

CZ4003: Computer Vision

Lab 1: Point Processing + Spatial Filtering + Frequency Filtering Imaging Geometry

Lin HuangJiayin

U1721208D

Contents

[1. Contrast Stretching 3](#_Toc52799738)

[1.1 Checking the minimum and maximum intensities present in the image 3](#_Toc52799739)

[1.2 2 Lines of MATLAB code to do contrast stretching 3](#_Toc52799740)

[2. Histogram Equalization 4](#_Toc52799741)

[2.1 Display the image intensity histogram of P using 10 bins and 256 bins 4](#_Toc52799742)

[2.2 Equalize the histograms for P3 with 10 and 256 bins 5](#_Toc52799743)

[2.3 Rerun the histogram equalization 6](#_Toc52799744)

[3. Linear Spatial Filtering 7](#_Toc52799745)

[3.1 Generate Gaussian filters 7](#_Toc52799746)

[3.2 Filter ‘ntu-gn.jpg’ with Gaussian filter 8](#_Toc52799747)

[3.3 Filter ‘ntu-sp.jpg’ with Gaussian filter 9](#_Toc52799748)

[4. Median Filtering 10](#_Toc52799749)

[4.1 Repeat steps(b)-(e) in section 3 using medfilt2 with different neighbour sizes of 3x3 and 5x5 10](#_Toc52799750)

[5. Suppressing Noise Interference Patterns 11](#_Toc52799751)

[5.1 Power Spectrum with fftshift 11](#_Toc52799752)

[5.2 Power Spectrum without fftshift 11](#_Toc52799753)

[5.3 Set to zero the 5x5 neighbourhood elements at location corresponding to the above peaks 13](#_Toc52799754)

[5.4 Inverse Fourier transform 14](#_Toc52799755)

[5.5 “Free” the primate by filtering out the fence 16](#_Toc52799756)

[6. Undoing Perspective Distortion of Planar Surface 18](#_Toc52799757)

[6.1 ginput function for image coordinates 18](#_Toc52799758)

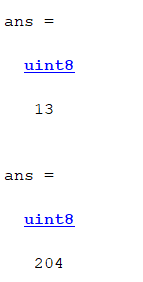
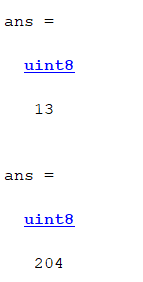
[6.2 Set up matrices to estimate the projective transformation based on the equation (\*) above and wrap the image. 19](#_Toc52799759)

[6.3 Display the image. Is this what you expect? Comment on the quality of the transformation and suggest reasons. 20](#_Toc52799760)

# Contrast Stretching

## 1.1 Checking the minimum and maximum intensities present in the image





Minimum value of P = 13, Maximum value of P = 204

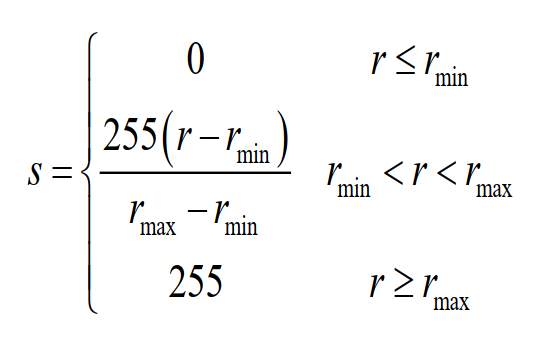
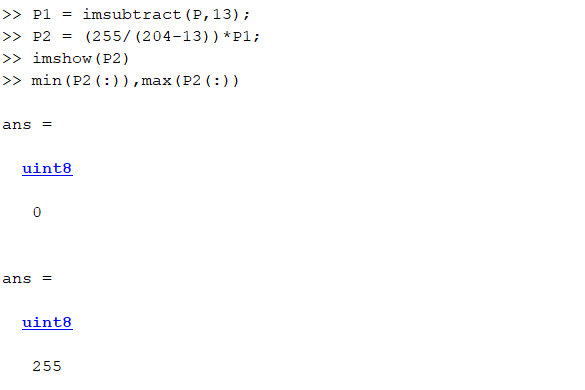


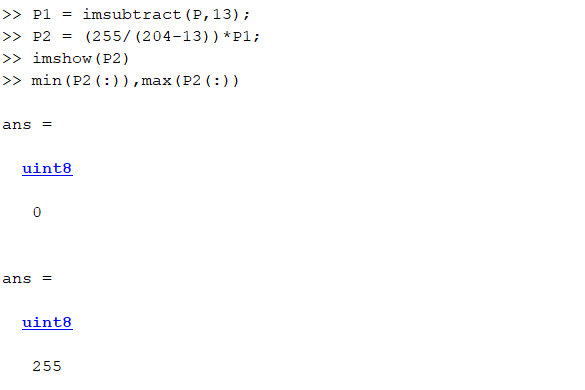
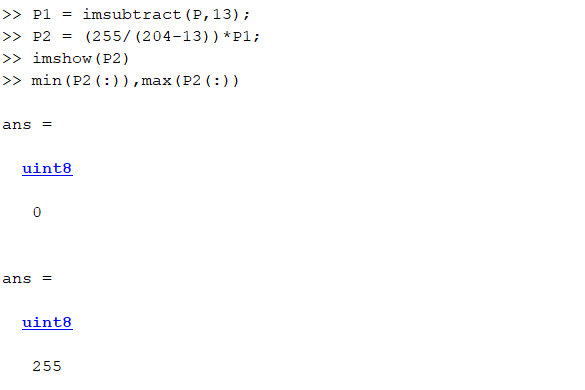
Figure 1. Equation for contrast stretching

## 1.2 2 Lines of MATLAB code to do contrast stretching

Using this equation, r(min) = 13 , r(max) = 204



As such, contrast stretching is done with these 2 operations.



