DIP Homework Assignment #1

姓名：鍾毓安

學號：B01902040

系級：資訊四

Email: [iamyuanchung@gmail.com](mailto:iamyuanchung@gmail.com)

**Source Code**

The following script/functions were implemented and their introductions were described as follows. As for detailed descriptions, please refer to the function files.

1. README.m: This is the main script that works like the main function. All the required tasks, including Warm-Up, Problem 1 & 2 will be done one by one when README.m is executed.
2. flipVertical.m: The function flips the given 2D image matrix up-side-down.
3. flipHorizontal.m: The function reverses the given 2D image matrix left-to-right.
4. plotHistogram.m: For a given 2D gray-scale image matrix, plotHistogram returns a 256-length 1D array where the i-th entry stores the number of pixels whose value equals to (i – 1).
5. histEqual.m: The function performs the histogram equalization on the given 2D image matrix. Histogram equalization enhances the given image G by first converting the histogram of G into cumulative distribution function (CDF), then mapping the CDF to a uniform one so as to make the histogram of the enhanced G more uniformly distributed.
6. localHistEqual.m: The function performs the local histogram equalization on the given 2D image matrix. Similar to histogram equalization, local histogram equalization also tries to enhance the given image using the histogram. However, local histogram equalization introduces an extra window that will go through the image from top to bottom and from left to right. Then, histogram equalization is applied to the region inside the window, and this is how the word “local” comes from. The original histogram equalization is sometimes referred to as the global histogram equalization to make it distinguishable from the local version.
7. logTransform.m: The function performs log transform on the given 2D image matrix. Log transform enhances the low intensity pixels due to its property of concave downward. Especially, for each entry of an given image matrix , where is already scaled to range by dividing 255, the log transform does the following transformation:

,

where is the scaling constant that ensures the resulting image has a maximum magnitude of 255.  
(Reference: http://homepages.inf.ed.ac.uk/rbf/HIPR2/pixlog.htm)

1. invLogTransform.m: The function performs inverse log transform on the given 2D image matrix. Different from log transform, the concave upward property makes the inverse log transform more useful in a situation when more details of high intensity pixels are desired. Especially, for each entry of an given image matrix , where is already scaled to range by dividing 255, the inverse log transform does the following transformation:

,

where is the scaling constant that ensures the resulting image has a maximum magnitude of 255.

1. powerLawTransform.m: The function performs the power-law transform on the given 2D image matrix. For each entry of an given image matrix , where is already scaled to range by dividing 255, the power-law transform does the following transformation:

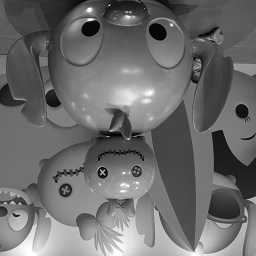
,

where is the scaling constant that ensures the resulting image has a maximum magnitude of 255, and is a parameter that can be flexibly controlled. When , ’s property of concave downward makes it a suitable choice for enhancing image with low intensity pixels; when , becomes concave upward, which is a good option for enhancing image with high intensity pixels; and when , simply performs the linear mapping.  
(Reference: http://funnotes.net/tofpages/TopicOfFortnight.php?tofTpcFl=topicoffortnight22)

1. addGaussianNoise.m:
2. addSaltPepperNoise.m:

