Assignment 4 EEN020 Computer Vision

Erik Norlin

14 December 2023

Robust Epipolar Geometry and Two-View Reconstruction

Theoretical Exercise 1

We have the camera pair

$$P_1 = [R_1 \mid t_1], \text{ and } P_2 = [R_2 \mid t_2]$$

We transform the camera pair with

$$H = \begin{bmatrix} R_1^T & -R_1^T t_1 \\ \mathbf{0} & 1 \end{bmatrix}$$

we get that

$$P_1 H = [R_1 \mid t_1] \begin{bmatrix} R_1^T & -R_1^T t_1 \\ \mathbf{0} & 1 \end{bmatrix} = [I \mid \mathbf{0}]$$

$$P_2 H = [R_2 \mid t_2] \begin{bmatrix} R_1^T & -R_1^T t_1 \\ \mathbf{0} & 1 \end{bmatrix} = [R_2 R_1^T \mid -R_2 R_1^T t_1 + t_2] = [R \mid t]$$

The essential matrix of the transformed camera pair is

$$E = [t]_\times R$$

where $t = -R_2 R_1^T t_1 + t_2$ and $R = R_2 R_1^T$.

Theoretical Exercise 2

The essential matrix is defined up to scale, has determinant zero, and has two equal singular values. Since it has 9 parameters with the above constraints, it loses 4 dofs. Hence, the essential matrix has 5 dofs. Because of this, it is really only necessary to have 5 points to determine it, which is what the 5-point algorithm exploits. The 8-point

algorithm on the other hand needs 8-points, as in the name, to determine E because the constraint on the determinant and its singular values are not imposed in the computation of this algorithm, only the constraint of scalability can be applied here.

If the proportion of incorrect correspondences is 25%, to calculate the number of iterations of RANSAC with an eight point solver we need to find an outlier free sample set with 99% probability is

$$T = \left\lceil \frac{\ln(1-\alpha)}{\ln(1-\epsilon^s)} \right\rceil = \left\lceil \frac{\ln(1-0.99)}{\ln\left(1-\left(\frac{3}{4}\right)^8\right)} \right\rceil = 44 \text{ iterations.}$$

Computer Exercise 1

Estimating the essential matrix E with all 6822 matched points we get that the RMS distance between the image points (x_1 and x_2) and corresponding epipolar lines (l_1 and l_2, respectively) is 155.96. Figures 1 and 2 show histograms of the point-to-line distance in pixels (error) from the image points to the epipolar lines for both images respectively when estimating E with all points. The error is large for both images with means of 61.33 and 63.47 pixels respectively. The reason for these large errors is due to the set of image points containing outliers.

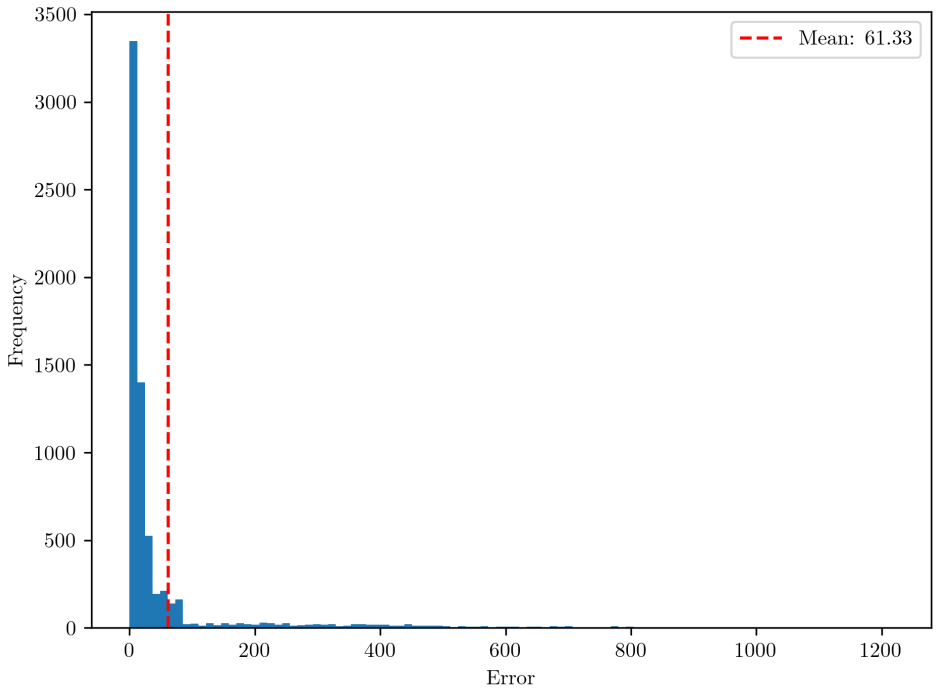

Figure 1: Histogram of the point-to-line distance in pixels (error) from the image points to the epipolar lines of the first image when estimating E with all points. The mean error is 61.33 pixels.

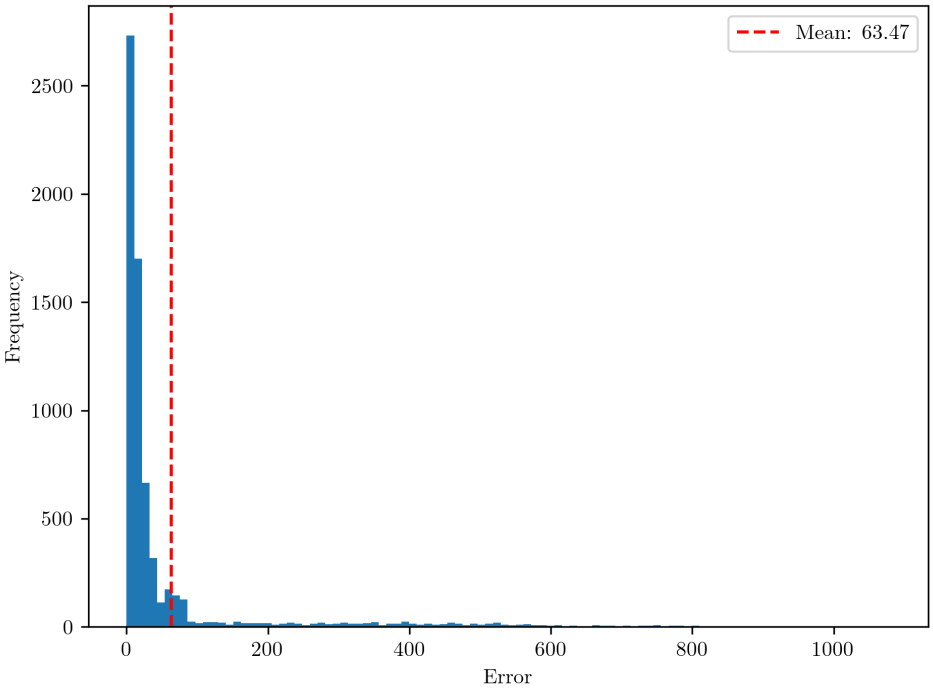

Figure 2: Histogram of the point-to-line distance in pixels (error) from the image points to the epipolar lines of the second image when estimating E with all points. The mean error is 63.47 pixels.

