**Project Title:  Image Processing Fundamentals**

**Project Number**: Project 7

**Course Number**: CEG 7850-01

**Student's Name**: Alex Reigle

**Date Due**: November 12, 2021

**Date Submitted**:  November 12, 2021

**Declaration Statement:**

I hereby declare that this Report and the Matlab codes were written/prepared entirely by me based on my own work, and I have not used any material from another Project at another department/ university/college anywhere else, including Wright State. I also declare that I did not seek or receive assistance from any other person and I did not help any other person to prepare their reports or code.  The report mentions explicitly all sources of information in the reference list. I am aware of the fact that violation of these clauses is regarded as cheating and can result in invalidation of the paper with zero grade. Cheating or attempted cheating or assistance in cheating is reportable to the appropriate authority and may result in the expulsion of the student, in accordance with the University and College Policies.

**Abstract:**

In this project I developed a function to add salt and pepper noise to an image and a median filtering function to demonstrate the elimination of this noise type. I then demonstrated periodic noise reduction in an image using a Gaussian high pass notch filter. Coloring techniques are shown to highlight intensity related characteristics of an image through segmentation and pseudo coloring to distinguish dark regions in a picture and to highlight a river, respectively. Image coloring was also used to improve intensity discernment in a medical image and enhance the quality of another picture.

**Technical Discussion:**

This project implements a salt and pepper noise function that utilizes eqn. (5-16) from the textbook

where using a random number so that with a Pa = Pb = 0.2 likelihood, image pixels are assigned the maximum or minimum possible value and is unchanged otherwise. The noise is then removed by using a 3x3 median mask filter. An alternative median filter function is considered that shows the difference of filtering with a row replicated and zero padded border.

Periodic nose is observed in an image through the use of a previously developed 2D fast Fourier transformation function that has been altered for the purposes of the current problem. This alteration entails a transform function, H, that has two high pass regions instead of one, defined as:

H(i,ii) = Duv1(i,ii)\*Duv2(i,ii)

where, Duv1 and Duv2 have the same center points with offsets of negative value

where r is the radius from the considered center offset (of which there are two) and R is the cutoff frequency. A cutoff frequency of 40 was used in the images generated below.

Pseudo-Coloring was employed by sweeping through the image pixels and assigning an RGB color value to the pixel based on its grayscale intensity value compared to a threshold. For the river problem, the threshold was set to a pixel intensity of 20.

To achieve image enhancement, the RGB color vectors were separated into three distinct vectors to calculate the histogram for each. The histograms were then combined into an average to then attempt to equalize the image histogram and improve the color quality of the image.

Segmentation was used in the final problem through the selection of observation regions from the image to calculate the statistic properties of those regions of interest. Then, a cutoff value of 1.1 standard deviations from the mean was selected to exclude any image intensity value that was not sufficiently low enough. This was used to highlight the darkest regions of the image.

**Results:**

Problem 1 implemented eqn. (5-16) from the text to introduce salt and pepper noise to the orignal image to create the image in Fig.. The salt and pepper noise were generate with an equal 0.20 probability.

A picture containing text

Description automatically generated

Figure 1: Salt and pepper noise generated image.

The image was then passed through a median filter which considered the median value of the nearest nine pixels, to result in the images shown in Fig. 2-4. Notice that the boarder of figures 3 and 4 has slight differences that arise due to the different techniques used at the border.

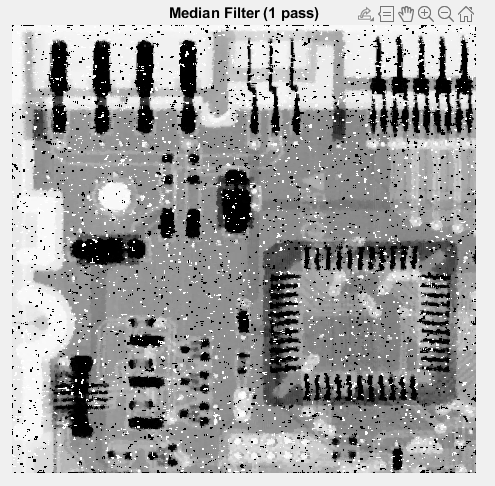


Figure 2: Resultant image from Fig. 1 after a single pass of median filter noise reduction with a zero-padded image.