**Warm-Up: Simple Manipulation**

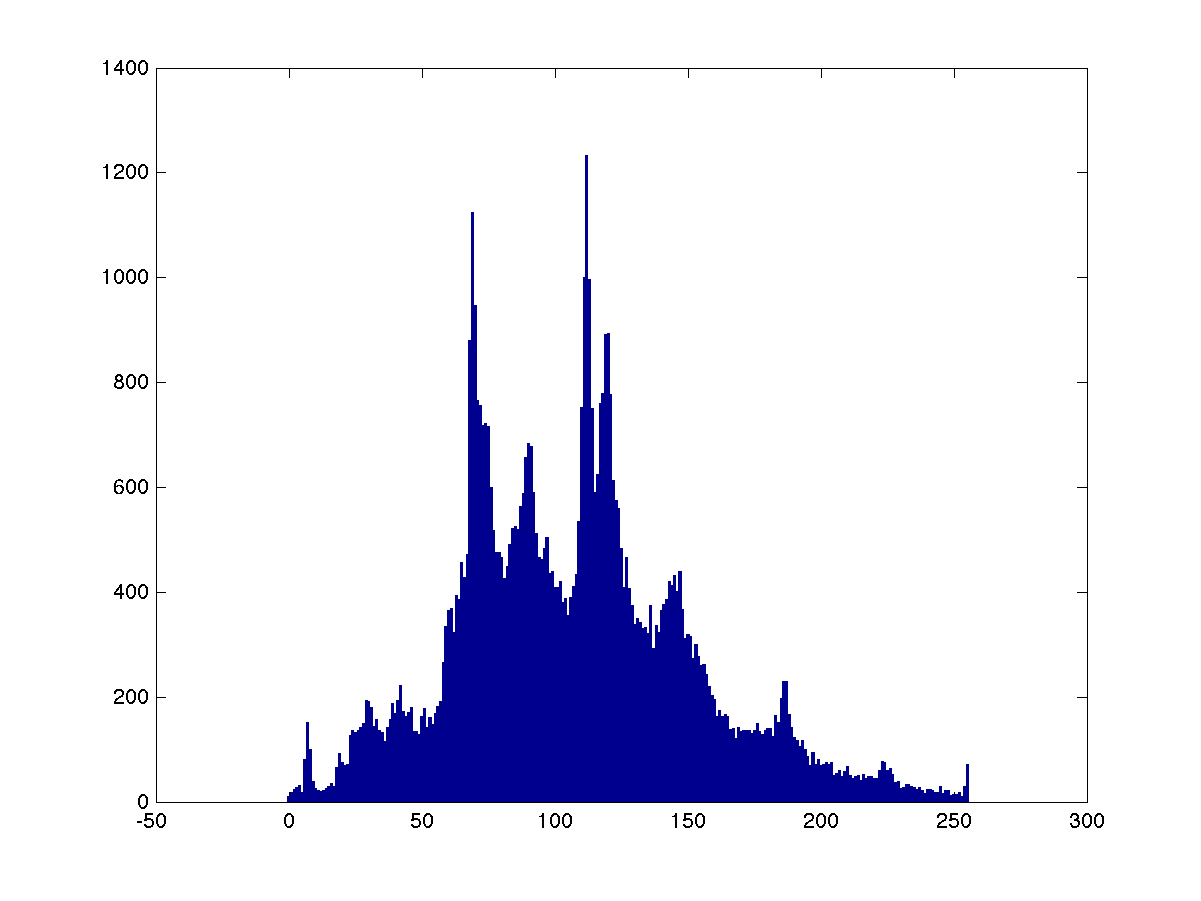
Flipped I vertically and horizontally. The resultant images were displayed as follows.

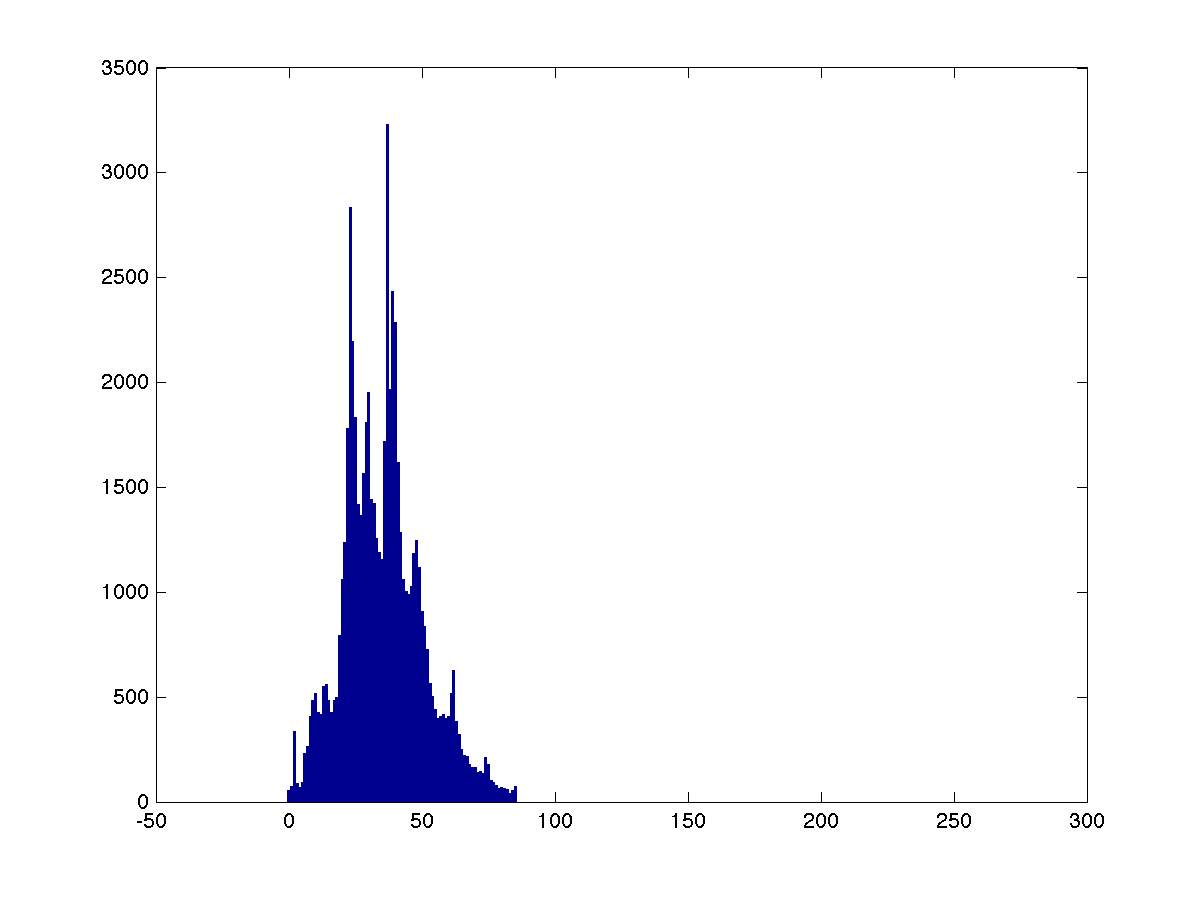
sample1.vertical.png: Flipped I vertically sample1.horizontal.png: Flipped I horizontally