Figures 3 and 4 show the two images, the same 20 random image points and their corresponding epipolar lines. From observation it is noticeable that the epipolar lines do not lie satisfyingly close to the image points. Again, the reason for this is that there are outliers in the set of image points.

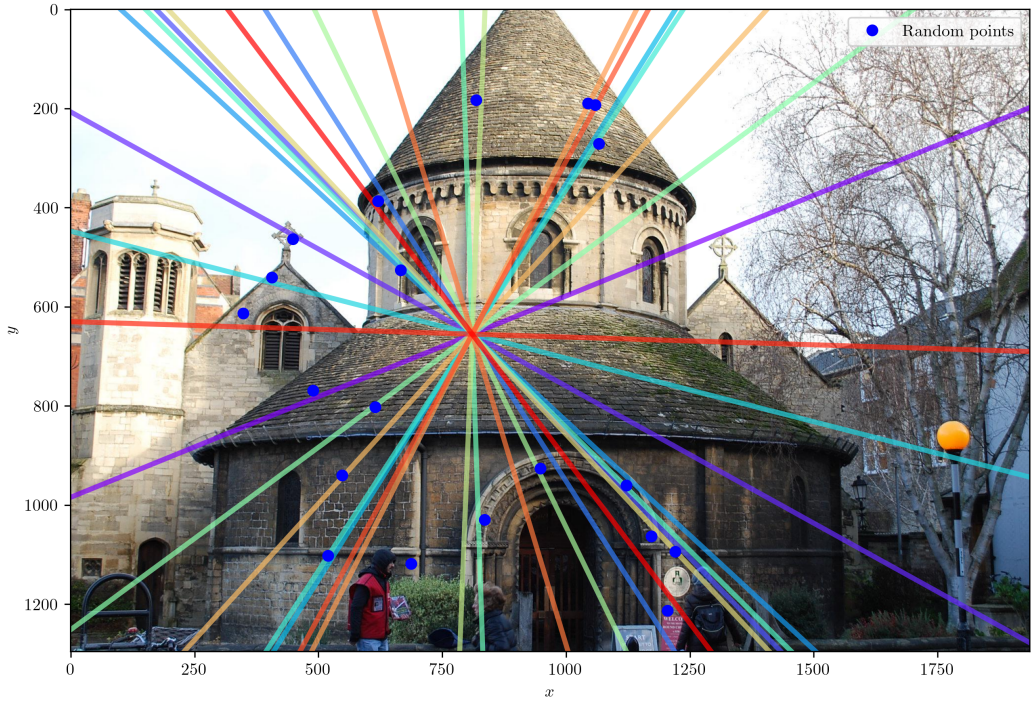

Figure 3: Plot of the first image, 20 random image points and their corresponding epipolar lines.

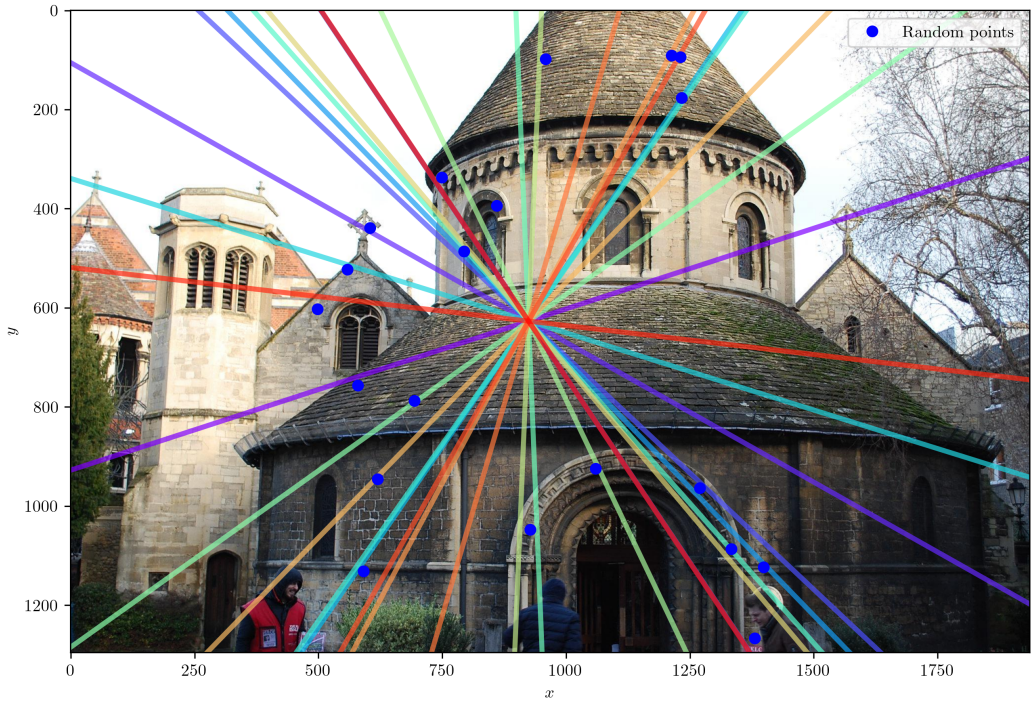

Figure 4: Plot of the second image, 20 random image points and their corresponding epipolar lines.

The errors are greatly decreased by using RANSAC to robustly estimate E. Computing the errors with the 5808 obtained inliers and the robustly estimated E using an error threshold of two pixels, we get that an RMS distance of 0.38. Figures 5 and 6 show histograms of the point-to-line distance between the image points and corresponding epipolar lines in the images with mean distances of 0.27 and 0.30 pixels. These huge improvements show that robustly estimating E using RANSAC has a significant positive effect.