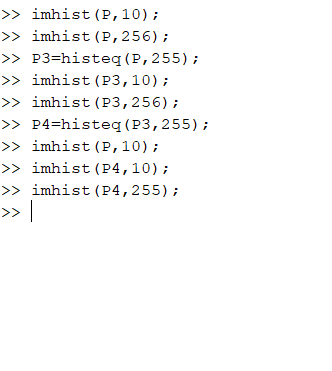
A checking of min and max value has proven that P2 had its minimum and maximum intensified.

Figure 2. Contrast Stretched Vs Original Image

# Histogram Equalization

## 2.1 Display the image intensity histogram of P using 10 bins and 256 bins



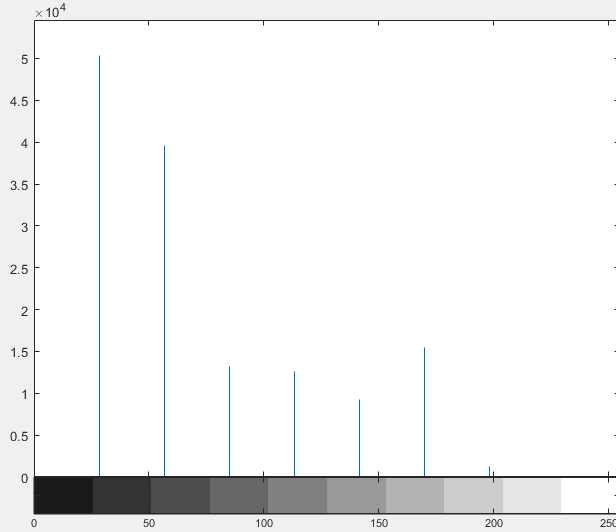
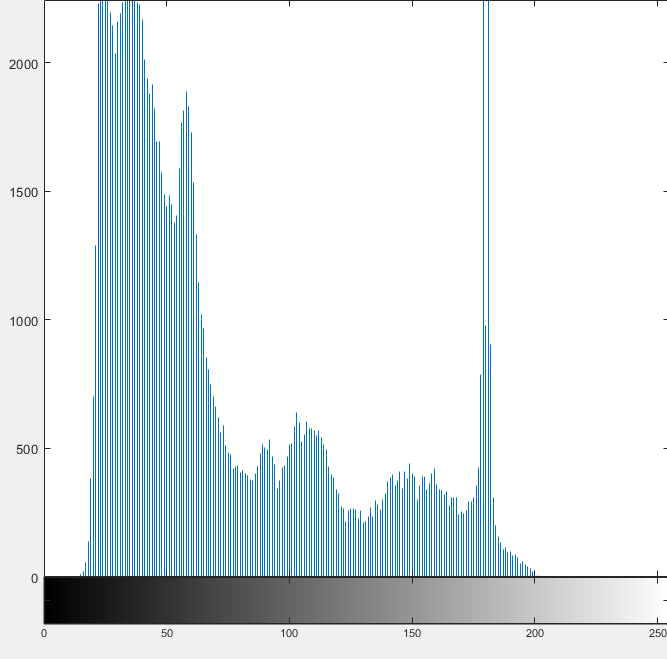
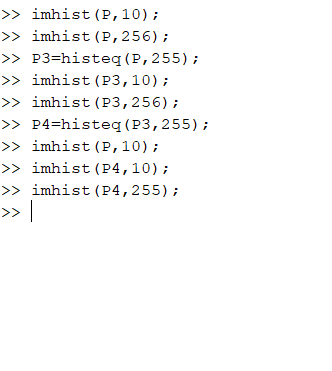
 

Figure 3. P: 10-bins Figure 4. P: 256-bins

Difference between the histograms:

* The image is greatly flattened with the increase in bins. Y axis for the 10-bins histogram is out of 10^4 while the 256-bins histogram is out of 10^3.
  + Having (320x443)pixels in 10-bins results in each bins having a higher number of pixels, as compared to 256-bins
* The increase in bins also shows more details as compared to a trend.