**Problem 1: Image Enhancement**

1. Plot the histograms of I and D. What can you observe from these two histograms? What can you do to make D look like I?



sample1.hist.png: The histogram of image I



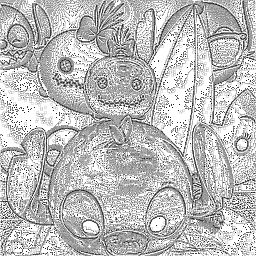
sample2.hist.png: The histogram of image D

1. Perform histogram equalization on D and out the result as H.



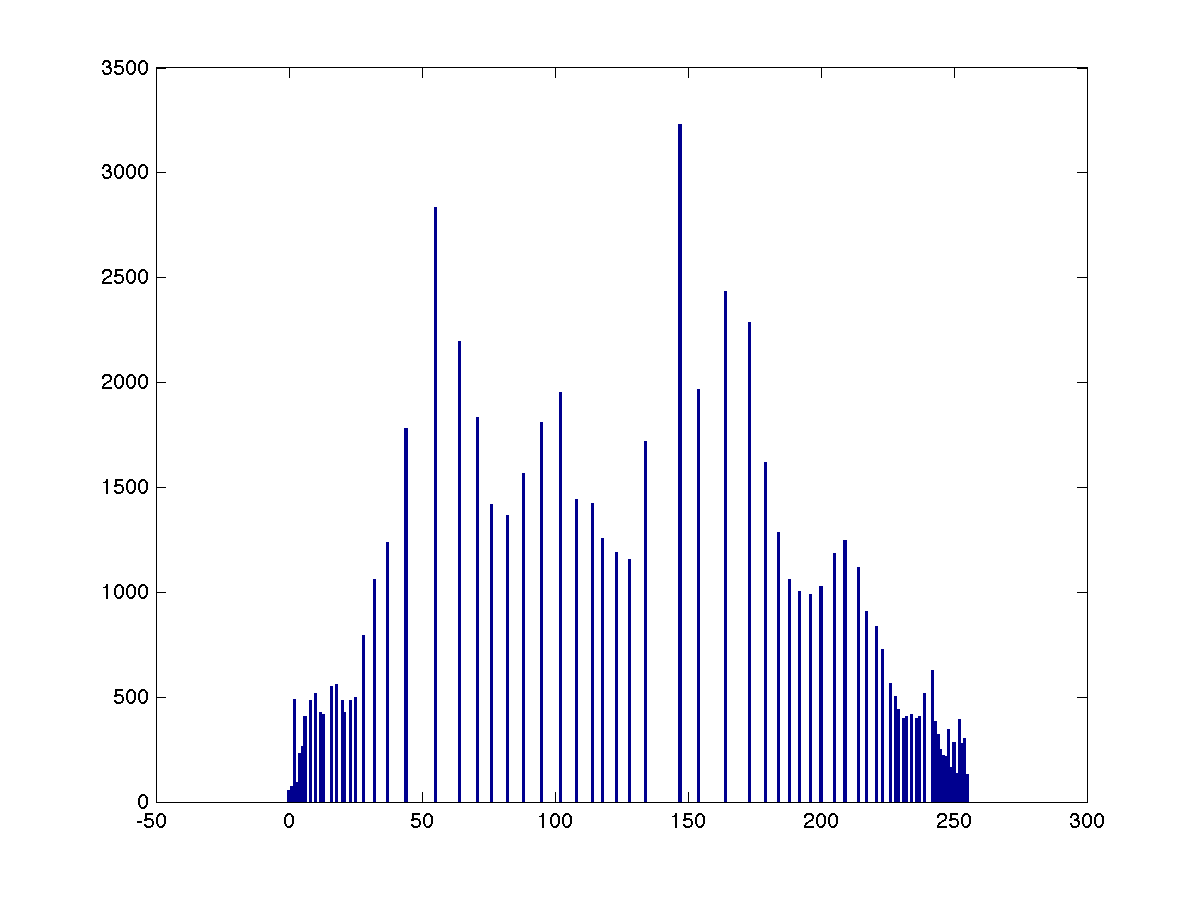
sample2.hist.equal.png: The histogram equalized image D

