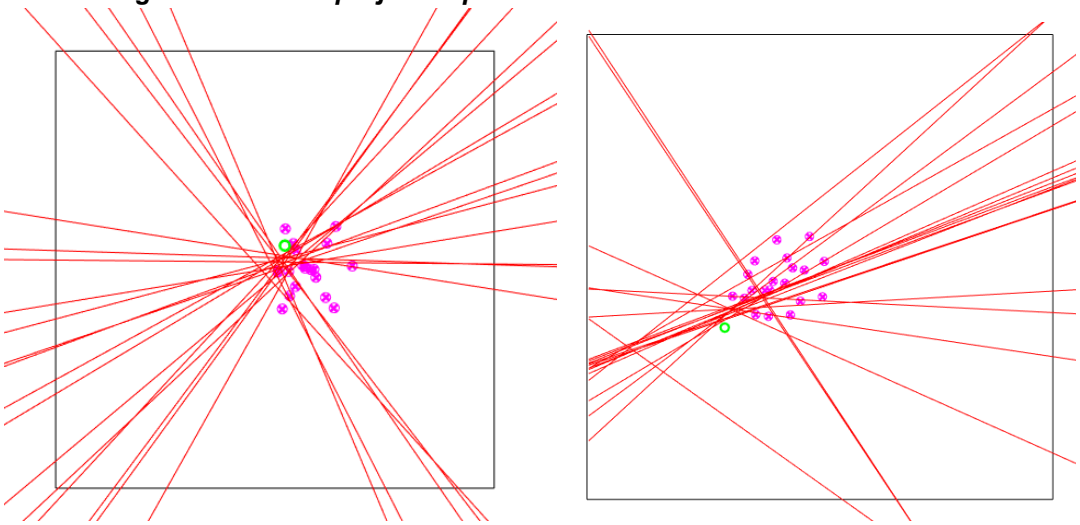


Figure 8: Zoomed-in version of Fig.8 (a) & (b)

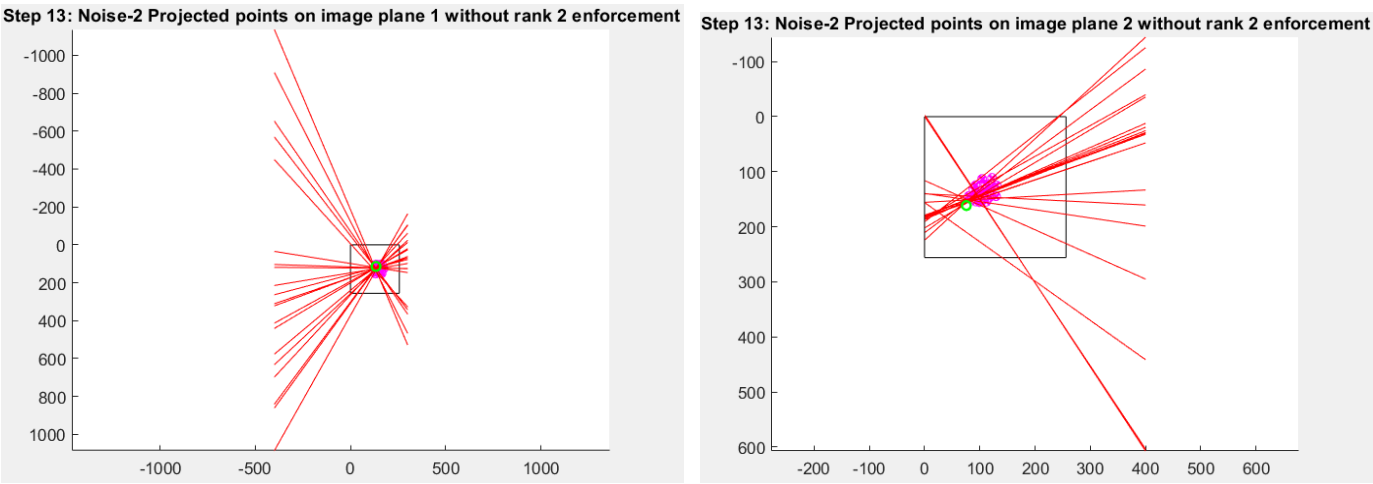


Figure 9: a) Noise-2 projected points on I1 without rank enforcement. b) Noise-2 projected points on I2 without rank enforcement.