## 2.2 Equalize the histograms for P3 with 10 and 256 bins



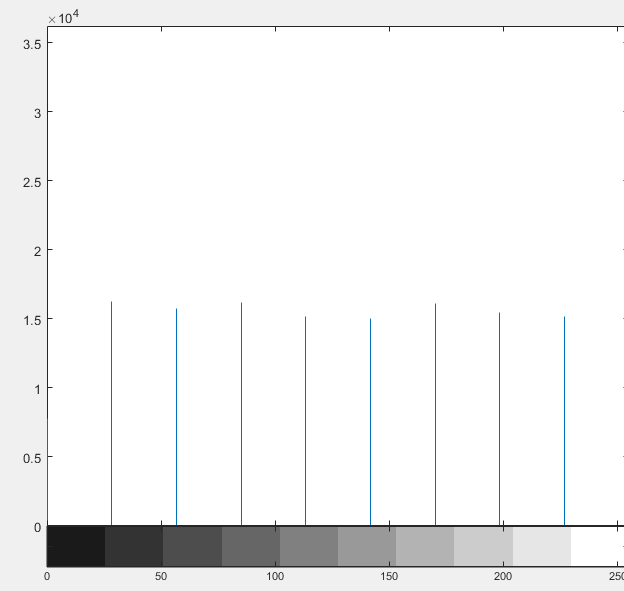
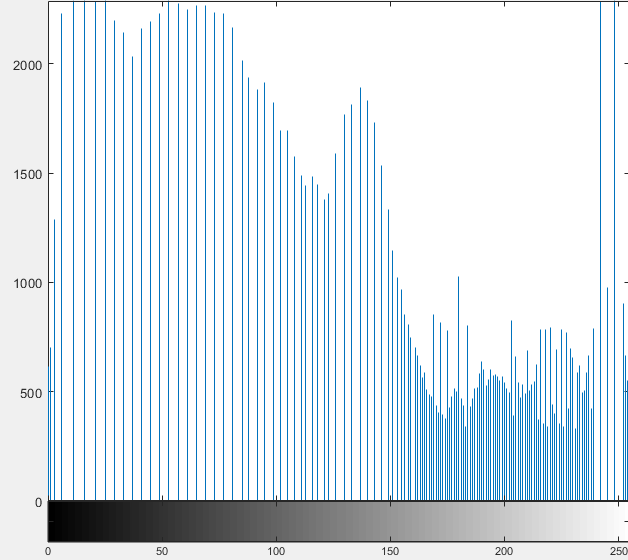
 

Figure 5. P3: 10-bins Figure 6. P3: 256-bins

The histograms for P3 are definitely equalized from P, showing a much more stable histogram.

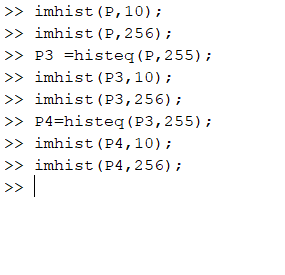
Similarities between the latter 2 histograms:

* Both histograms are more equally distributed in terms of height than their counter part in P(before equalization)

Difference between the latter 2 histograms:

* The 10 bins histogram is well divided unlike the 256 bins histogram
  + 256 bins histogram is heavily clustered at the lighter region
* Like the difference in bins for P: histogram for 256-bins is greatly flattened than 10-bins

## 2.3 Rerun the histogram equalization



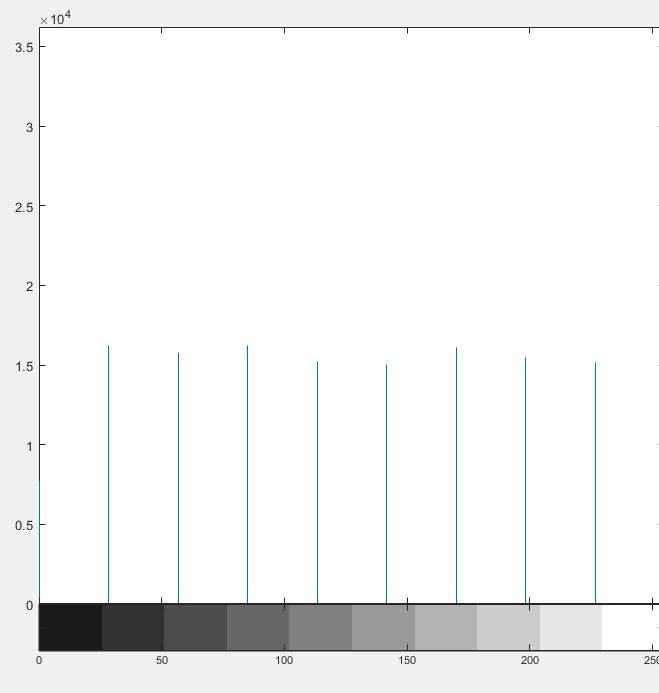
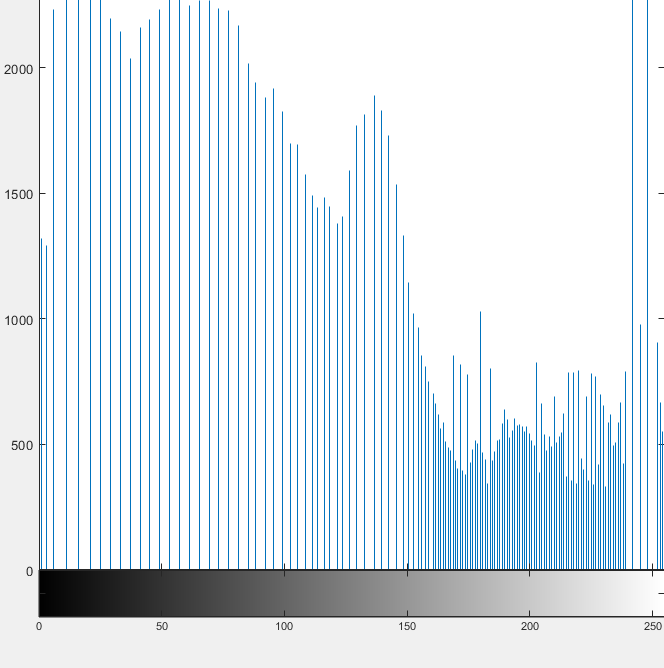
 