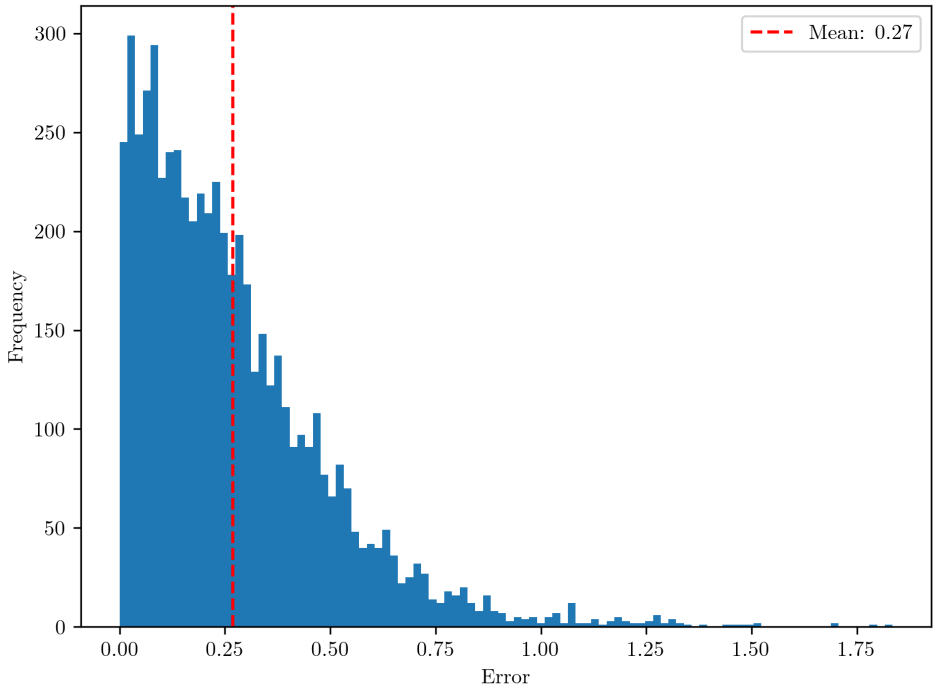

Figure 5: Histogram of the point-to-line distance in pixels (error) from the image points to the epipolar lines of the first image when estimating E with only inliers. The mean error is 0.27 pixels.

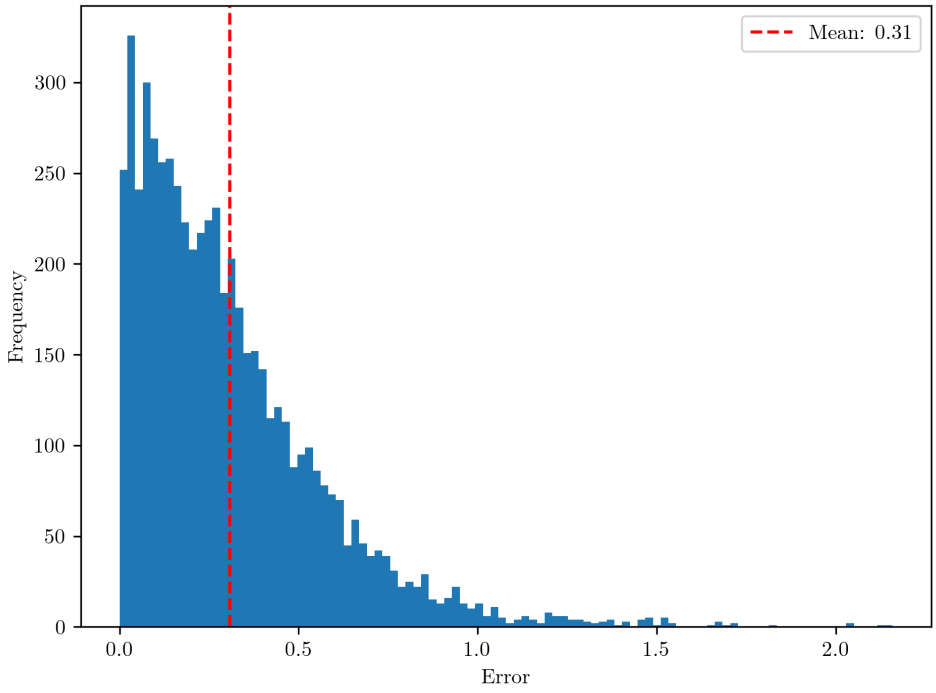

Figure 6: Histogram of the point-to-line distance in pixels (error) from the image points to the epipolar lines of the second image when estimating E with only inliers. The mean error is 0.30 pixels.

Figures 7 and 8 show the two images, the same 20 random inliers and their corresponding epipolar lines. From inspection we can see that the epipolar lines fall on the image points.

This is because the outliers have been excluded from the set of points both when plotting and estimating E. We can conclude that the estimated essential matrix using RANSAC gives better results because the outliers are excluded.

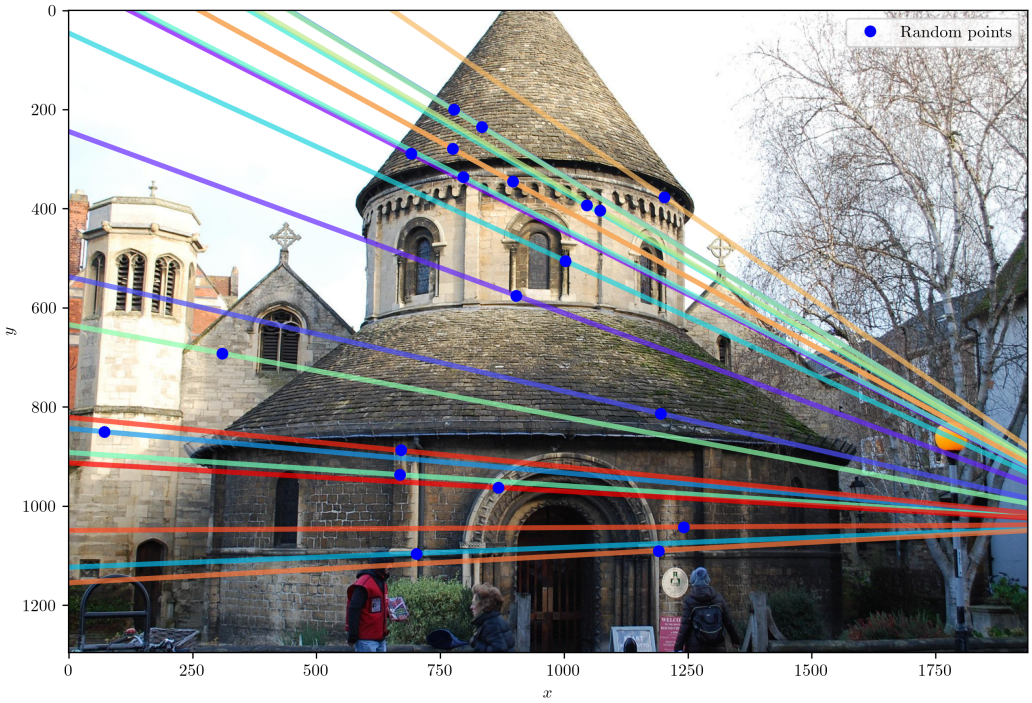

Figure 7: Plot of the first image, 20 random inliers and their corresponding epipolar lines.

