AVD624 - Computer Vision

Programming Assignment - 2

Harris Corner Detection

Submitted by

Vatsalya Gupta
SC19B098
B.Tech. ECE VI Semester

Department of Avionics
Indian Institute of Space Science and Technology
Thiruvananthapuram - 695 547

March 2022

1 Hessian Matrix

The filters used for implementation were - Gaussian, first derivative of Gaussian (DoG) and second derivative of Gaussian (LoG). The different parameters considered were:

- Kernel size = 5
- Scale of Gaussian = 1

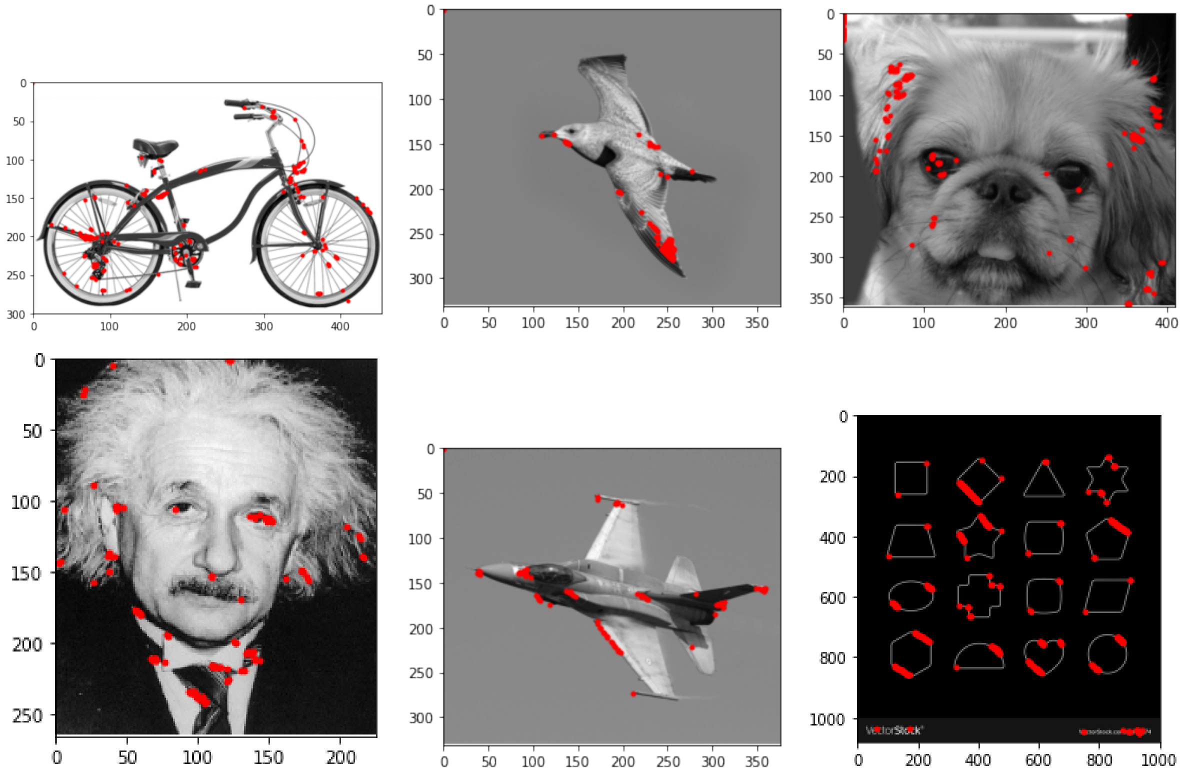

Figure 1: Corner detection using Hessian matrix

The thresholds for eigenvalues for different images were adjusted manually to get the best possible qualitative results. The final threshold values were (from left to right, along row): 21, 12, 6, 15, 10, 38.

Google Colab was used for computation and the times for execution were:

- Input Image 1: 3.74 seconds Input Image 2: 3.38 seconds
- Input Image 3: 4.05 seconds Input Image 4: 1.69 seconds
- Input Image 5: 3.33 seconds Input Image 6: 26.32 seconds

Conclusion: When using Hessian matrix, we can see that the corner repeatability increases as the overlap threshold is increased.