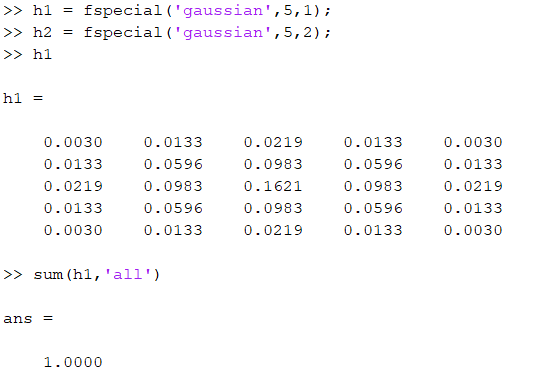
Figure 7. P4: 10-bins Figure 8. P4: 256-bins

The histogram does not become more uniform. There is no changes before or after the equalization (P3-> P4). Histogram has already been flattened and cannot be flatten anymore. There is no changes to the variables in the equation for histogram equalization.

Quoting the lecture note: “Re-applying the algorithm does not change the result”

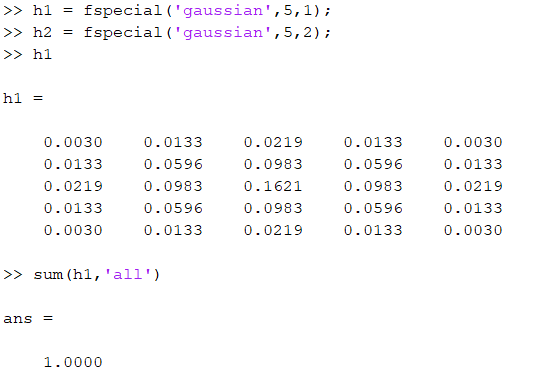
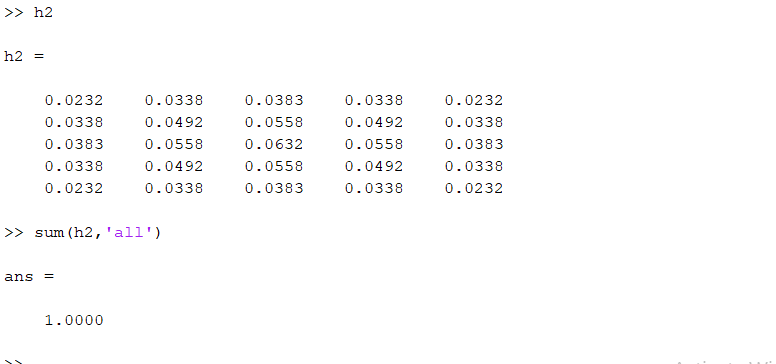
# Linear Spatial Filtering

## 3.1 Generate Gaussian filters

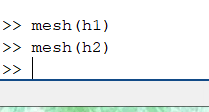


The fspecial is a function provided by MATLAB for gaussian filtering. It normalized the filters at the same time.

To check the normalization:

The elements in H1 and H2 adds up to 1.0.



The mesh function will show the mesh graphs

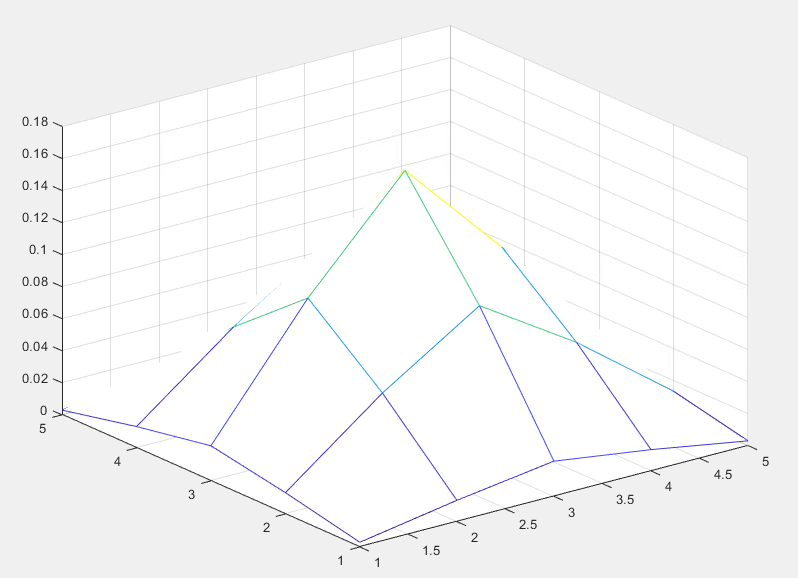
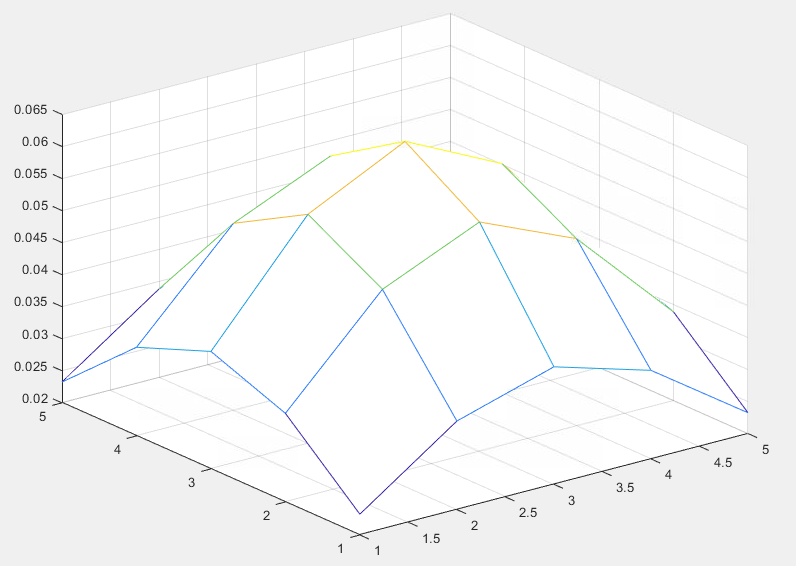
 