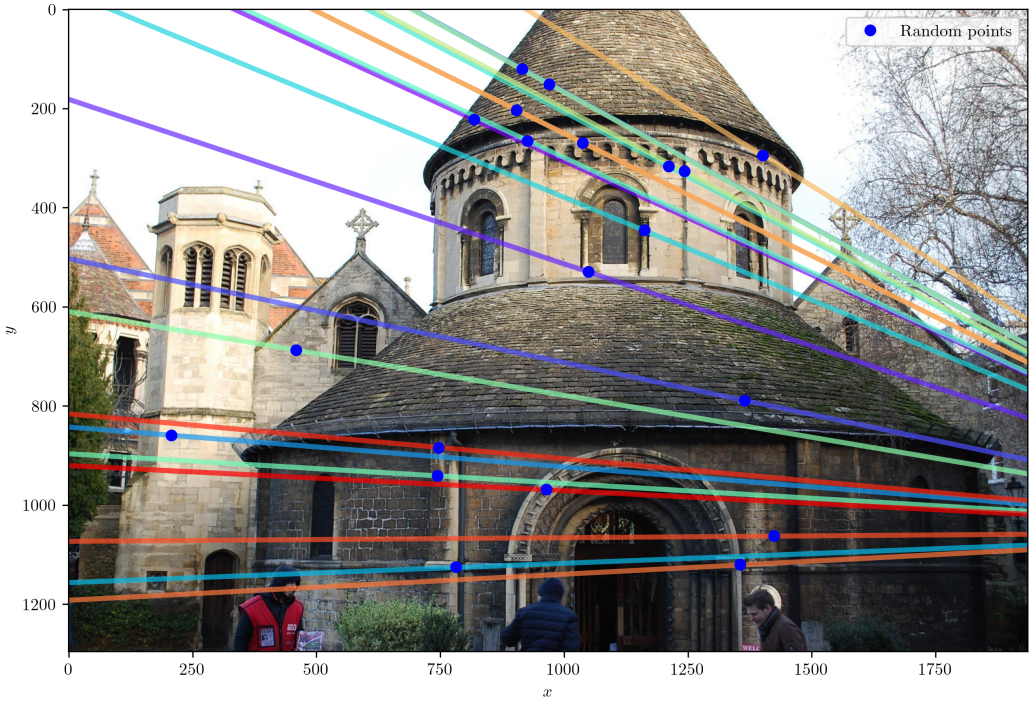

Figure 8: Plot of the second image, 20 random inliers and their corresponding epipolar lines.

Computer Exercise 2

Table 1 shows the number of points found in the first image, second image, the number of matches between the images, and the number of inliers obtained by running the RANSAC algorithm for 100,000 iterations with a threshold error of 4 pixels.

Table 1: The number of points found in each images and the number of matches using SIFT (VLFEAT), and the number of inliers from using RANSAC.

	No. points
fountain1	39,561
fountain2	38,775
Matches	2,604
Inliers	1,840

Figures 9, 10, 11, and 12 show the four different solutions of the second camera with the corresponding 3D reconstructions. From inspection, the second solution (Figure 10) must indeed be the correct solution because it is the only solution that yields expected camera positions and directions pointing toward the 3D reconstruction with a 3D reconstruction that makes sense and is similar to what can be seen in the images. The computation tells us that the solution that yields the most points in front of both cameras is the second solution (Figure 10) which is aligned with the observation of the figures.

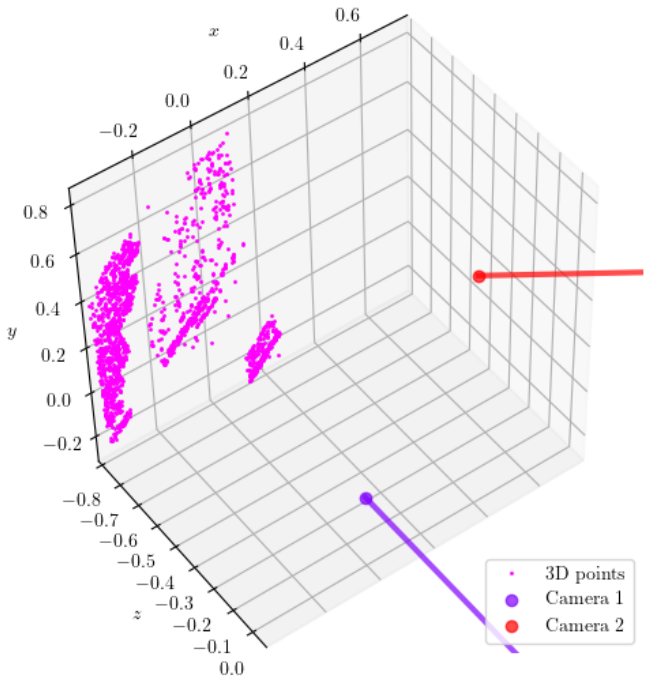

Figure 9: Cameras and 3D reconstruction corresponding to the first solution of camera 2.

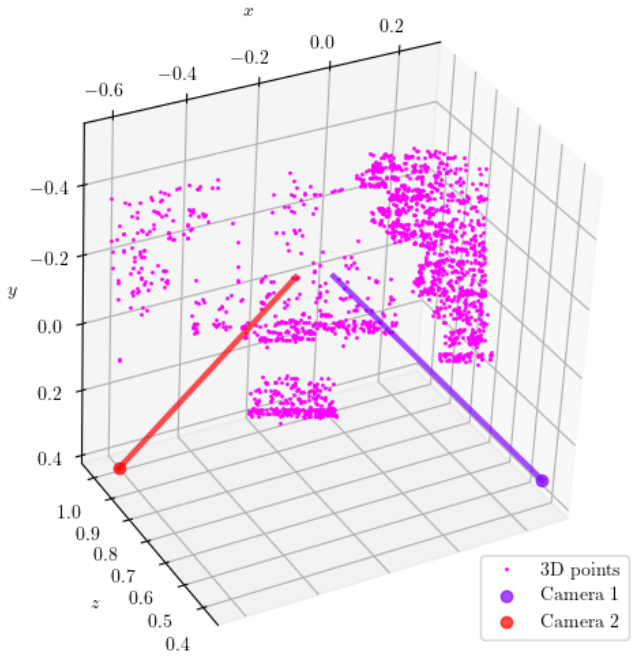

Figure 10: Cameras and 3D reconstruction corresponding to the second solution of camera 2. Camera 1 faces the object from the right and camera 2 faces the object from the left as expected, and the 3D reconstruction is similar to the object in the images.