2 Cornerness (determinant and trace)

The filters used for implementation were - Gaussian and first derivative of Gaussian (DoG). The different parameters considered were:

- Kernel size = 5
- Scale of Gaussian = 2 (more smoothing as compared to Hessian matrix)

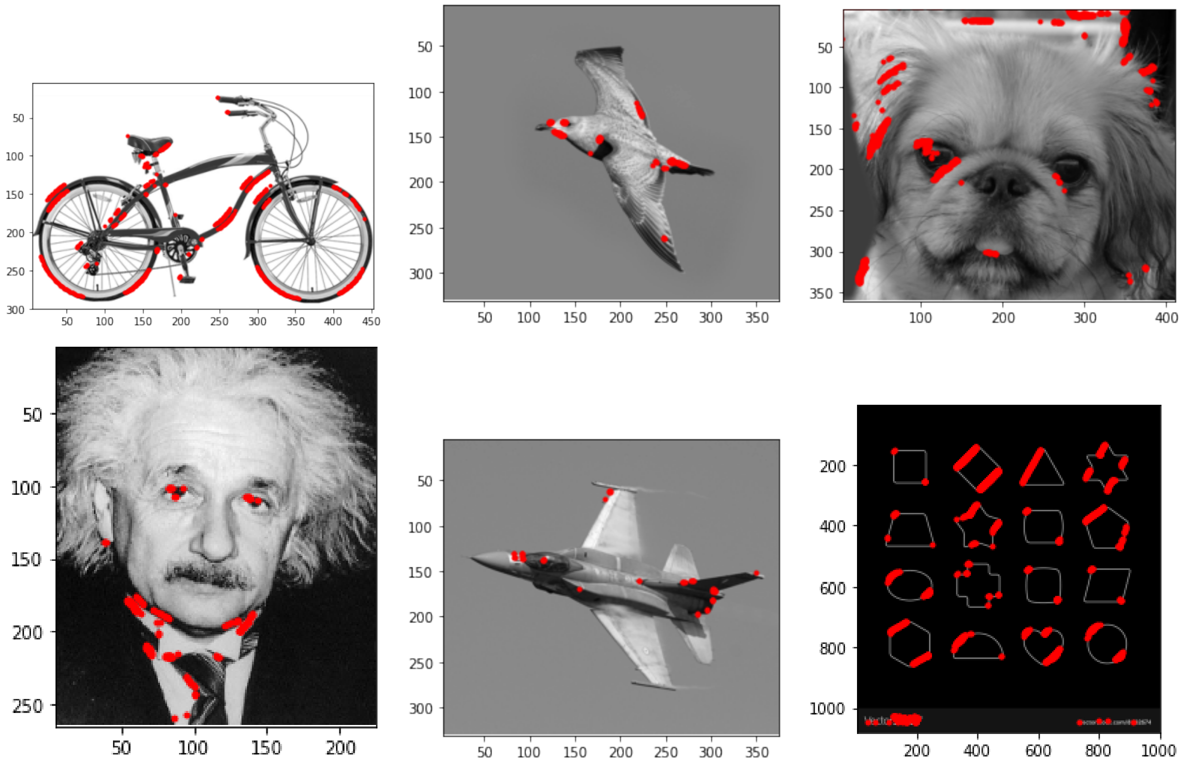

Figure 2: Corner detection using Cornerness measure (determinant and trace)

The thresholds for cornerness for different images were adjusted manually to get the best possible qualitative results. R or $\alpha = 0.6$ was chosen for all the images. The final threshold values chosen were (from left to right, along row): 12, 3, 1, 6, 4, 15.

Google Colab was used for computation and the times for execution were:

- Input Image 1: 2.68 seconds Input Image 2: 2.45 seconds
- Input Image 3: 2.89 seconds Input Image 4: 1.27 seconds
- Input Image 5: 2.45 seconds Input Image 6: 19.65 seconds

Conclusion: Using the Cornerness measure in terms of determinant and trace, gives the most efficient results and reduction in false positives. But there seems to be a compromise on precision when compared to Hessian matrix, as multiple corners smudge to form part of the edge. If we wish to decrease the value of α, then we would also need to lower the threshold value appropriately. This will give us similar results. There is a trade-off between α and threshold.