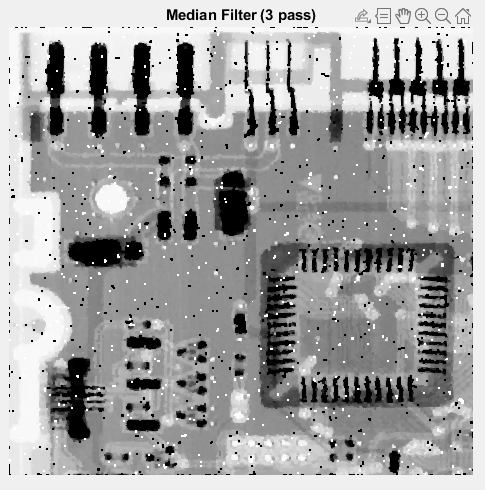


Figure 3: Resultant image from Fig. 1 after a three passes of median filter noise reduction with a zero-padded image.

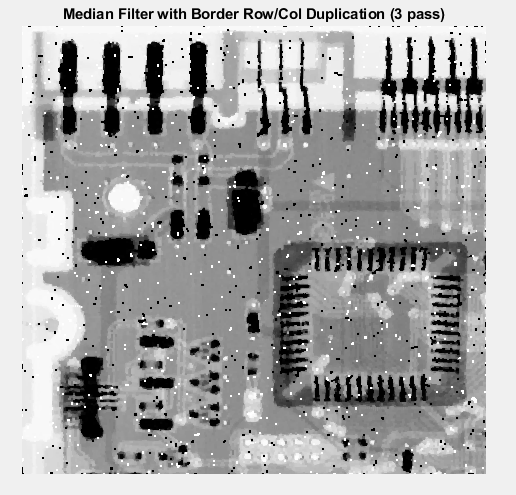


Figure 4: Resultant image from Fig. 1 after a three passes of median filter noise reduction with a row and column replicated outer boarder.

Problem 2 began with the image shown in Fig. 5, showing a very stark amount of periodic noise. This noise, due to its periodicity, can be removed in the Fourier/frequency domain.