Figure 9. mesh h1(sigma 1) Figure 10. mesh h2 (sigma 2)

## 3.2 Filter ‘ntu-gn.jpg’ with Gaussian filter

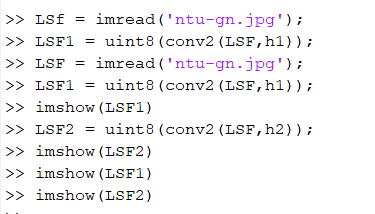




Figure 11. Original image of ‘ntu-gn.jpg’

Figure 12. conv2 of sigma = 1 for ‘ntu-gn.jpg’ Figure 13. conv2 of sigma = 2 for ‘ntu-gn.jpg’

Effectiveness:

* The filters are decently effective. From a normal human (me)’s point of view, I can hardly make out the dots(noise) in the picture. The noise is even less obvious when the sigma is higher.

Trade-offs:

* Lost of details with the increase of sigma. The higher the signma, the higher is the blur filter. As we can see from Figure 11, the original image is much sharper and details

## 3.3 Filter ‘ntu-sp.jpg’ with Gaussian filter



Figure 14. Original image of ‘ntu-sp.jpg’

Figure 15. conv2 of sigma = 1 for ‘ntu-sg.jpg’ Figure 16. conv2 of sigma = 2 for ‘ntu-sg.jpg’

The gaussian filters are better at handling Gaussian noise than speckle noise :

* The result of (sigma = 2.0)’s filter for ‘ntu-sg.jpq’(speckle noise) has more noise than the result of (sigma = 1.0)’s filter for ‘ntu-gn.jpg’(gaussian noise).

# Median Filtering

## 4.1 Repeat steps(b)-(e) in section 3 using medfilt2 with different neighbour sizes of 3x3 and 5x5

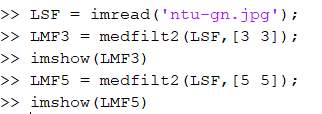


Figure 17. Median filter for ‘ntu-sg.jpg’ of size 3x3 Figure 18. Median filter for ‘ntu-sg.jpg’ of size 5x5

Median filter has removed the noise from ntu-sg.jpg, unlike Gaussian filter.

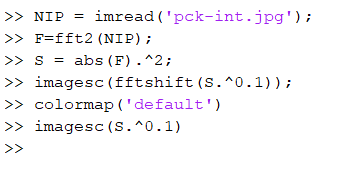
From these 2 images, we can conclude that Gaussian filter is better for Gaussian noise, and median filter is better for speckle noise.

Trade off:

* Sharpness/ Blur-ness
  + Median filter preserves edges better than Gaussian filter
* Certainty
  + It is harder to predict the result of median filter as it is difficult for theoretical analysis, as compared to Gaussian filter