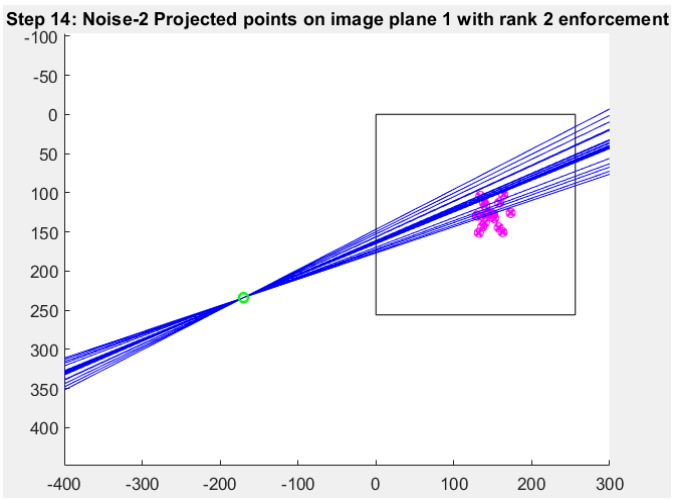
Step 14: Normalization of Noisy Camera Points

In this step, we applied normalization to the camera points in step 13, which have the most noise. First, we created a function as shown in the code section that normalizes the noisy camera points while getting the normalized fundamental matrix. We then denormalize this fundamental matrix. Lastly, we compared the performance of the fundamental matrix with rank 2 constraint enforcement and without rank enforcement.

As shown in Figure 10, the epipolar lines passed through the epipoles and some of the lines passed through the noisy camera points created in step 13. Also in Figure 12 shows the performance window of the fundamental matrix without rank 2 constraint enforcement. As we can see in Figure 11, the epipolar lines did not pass through the epioples.

Results:

```
% Apply normalization to noisy points in step 13
function [normalized_points, norm_matrix] = normalize(camera_2dp, camera_2d)
%center of points
center_point = sum(camera_2dp, 2)/size(camera_2dp, 2);
center_points = camera_2dp - (center_point*ones(1, size(camera_2dp, 2)));
%scale factor
scale_factor = sqrt(2)/ (sum(sqrt(sum(center_points.^2)))/size(camera_2dp,
2));
%norm matrix
norm_matrix = [scale_factor*eye(2), -scale_factor*center_point; 0 0 1];
normalized_points = norm_matrix*[camera_2d; ones(1, size(camera_2d, 2))];
end
```



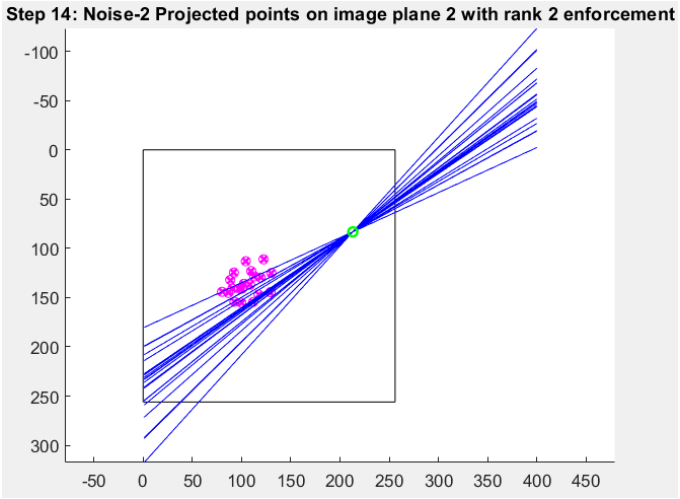


Figure 10: Noise-2 projected points on I1 & I2 with normalized F & rank enforcement.