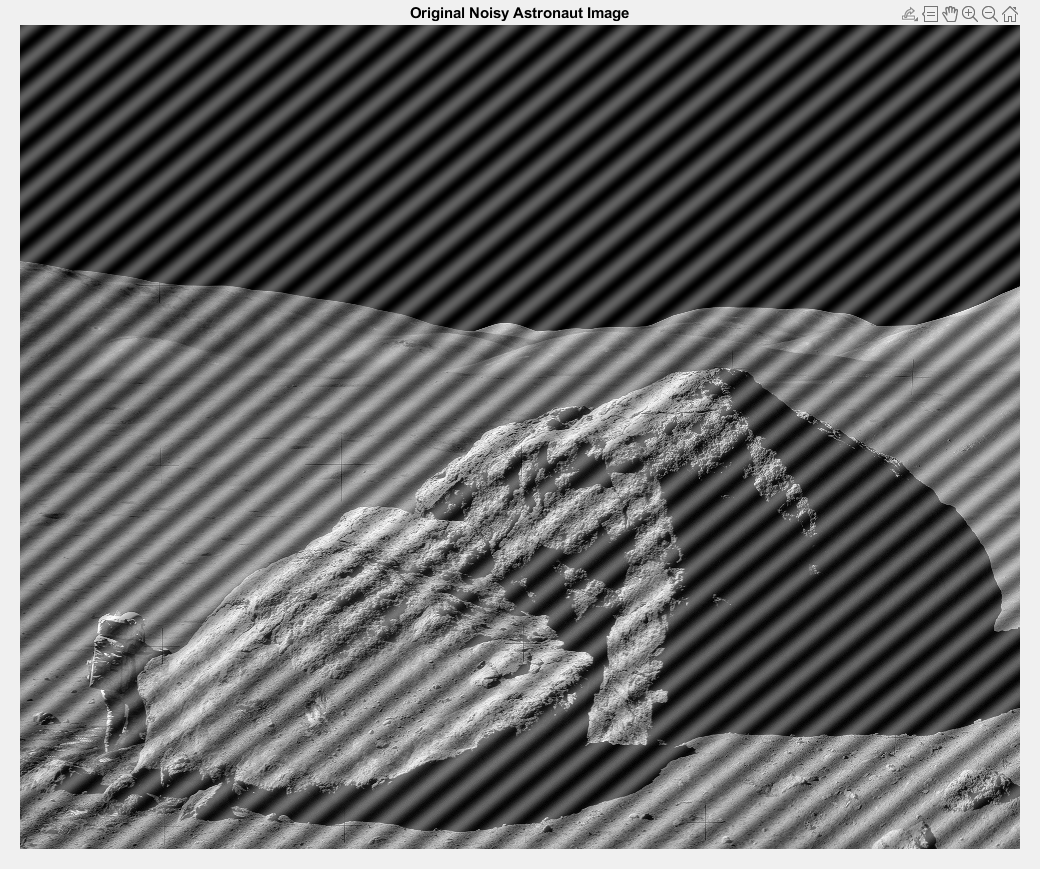


Figure 5: Noisy NASA astronaut image with periodic noise patterns.

Using a Fourier transform, and representing the resultant image in logarithmic units, it is clear to see from Fig. 6 that there are two noise sources that are contributing to the noise in the image.

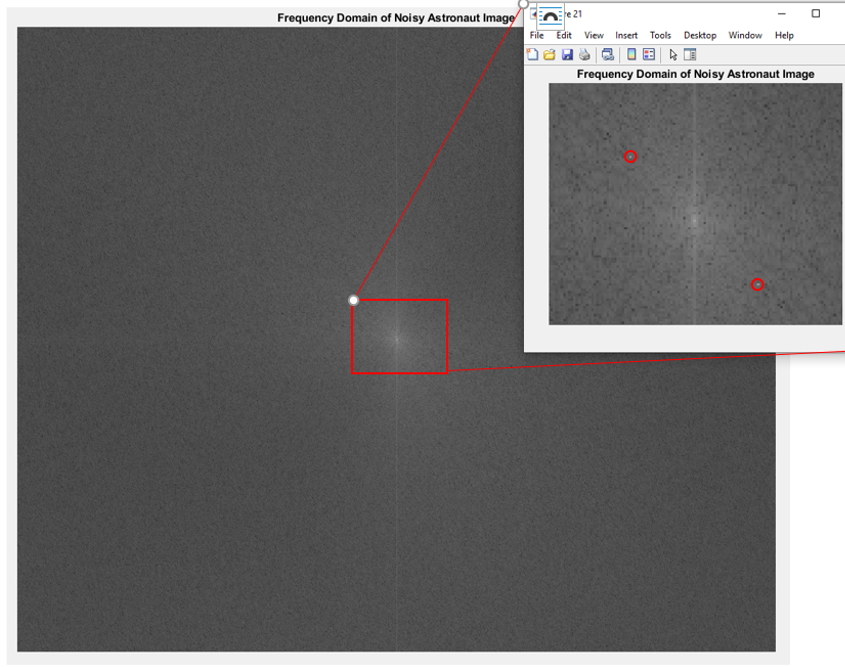


Figure 6: The frequency domain of the image in Fig. 5 is shown, with a zoomed in view of the red box to and red circles to highlight the noise artifacts.

Using a Gaussian notch filter, the noise artifacts highlighted in Fig. 6 were removed, resulting in the image shown in Fig. 7, below.