# Suppressing Noise Interference Patterns

## 5.1 Power Spectrum with fftshift



The power spectrum is evaluated with the direct approach

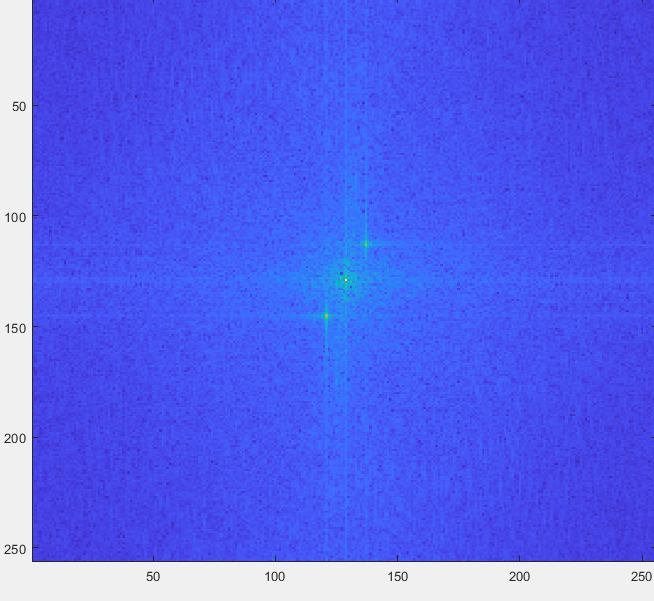


Figure 20. Display of S with fftshift

## 5.2 Power Spectrum without fftshift

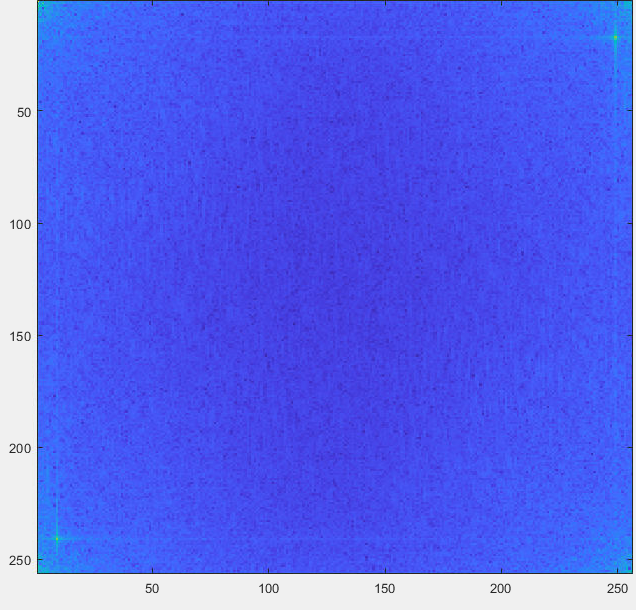


Figure 21. Display of S without fftshift

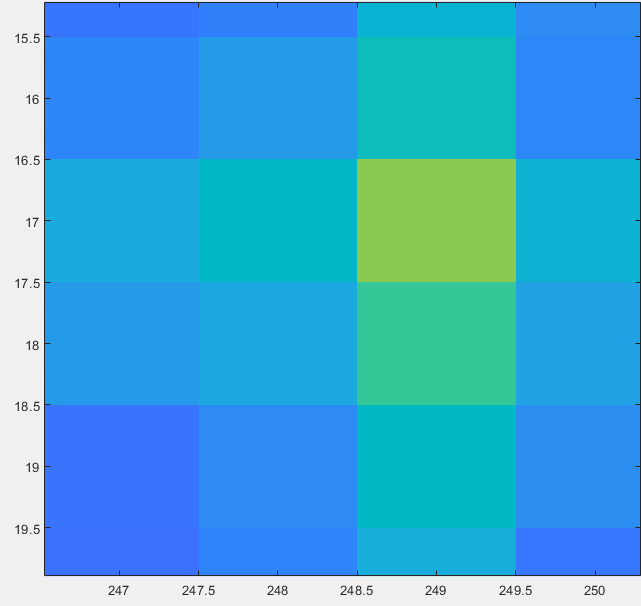
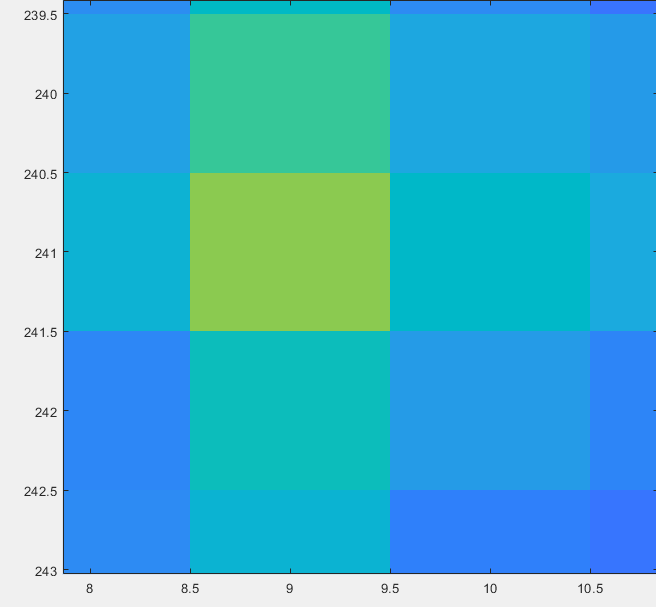
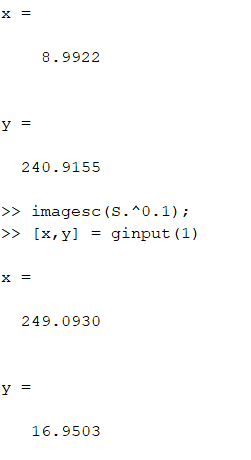
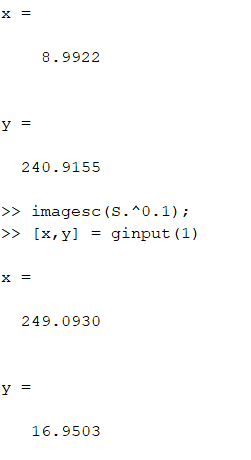


Figure 22. Display of S without fftshift peak 1 Figure 23. Display of S without fftshift peak 2

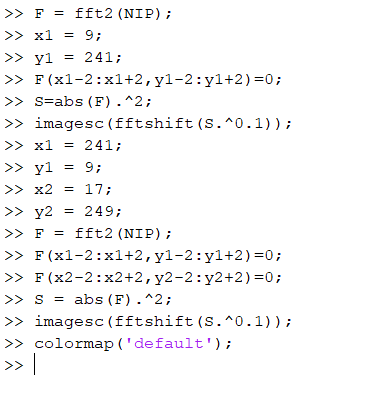
Peak:

* From Figure 22:
  + - x = [8.5, 9.5] , y = [240.5, 241.5] => (9, 241)
  + From figure 23:
    - x = [248.5, 249.5] , y = [16.5, 17.5] => (249, 17)



ginput gives a similar result

## Set to zero the 5x5 neighbourhood elements at location corresponding to the above peaks



(x1-2: x1+2) is used since length of [x1-2, x1-1, x, x1+1, x1+2] = 5

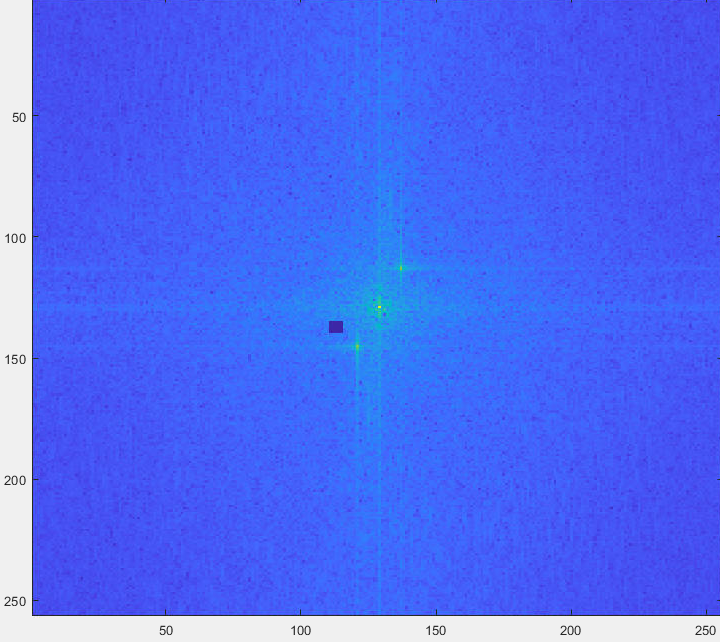
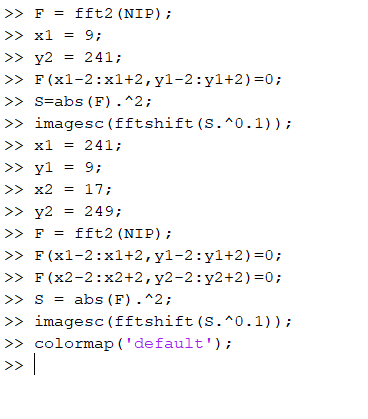


Figure 24. Setting to zero around (9,241)

However, (9,241) does not cover the peak value. This means that the point is (241,9) instead, such that the x and y axes are swapped.  


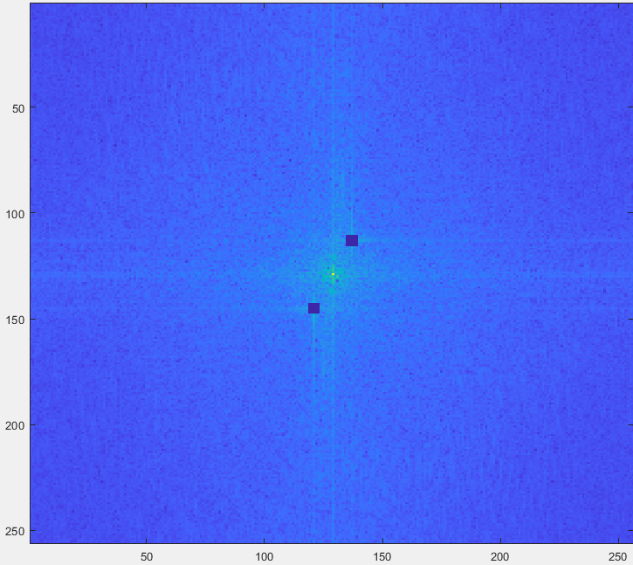


Figure 25. Setting to 0

When the X and Y values are swapped, the symmetric frequency peaks are covered( set to 0)

## 5.4 Inverse Fourier transform

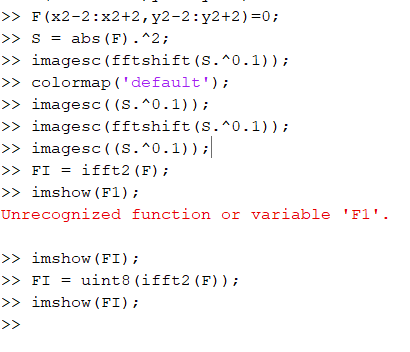


Figure 26. Original image Figure 27. iff2 on F

The noise is reduced efficiently after using ifft2 on the image. There is lesser interference towards the centre of the image. This is due to the fact that the 2 frequency peaks is set to 0

Similarity with step ( c ):

* The interference is now at the edge of the image

Improvement:

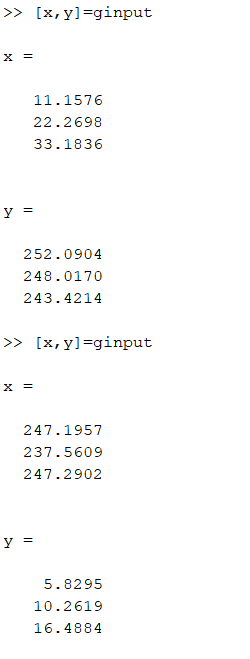
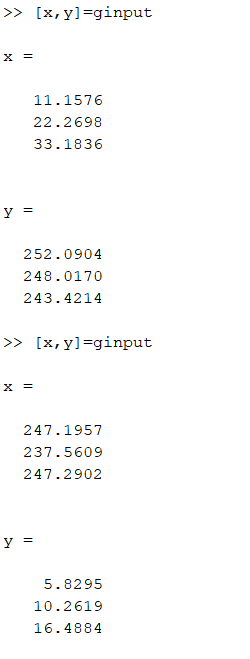
* Increasing the neighbourhood elements around the peak frequencies