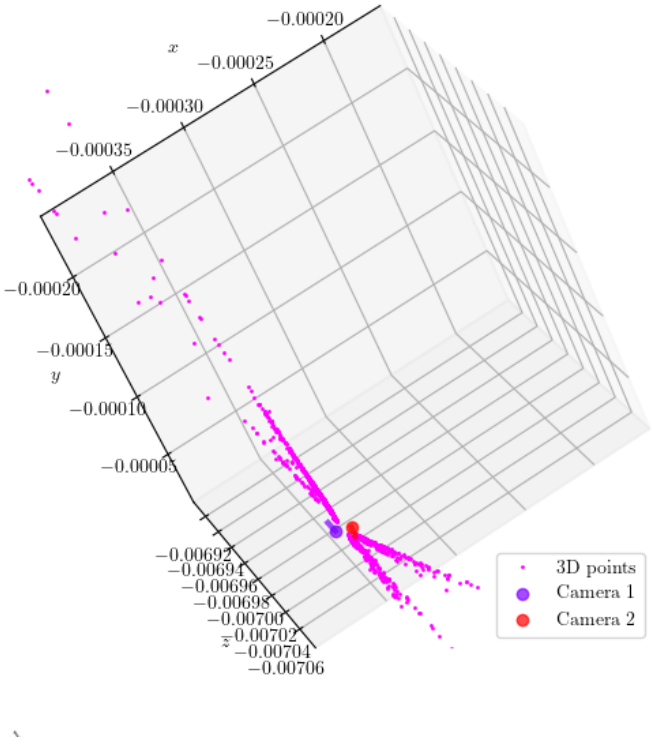

Figure 11: Cameras and 3D reconstruction corresponding to the second solution of camera 2. The computer had trouble zooming in further than this.

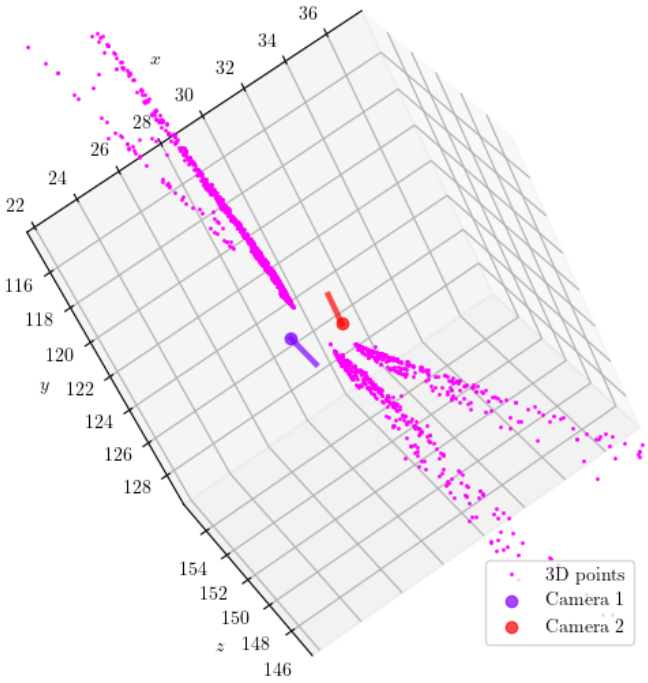

Figure 12: Cameras and 3D reconstruction corresponding to the second solution of the camera 2.

Levenberg-Marquardt for Structure from Motion Problems

Theoretical Exercise 3

a)

Given that

$$\mathbf{r}_i(X_j) = \begin{pmatrix} x_{ij,1} - \frac{P_i^1 \mathbf{X}_j}{P_i^3 \mathbf{X}_j} \\ \\ x_{ij,2} - \frac{P_i^2 \mathbf{X}_j}{P_i^3 \mathbf{X}_j} \end{pmatrix} \in \mathbb{R}^{2\times 1}$$

We compute the jacobian $J_i(\mathbf{X}_j)$ of $\mathbf{r}_i(\mathbf{X}_j)$ by using the chain rule of derivatives as

$$J_i(\mathbf{X}_j) = \begin{bmatrix} \frac{\partial \mathbf{r}_{i,1}(\mathbf{X}_j)}{\partial \mathbf{X}_j} \\ \\ \frac{\partial \mathbf{r}_{i,2}(\mathbf{X}_j)}{\partial \mathbf{X}_j} \end{bmatrix} = \begin{bmatrix} -\left(P_i^1 \mathbf{X}_j \frac{\partial}{\partial \mathbf{X}_j} (\frac{1}{P_i^3 \mathbf{X}_j}) + \frac{1}{P_i^3 \mathbf{X}_j} \frac{\partial}{\partial \mathbf{X}_j} (P_i^1 \mathbf{X}_j) \right) \\ \\ -\left(P_i^2 \mathbf{X}_j \frac{\partial}{\partial \mathbf{X}_j} (\frac{1}{P_i^3 \mathbf{X}_j}) + \frac{1}{P_i^3 \mathbf{X}_j} \frac{\partial}{\partial \mathbf{X}_j} (P_i^2 \mathbf{X}_j) \right) \end{bmatrix}$$

$$= \begin{bmatrix} -\left(-\frac{P_i^1\mathbf{X}_j}{(P_i^3\mathbf{X}_j)^2}P_i^3 + \frac{1}{P_i^3\mathbf{X}_j}P_i^1\right) \\ -\left(-\frac{P_i^2\mathbf{X}_j}{(P_i^3\mathbf{X}_j)^2}P_i^3 + \frac{1}{P_i^3\mathbf{X}_j}P_i^2\right) \end{bmatrix} = \begin{bmatrix} \frac{P_i^1\mathbf{X}_j}{(P_i^3\mathbf{X}_j)^2}P_i^3 - \frac{1}{P_i^3\mathbf{X}_j}P_i^1 \\ \frac{P_i^2\mathbf{X}_j}{(P_i^3\mathbf{X}_j)^2}P_i^3 - \frac{1}{P_i^3\mathbf{X}_j}P_i^2 \end{bmatrix}$$

where $P_i^k\mathbf{X}_j$ for $k = 1,2,3$ are scalars, and P_i^k for $k = 1,2,3$ are row vectors of four columns. Hence, $J_i(\mathbf{X}_j) \in \mathbb{R}^{2\times 4}$.