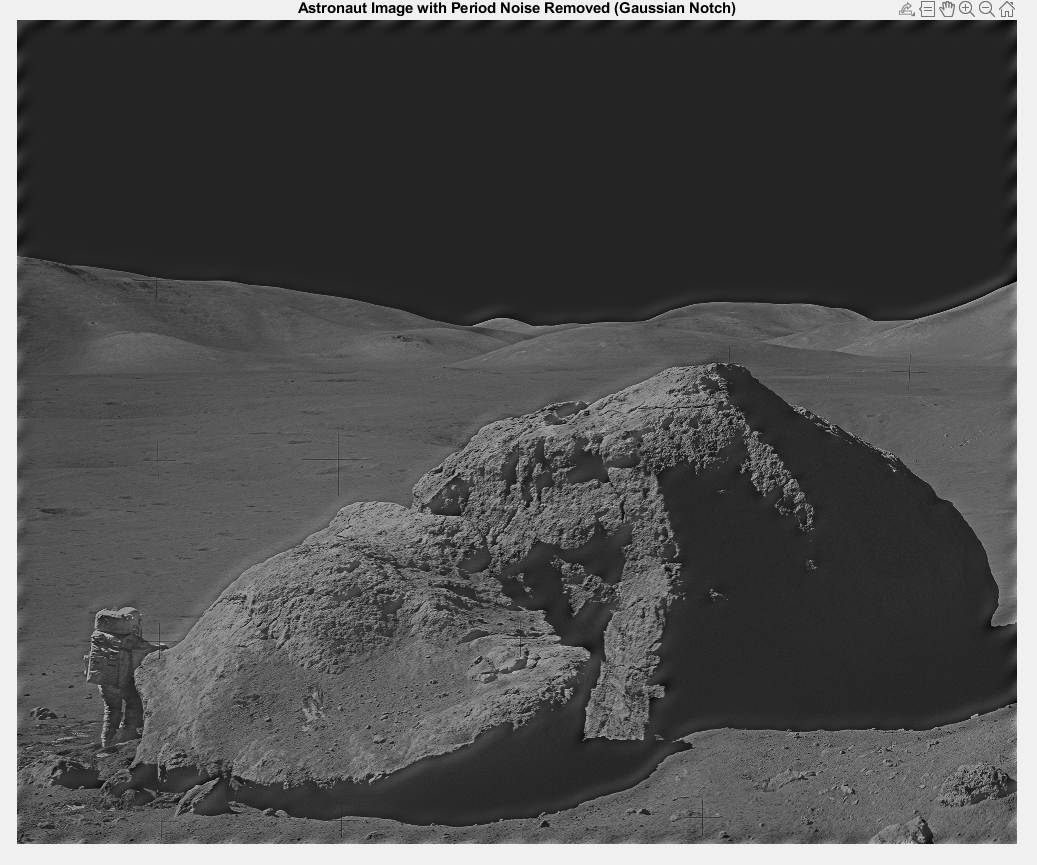


Fig. 7: The noise reduced astronaut image, after the removal of the two periodic noise artifacts.

The image shows a drastic improvement in quality from the high pass filtering technique. However, there are still remnants of the noise artifacts apparent at the image edges. These artifacts can be removed by increasing the cutoff frequency radius of the filter, but this makes the image significantly darker and of lower quality.

Problem 3 consisted of two pseudo coloring techniques. The first problem was to set a threshold level so that dark image pixels associated with water would be changed to a yellow color value and all other pixels would remain unchanged. The results of this thresholding are shown in Fig. 8.

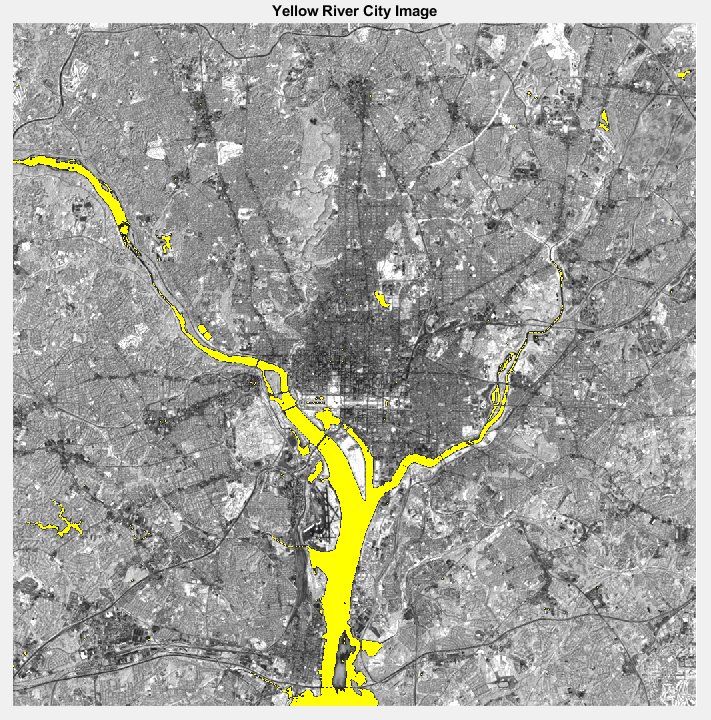


Figure 8: The result of thresholding an image so that values below the selected threshold of 20 are replaced with a color value corresponding to yellow.

Similarly, in the second part of problem 3 a medical image underwent thresholding in a similar manner to set eight color values to replicate the image from Figure 6.18(b) in the textbook. The results are shown below in Fig. 9.