3 Cornerness (eigenvalues)

The filters used for implementation were - Gaussian and first derivative of Gaussian (DoG). The different parameters considered were:

- Kernel size = 5
- Scale of Gaussian = 2 (more smoothing as compared to Hessian matrix)

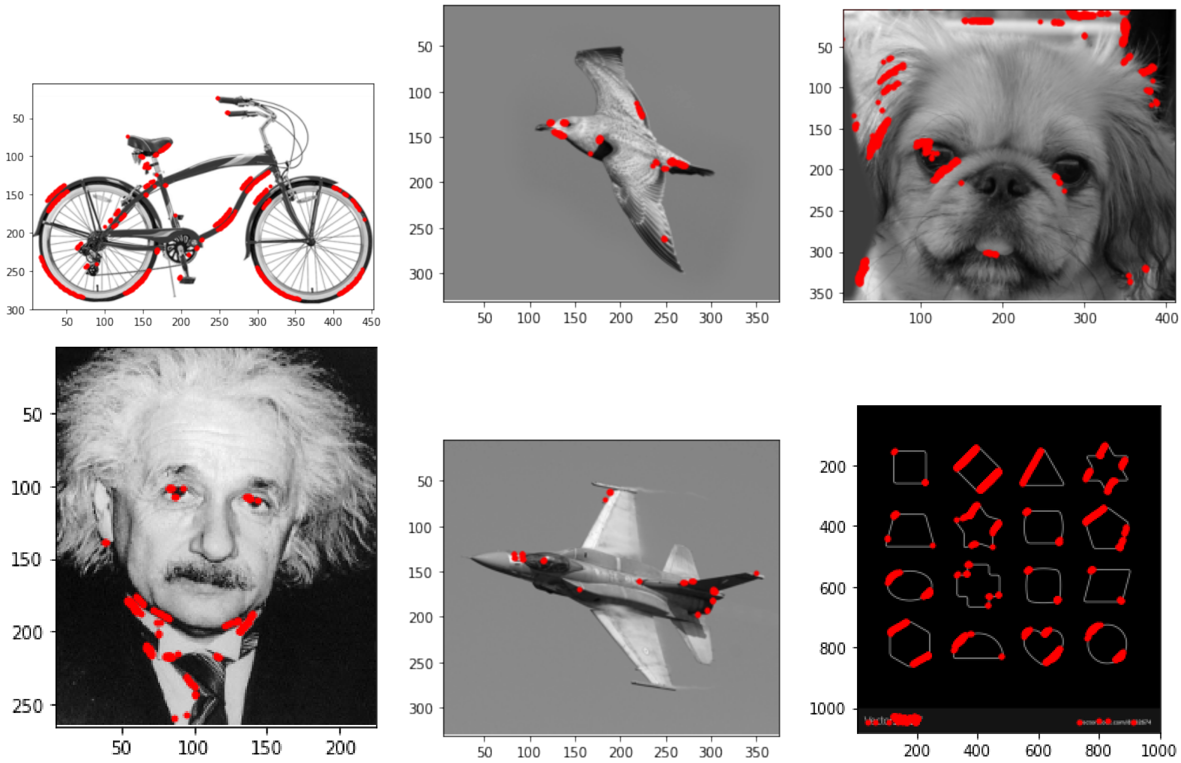

Figure 3: Corner detection using Cornerness measure (eigenvalues)

The thresholds for cornerness (in terms of eigenvalues) for different images were adjusted manually to get the best possible qualitative results. R or $\alpha = 0.6$ was chosen for all the images. The final threshold values chosen were (from left to right, along row): 12, 3, 1, 6, 4, 15.

Google Colab was used for computation and the times for execution were:

- Input Image 1: 6.80 seconds Input Image 2: 6.45 seconds
- Input Image 3: 7.71 seconds Input Image 4: 3.20 seconds
- Input Image 5: 6.44 seconds Input Image 6: 50.92 seconds

Conclusion: Measuring Cornerness in terms of eigenvalues reduces the efficiency if we want to achieve the same accuracy, due to the more number of operations required in computing the eigenvalues and then thresholding.