1. Perform local histogram equalization on image D and output the result as L.

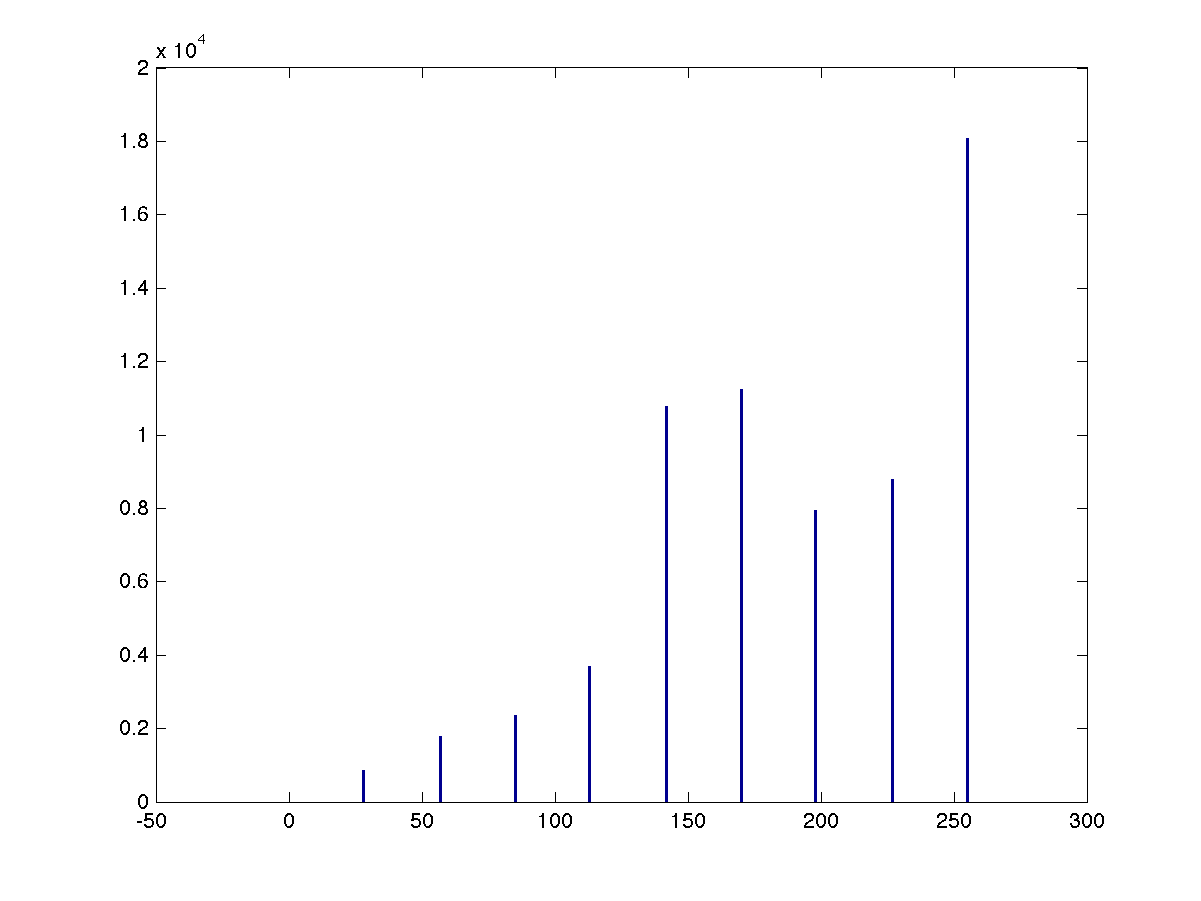


sample2.local.hist.equal.png: The local histogram equalized image D

1. Plot the histograms of H and L. What’s the main difference between local and global histogram equalization?



sample2.hist.equal.hist.png: The histogram of the histogram equalized image D