b)

We know that $\mathbf{r}_i(\mathbf{X}_j) \in \mathbb{R}^{2\times 1}$ and $J_i(\mathbf{X}_j) \in \mathbb{R}^{2\times 4}$. Expanding the linearized reprojection error yields

$$\sum_{i=1}^{m} ||\mathbf{r}_i(\mathbf{X}_j) + J_i(\mathbf{X}_j)\delta\mathbf{X}_j||^2$$

$$= \sum_{i=1}^{m} [\mathbf{r}_i(\mathbf{X}_j)^T\mathbf{r}_i(\mathbf{X}_j) + 2\mathbf{r}_i(\mathbf{X}_j)^T J_i(\mathbf{X}_j)\delta\mathbf{X}_j + \delta\mathbf{X}_j^T J_i(\mathbf{X}_j)^T J_i(\mathbf{X}_j)\delta\mathbf{X}_j]$$

$$= \begin{bmatrix} \mathbf{r}_1(\mathbf{X}_j) \\ \vdots \\ \mathbf{r}_m(\mathbf{X}_j) \end{bmatrix}^T \begin{bmatrix} \mathbf{r}_1(\mathbf{X}_j) \\ \vdots \\ \mathbf{r}_m(\mathbf{X}_j) \end{bmatrix} + 2 \begin{bmatrix} \mathbf{r}_1(\mathbf{X}_j) \\ \vdots \\ \mathbf{r}_m(\mathbf{X}_j) \end{bmatrix}^T \begin{bmatrix} J_1(\mathbf{X}_j) \\ \vdots \\ J_m(\mathbf{X}_j) \end{bmatrix} \delta\mathbf{X}_j + \delta\mathbf{X}_j^T \begin{bmatrix} J_1(\mathbf{X}_j) \\ \vdots \\ J_m(\mathbf{X}_j) \end{bmatrix}^T \begin{bmatrix} J_1(\mathbf{X}_j) \\ \vdots \\ J_m(\mathbf{X}_j) \end{bmatrix} \delta\mathbf{X}_j$$

$$= \mathbf{r}(\mathbf{X}_j)^T\mathbf{r}(\mathbf{X}_j) + 2\mathbf{r}(\mathbf{X}_j)^T J(\mathbf{X}_j)\delta\mathbf{X}_j + \delta\mathbf{X}_j^T J(\mathbf{X}_j)^T J(\mathbf{X}_j)\delta\mathbf{X}_j$$

$$= ||\mathbf{r}(\mathbf{X}_j) + J(\mathbf{X}_j)\delta\mathbf{X}_j||^2$$

where

$$\mathbf{r}(\mathbf{X}_j) = \begin{bmatrix} r_1(\mathbf{X}_j) \\ \vdots \\ r_m(\mathbf{X}_j) \end{bmatrix} \in \mathbb{R}^{2m\times 1}, \text{ and } J(\mathbf{X}_j) = \begin{bmatrix} J_1(\mathbf{X}_j) \\ \vdots \\ J_m(\mathbf{X}_j) \end{bmatrix} \in \mathbb{R}^{2m\times 4}$$

Computer Exercise 3

Table 2 shows the total and median reprojection errors of the estimated 3D reconstruction $\mathbf{X}$ from triangulation, and the same 3D reconstruction but optimized using Levenberg-Marquardt (LM). We can see that the optimized set of 3D points has lower total and median reprojection errors. It is not a drastic improvement but it still shows that optimizing using LM does improve the solution.

Table 2: The total and median reprojection errors of the estimated and optimized 3D points respectively.

Reprojection error of $\mathbf{X}$		
	Estimated	Optimized
Total	3,420.75	3,250.65
Median	1.50	1.44

Figure 13 shows the estimated 3D points, the optimized 3D points and the camera pair. It can be observed that the optimized points align closely with the estimated points but it is difficult to draw conclusions from inspection alone that the set of optimized points would be a better solution. However, from Table 2 we know that this is the case.

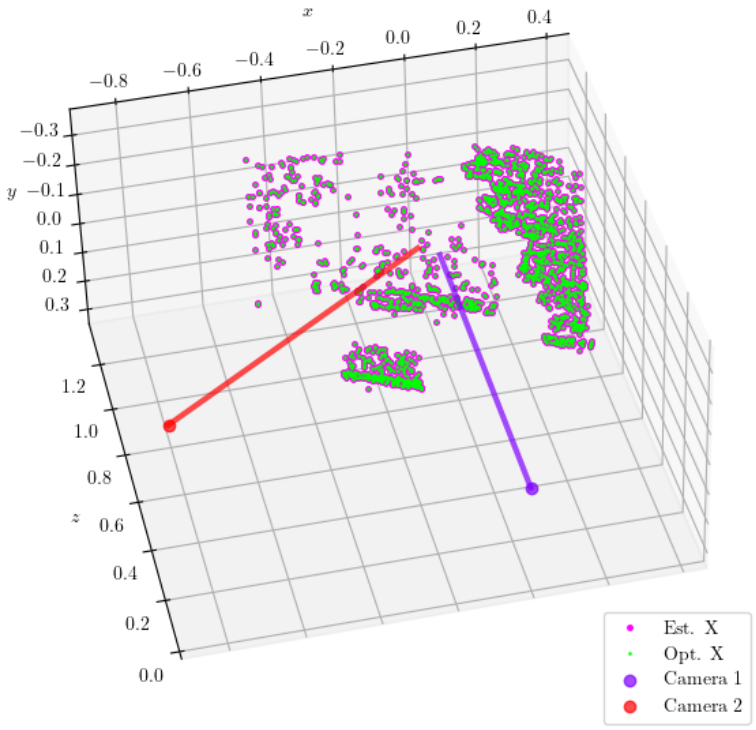

Figure 13: Plot of the estimated 3D points, the optimized 3D points and the camera pair.