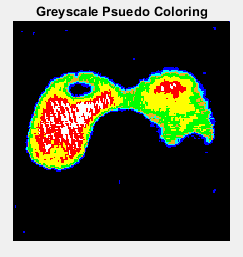


Figure 9: The pseudo color scale of the image from Figure 6.18(a) from the textbook to recreate the image in Figure 6.18(b).

Noting that Fig. 9 shows blue dots in the black region that are not seen in the book example, the complete recreation of the image with the file given is not possible. However, given a relatively low resolution image I was able to replicate the largest order features of the image from the text.

Problem 4 tasked me with recreating the image enhancement that was shown in Fig. 6.33 from the textbook. Figure 10 shows the image enhancement that were made in comparison to the original image (Fig. 10a). Figure 10b is the result of taking the mean of the RGB histogram equalization returns which are applied individually to each color vector in Fig. 10c. These images show very little improvements in image quality, but a slight enhance on the rock color is noticeable in Fig. 10b. In Fig. 10c, a large enhancement in picture quality can be seen as a result of multiplying the return values of the histeq function by a semicircular function that I modeled to represent the RGB values that are shown in the book.



Figure 10: The original image from Figure 6.33 in the text and the three corresponding processing outputs.

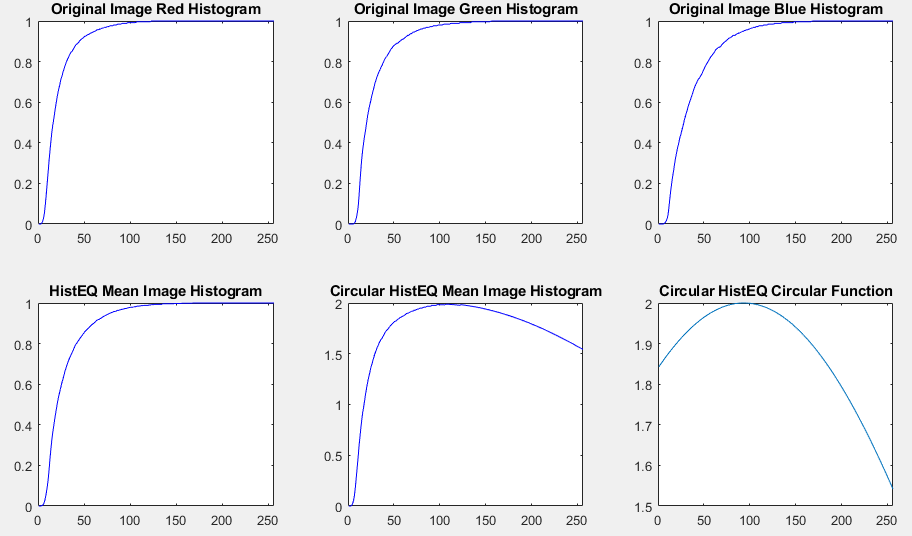


Figure 11: The histograms corresponding to the images in Fig. 10

The histograms from Fig. 11 make it clear that each method does change the color map of the image, however, in the circular case the improvement is noticeable. This is most likely due to the fact that the image histogram was multiplied by the histogram shown in the textbook, which results in a very different looking histogram function, but very similar looking results.

Problem 5 entailed image segmentation. The image was considered in four observation samples which were selected through inspection and used to calculate the standard deviation and mean of the darkest regions of the image. I then selected an N=1.1 standard deviations from the mean as the cut off threshold for a region of sufficient darkness. The resulting image is shown below in Fig. 12.

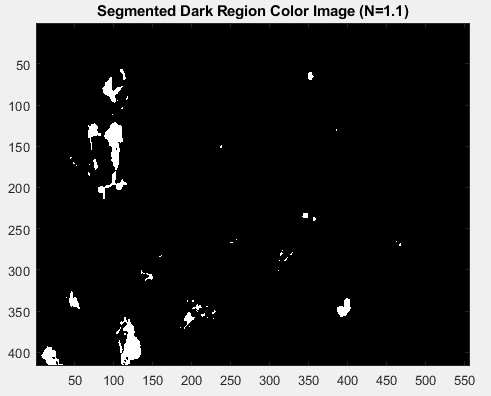


Figure 12: The segmented image resulting from Fig. 7.26 in the textbook. Here the white values correspond to the darkest values in the original image.