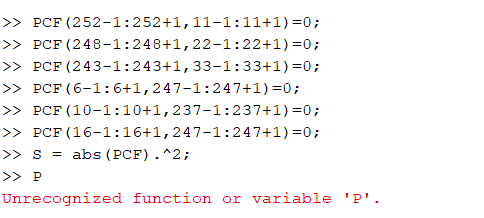
Figure 28. 17x17 neighbourhood elements set to 0

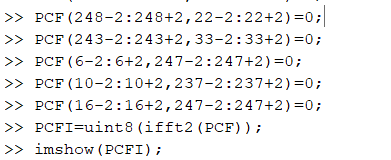
* + We can see that the noise at the edge greatly reduced as compared to Figure 27. However, the middle became worse. This is due to face the high frequency areas are a rectangle instead of a square. Hence, it is possible to obtain an optimal image, but it is definitely not 17x17.
* Using a low-pass filtering
  + Good for supressing high frequencies

## 5.5 “Free” the primate by filtering out the fence



Using ginput, I have picked out 6 points out of the image, 3 from bottom left, 3 from top right. I decided to increase the number of neighbourhood elements gradually.





The above code is repeated iteratively from (+/- 1) to (+/- 5)

Figure 29. 3x3 neighbourhood elements Figure 30. 5x5 neighbourhood elements

Figure 31. 7x7 neighbourhood elements Figure 32. 9x9 neighbourhood elements

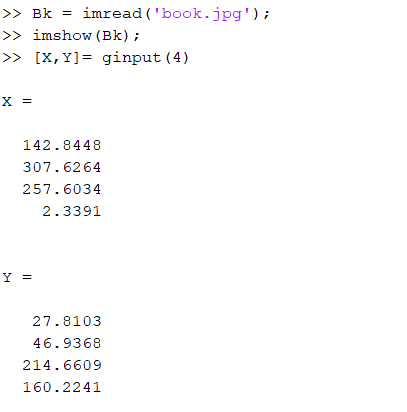


Figure 33. 11x11 neighbourhood elements

From the figures above, we can see that a 9x9 neighbourhood elements provide a the best trade off between the clarity of the primate and the cage. While 11x11 is able to blur the cage more, the primate has lost too many details.

# Undoing Perspective Distortion of Planar Surface

## 6.1 ginput function for image coordinates



I have used “Bk” instead of “P’ accidentally, so I will be using “Bk”.

Using ginput function I have obtained the following coordinates:

Top left: (142.8448, 27.8103) Top right: (307.6264, 46.9368)

Bottom left: (2.3391, 160.2241) Bottom right: (257.6034, 214.6609)

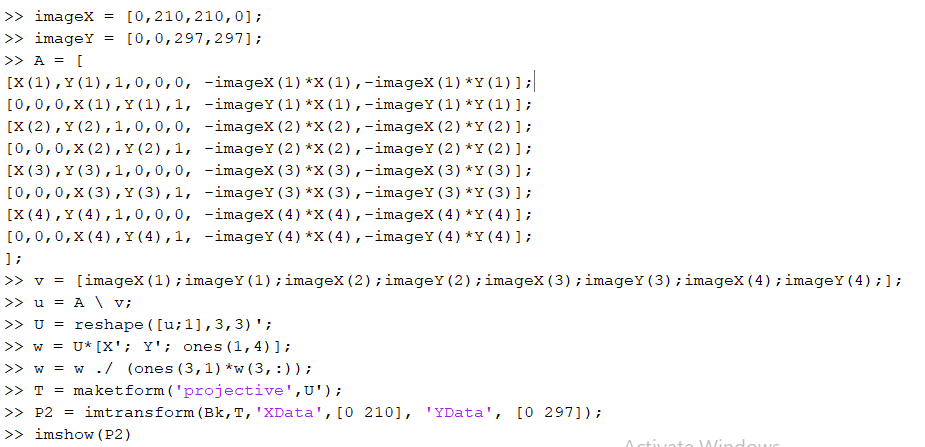
For the desired image dimensions of a 210mm x 297mm:

Top left: (0,0) Top right: (210,0)

Bottom left: (0,297) Bottom right: (210,297)

## 6.2 Set up matrices to estimate the projective transformation based on the equation (\*) above and wrap the image.

Following the information provided:



## 6.3 Display the image. Is this what you expect? Comment on the quality of the transformation and suggest reasons.

Figure 34. Original image Figure 35. Transformed image

Honestly, I did not expect much from the process (equations), so the result was a pleasant surprise. The transformed image actually provides me more information of the graphics, in the middle, than the original image did. However, the text on the top worsened.

Comment on the quality:

* The distortion is fixed, book is now facing straight
  + Having the 4 points scales the image proportionally
* The saturation of the image has been reduced
  + The transformation matrix is unable to preserve and map the pixel value
* The clarity of the text dropped
  + The resolution of the image in the origin is too low for an acceptable transformation of the text