Computer Exercise 4

Tables 3, 4, 5 and 6 show reprojection errors where both the estimated 3D points and the 2D points are subjected to different Gaussian noise levels with zero mean. An obvious observation is that the set of optimized points has lower reprojection errors than the noisy points for all cases. What is particularly interesting is that when the noisy set of points give enormous reprojection errors the LM does a fantastic job in reducing the errors by many magnitudes. For example, when the points are subjected to noise of $\sigma_{\mathbf{X}} = 0.1$ m and $\sigma_x = 3$ pixels, after optimization the total error is decreased by almost 19,000% and the median by 54,000%. It is also observable that the largest errors are given when the 3D points are subjected to noise. Another interesting observation is that Tables 4 and 5 show significant large difference in the errors before optimization but about the same

error after optimization. It can be concluded that LM performs spectacularly good at reducing errors caused by Gaussian noise.

Table 3: The total and median reprojection errors of the noisy and optimized 3D points, with the noisy points subjected to Gaussian noise with $\sigma_{\mathbf{X}} = 0$ m and $\sigma_x = 0$ pixels (same as in Computer Exercise 3).

Reprojection error of $\mathbf{X}$.

	Noisy	Optimized
Total	3,420.75	3,250.65
Median	1.50	1.44

Table 4: The total and median reprojection errors of the noisy and optimized 3D points, with the noisy points subjected to Gaussian noise with $\sigma_{\mathbf{X}} = 0$ m and $\sigma_x = 3$ pixels.

Reprojection error of $\mathbf{X}$.

	Noisy	Optimized
Total	69,302.93	20,103.96
Median	31.31	4.85

Table 5: The total and median reprojection errors of the noisy and optimized 3D points, with the noisy points subjected to Gaussian noise with $\sigma_{\mathbf{X}} = 0.1$ m and $\sigma_x = 0$ pixels.

Reprojection error of $\mathbf{X}$.

	Noisy	Optimized
Total	713,674,263.63	20,103.95
Median	260,373.39	4.85

Table 6: The total and median reprojection errors of the noisy and optimized 3D points, with the noisy points subjected to Gaussian noise with $\sigma_{\mathbf{X}} = 0.1$ m and $\sigma_x = 3$ pixels.

Reprojection error of $\mathbf{X}$.

	Noisy	Optimized
Total	735,375,861.66	38,820.97
Median	256,616.18	9.24

Figures 14, 15, 16 and 17 show plots of all the combinations of noisy 3D points, as shown in the tables above, and the respective optimized 3D points. From inspection it is hardly possible to see the errors between the noisy and the optimized points since the noise levels are not significantly large enough to be able to see the errors visually. However, we do know from the tables above that the optimized 3D reconstructions are better in every case.

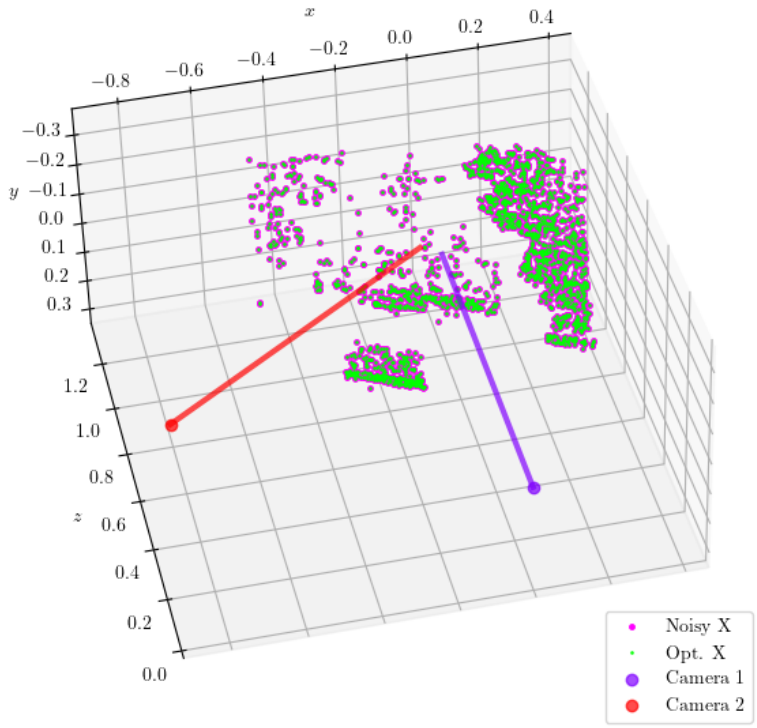

Figure 14: Plot of the noisy 3D points, the optimized 3D points and the camera pair for $\sigma_{\mathbf{X}} = 0$ m and $\sigma_x = 0$ pixels (same as in Computer Exercise 3).

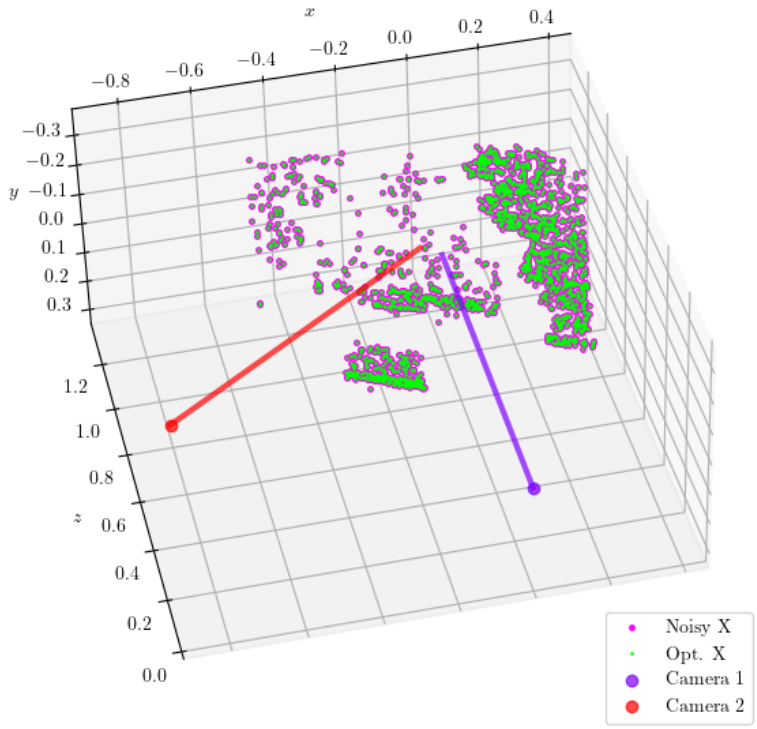

Figure 15: Plot of the noisy 3D points, the optimized 3D points and the camera pair for $\sigma_{\mathbf{X}} = 0$ m and $\sigma_x = 3$ pixels.