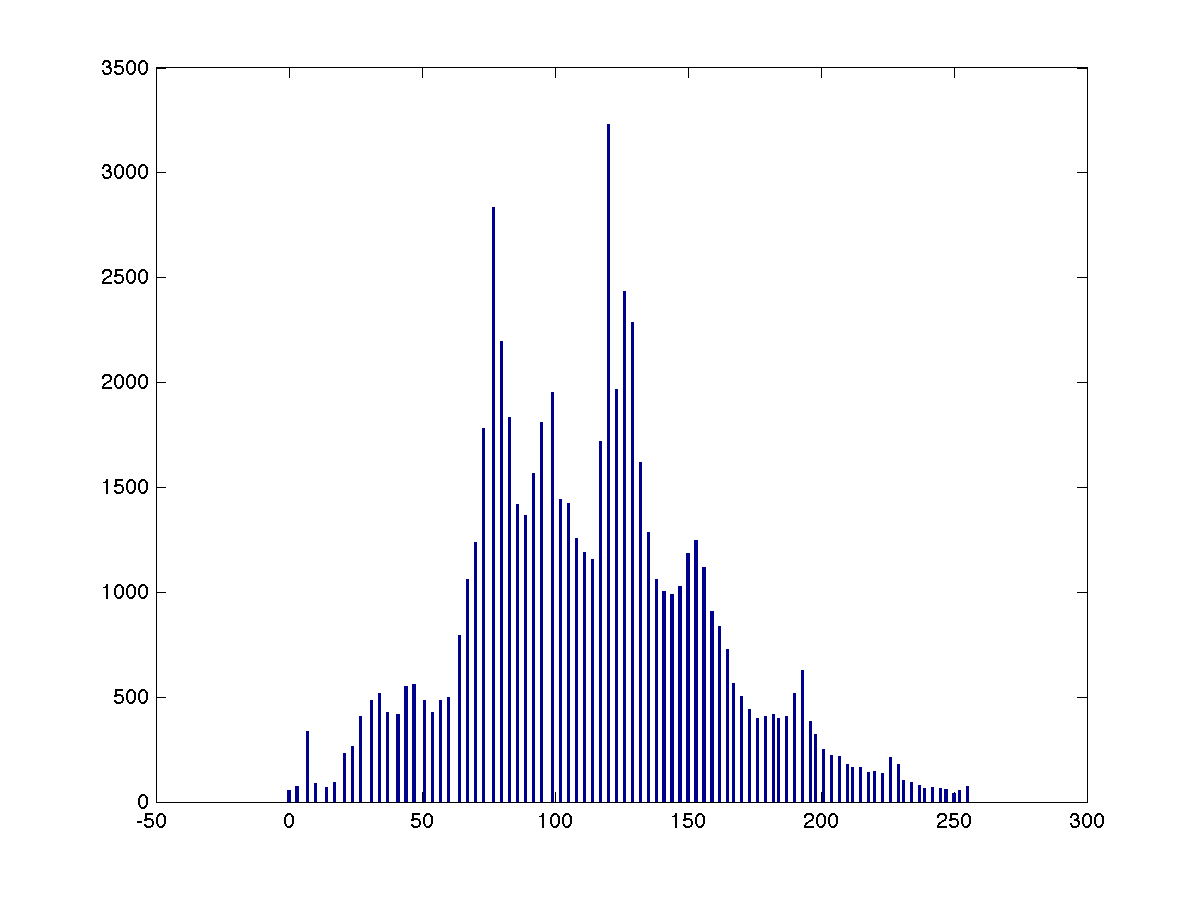
sample2.local.hist.equal.hist.png: The histogram of the local histogram equalized image D

1. Perform the log transform, inverse log transform and power-law transform to enhance image D. Please adjust the parameters as best as you can. Show the parameters, output images and corresponding histograms. Provide some discussions on the results as well.