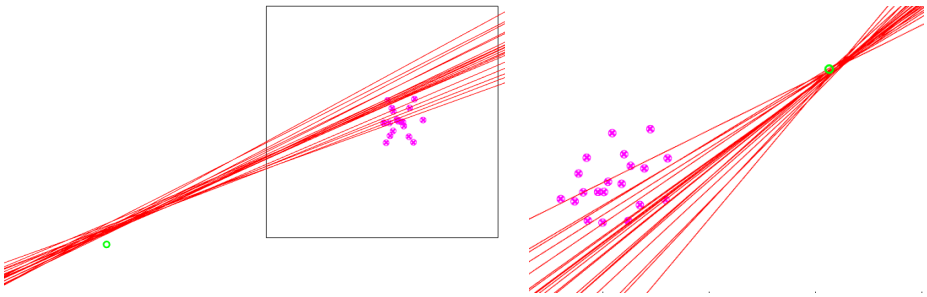


Figure 11: Zoomed-In versions of Fig. 12 (a) & (b)

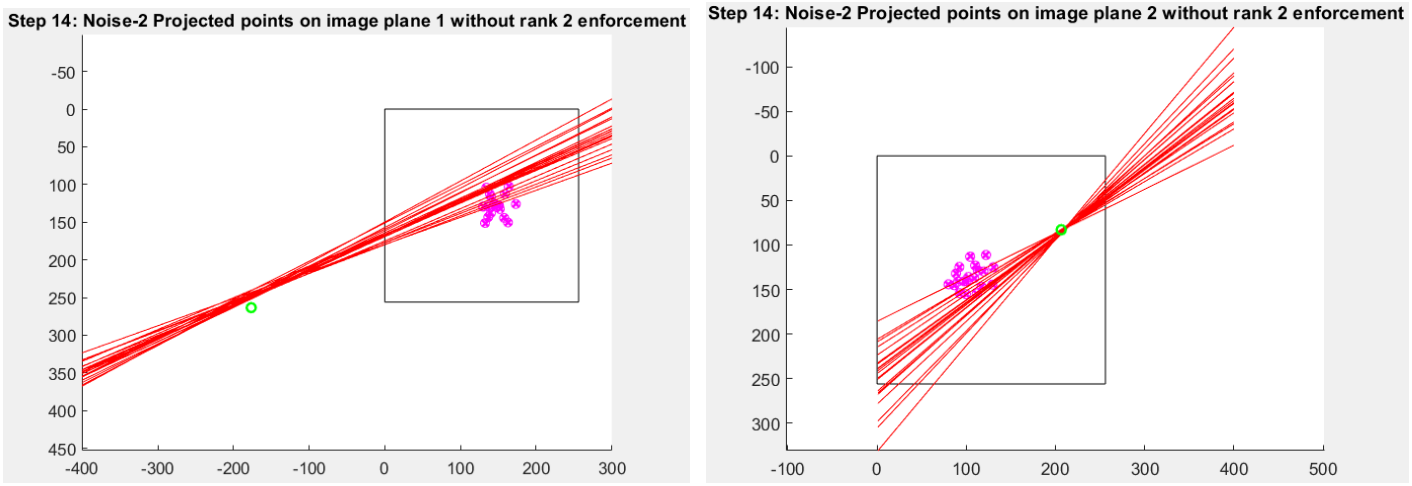


Figure 12: Noise-2 projected points on I1 & I2 with normalized F but without rank enforcement.

Are there improvements in terms of the sum of absolute differences (Step 9) and on the location of the epipolar lines (Step 10)?

Ans: The sum of absolute differences (SAD) increased when we normalized the fundamental matrix. Also, the epipolar lines improved, consistently passing through the epipoles, even though noise caused slight deviations from the noisy camera points.

References:

- *[1] Figure a.: Link*