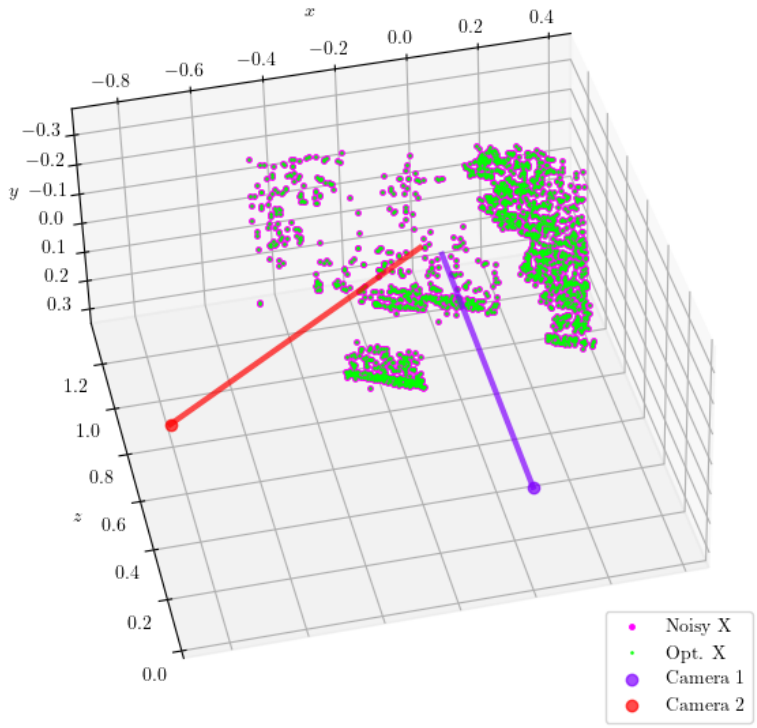

Figure 16: Plot of the noisy 3D points, the optimized 3D points and the camera pair for $\sigma_{\mathbf{X}} = 0.1$ m and $\sigma_x = 0$ pixels.

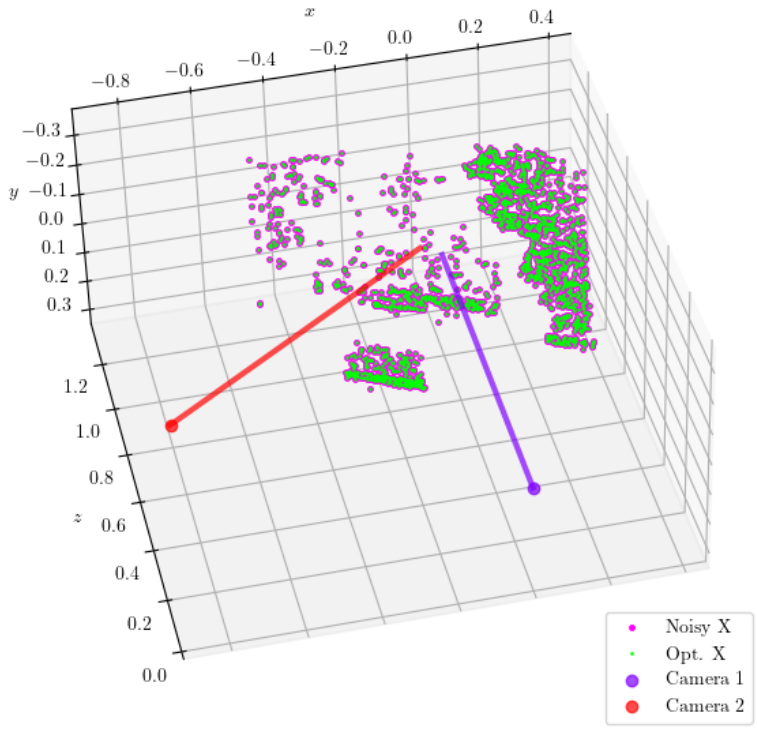

Figure 17: 3D plot of the noisy 3D points, the optimized 3D points and the camera pair for $\sigma_{\mathbf{X}} = 0.1$ m and $\sigma_x = 3$ pixels.

Theoretical Exercise 4

By theory, d is a descent direction for $F(v)$ if

$$\nabla_v F(v) d < 0$$

Let $d = -M\nabla_v F(v)$ and $w = \nabla_v F(v)$, then

$$\nabla_v F(v)^T d = -\nabla_v F(v)^T M \nabla_v F(v) = -w^T M v$$

If $M \succ 0$ it follows that

$$w^T M w > 0 \Rightarrow -w^T M w < 0 \text{ such that } ||w|| \neq 0$$

Hence,

$$\nabla_v F(v) d = -\nabla_v F(v)^T M \nabla_v F(v) < 0$$

Thus, $d = -M\nabla_v F(v)$ is a descent direction for $F(v)$.

In LM we have the update

$$d = -(J(v)^T J(v) + \mu I)^{-1} J(v)^T \mathbf{r}(v)$$

For nonlinear least squares

$$F(v) = ||\mathbf{r}(v)||^2 = \mathbf{r}(v)^T \mathbf{r}(v)$$

where the gradient is

$$\nabla_v F(v) = 2\mathbf{r}(v)^T J(v)$$

let $w = J(v)^T \mathbf{r}(v)$ and $M = (J(v)^T J(v) + \mu I)^{-1}$, then it follows that

$$\begin{aligned} \nabla_v F(v)^T d &\sim -\mathbf{r}(v)^T J(v)(J(v)^T J(v) + \mu I)^{-1} J(v)^T \mathbf{r}(v) \\ &= -w^T M w \end{aligned}$$

$J(v)^T J(v)$ is either semi-positive or positive definite. $\mu I \succ 0$ since $\mu > 0$. Adding them together $J(v)^T J(v) + \mu I \succ 0$. Since the inverse of a positive definite matrix is also a positive definite matrix, $M \succ 0$. We therefore have that

$$w^T M w > 0 \Rightarrow -w^T M w < 0 \text{ such that } ||w|| \neq 0$$

Hence,

$$\nabla_v F(v) d = -2\mathbf{r}(v)^T J(v)(J(v)^T J(v) + \mu I)^{-1} J(v)^T \mathbf{r}(v) < 0$$

Thus, the update $d = -(J(v)^T J(v) + \mu I)^{-1} J(v)^T r(v)$ in LM is a descent direction for $F(v) = ||r(v)||^2$.