**Log transform**



sample2.log.png: The log transformed image D

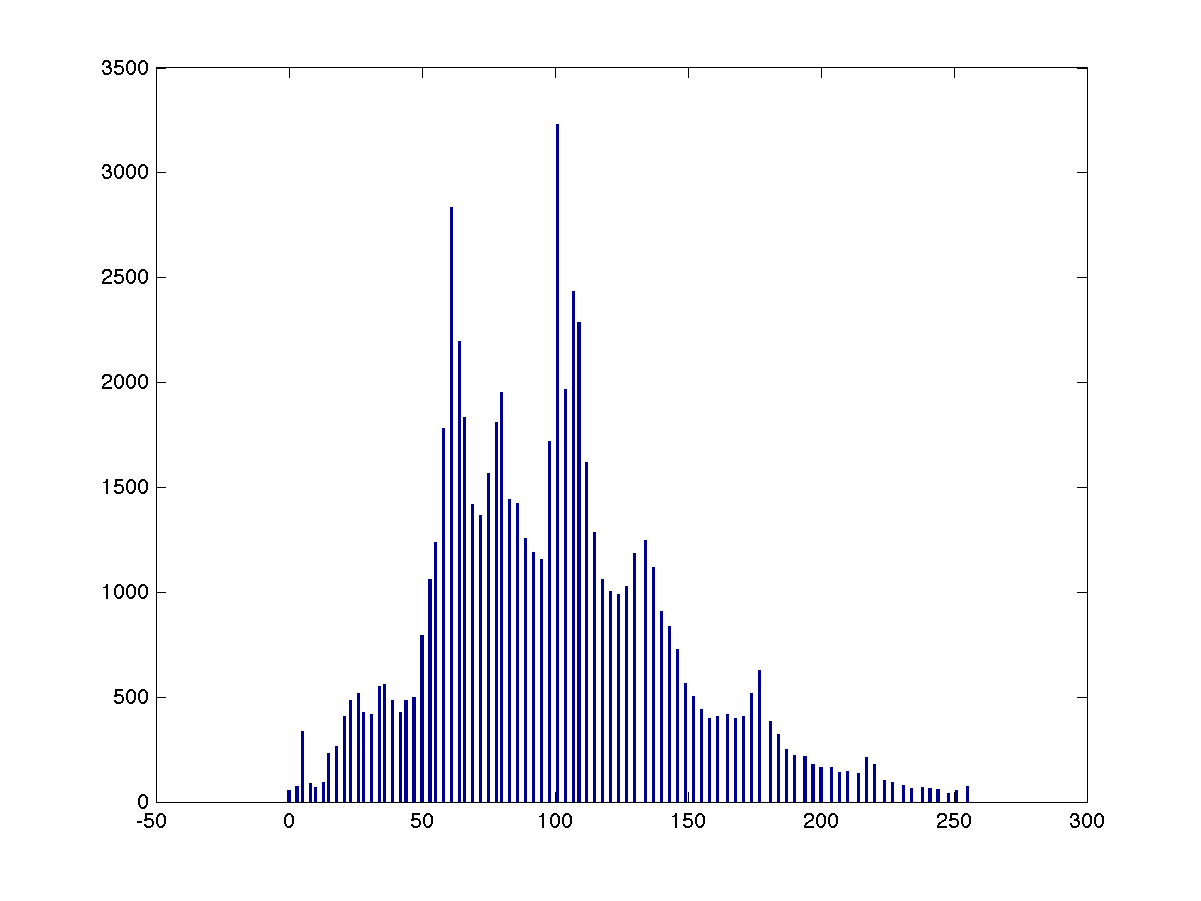


sample2.log.hist.png: The histogram of the log transformed image D

**Inverse log transform**



sample2.inv.log.png: The inverse log transformed image D



sample2.inv.log.hist.png: The histogram of the inverse log transformed image D

**Power-law transform**

**Problem 2: Noise Removal**

1. Add the same kind of noise as in sample3.raw to image I and denote the result as .
2. Add the same kind of noise as in sample4.raw to image I and the output is denoted as .
3. Choose proper filters and parameters to remove the noise in and , and denote the resultant images as and , respectively. Please specify the steps of your de-noise process and provide some discussions about the reason why those filters and parameters are chosen.

// 針對 outlier detection，畫一張橫坐標為epsilon，縱座標為PSNR的圖！

1. Compute the PSNR values of and and provide some discussions.