**Discussion of Results:**

Problem 1 shows a dramatic improvement in image quality between the first and third median filter pass. This is likely because only one-in-five pixels has been corrupted by noise and each pixel is calculated by considering nine pixels, which means it is very unlikely to highly influence the result. However, at the boundary cases this is not true. In the case where the border of the image is zero padded (to preserve image size), the pixels are much noisier than anywhere else on the image. However, by simply replicating the outermost row and column, this effect can be somewhat mitigated.

Problem 2 results show a significant difference from the image shown as the final result in the textbook. However, this image almost certainly underwent more than one type of filtering. While the center and the local area around the noise artifacts in the image are removed, the ringing is still present. To further improve the image, I would suggest subtracting a sinc function centered at each noise artifact to remove the ringing. I would also improve the brightness of the image by using a power law transform. I suspect that these or similar operations were done the image observed. I also note that there is a halo effect that occurs at low frequency areas, such as edges. This is likely also a side-effect of only removing the centroid of the noise artifact using the Gaussian filter.

Problem 3 shows the image coloring outcomes to be fairly successful. The coloring of the river and the medical picture were both employed using the same method. The medical image, however, cannot perfectly replicate the results in the text. I know this due to the blue artifacts that are visible in Fig. 9 despite still not having a low enough threshold to see the right lobe shown in the book. I suspect that this is due to the image quality.

Problem 4 shows that histogram equalization does improve the image quality. In Fig. 10, the mean histogram image has a slightly lighter coloring on the background rocks as well as more visible ripples in the water. The individual histogram equalization image shows very little, if any, improvement in quality. It appears to be darker than the original image. However, the greatest improvements to the image quality are seen in the image whose histogram was multiplied by a circular function. This image is lighter and shows better details in the foreground and background.

Problem 5 results vary greatly with the placement, size, and number of selected observations. For example, the entirety of the image can be taken as a single observation to achieve a very comparable result while requiring an N = 3 instead of 1.1. Likewise if the observations are entirely within the dark portions of the image, an N = 0 could achieve a similar effect.

**Figures:**

A picture containing text

Description automatically generated

Figure 1: Salt and pepper noise generated image.

A picture containing text

Description automatically generated

Figure 2: Resultant image from Fig. 1 after a single pass of median filter noise reduction with a zero-padded image.

A picture containing text, nature

Description automatically generated

Figure 3: Resultant image from Fig. 1 after a three passes of median filter noise reduction with a zero-padded image.

A picture containing text, nature

Description automatically generated

Figure 4: Resultant image from Fig. 1 after a three passes of median filter noise reduction with a row and column replicated outer boarder.

A picture containing text, window blind, grate

Description automatically generated

Figure 5: Noisy NASA astronaut image with periodic noise patterns.

Graphical user interface, application

Description automatically generated

Figure 6: The frequency domain of the image in Fig. 5 is shown, with a zoomed in view of the red box to and red circles to highlight the noise artifacts.

A picture containing calendar

Description automatically generated

Fig. 7: The noise reduced astronaut image, after the removal of the two periodic noise artifacts.

Map

Description automatically generated

Figure 8: The result of thresholding an image so that values below the selected threshold of 20 are replaced with a color value corresponding to yellow.

A picture containing text, vector graphics

Description automatically generated

Figure 9: The pseudo color scale of the image from Figure 6.18(a) from the textbook to recreate the image in Figure 6.18(b).

Graphical user interface, application, timeline, PowerPoint

Description automatically generated

Figure 10: The original image from Figure 6.33 in the text and the three corresponding processing outputs.

Chart

Description automatically generated

Figure 11: The histograms corresponding to the images in Fig. 10

A picture containing diagram

Description automatically generated

Figure 12: The segmented image resulting from Fig. 7.26 in the textbook. Here the white values correspond to the darkest values in the original image.

**Written Programs:**

* main
* addSaltPepper
* median3x3Filter
* ALTmedian3x3Filter
* fft2dFilter

**Utilized Programs:**

* subplot
* imshow
* cast
* imread
* blockproc
* sort
* length
* mean
* median
* std
* max
* min
* abs
* zeros
* size
* plot
* reshape
* nextpow2
* meshgrid
* fft2
* ifft2
* rgbplot
* linspace
* pwd
* imagesc
* sqrt
* imag
* real
* imhist
* imhist
* histeq
* xlim
* sin
* cos
* rand