Simple Image Transformations

Solve the following tasks using Matlab:

1.Display the test images from the module webpage. Reduce the intensity resolution (number of quantization levels) — which effects do you observe?

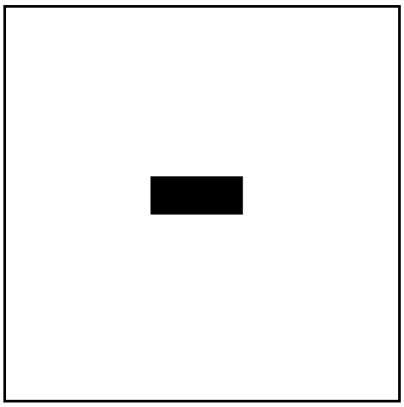
2.Display the histogram for the test image `boats.gif`. Enhance the contrast by histogram equalization. Implement your own function and compare it (speed, appearance, histogram) to the built-in Matlab function.

3.Try to outline simple uniformly colored regions by intensity thresholding (gray-level slicing). Use the simple test image `volcanonoise.jpg`. Remove noise by by median filtering (hint: use `medfilt2`) the `volcanonoise.jpg` image. Experiment with combinations of noise removal and intensity thresholding segmentation. Is the noise removal effective enough to permit simple segmentation?

Spatial Filtering and Fourier Analysis

1.Implement an averaging filter in Matlab. Use the `cameraman.jpg` standard test image for displaying the result of applying the filter. Do not use the built-in Matlab filter functions (such as `filter2` and `conv2`). Let the size of the filter be a parameter for a function that does the filtering. Measure the run-time for the function when the filter size is varied. Plot the run-times (possibly using `tic` and `toc`).

2.Create the Fourier Transformation of the box image illustrated below.



Vary the size and aspect ratio of the box, and observe the changes in the FT. Hint: Check the help in Matlab for the `fft2` and `fftshift` functions.

3.Define a Matlab function that given the image dimensions will create in the frequency domain and ideal low pass filter. Implement a Matlab function that given an image and this filter mask, performs the filtering operation and returns the filtered image. Now, experiment with this function for some of the test images (in particular, `cameraman.jpg`, `AuPsSn40.jpg` and `tree_rings.jpg`) - change the radius of the mask (ie filter cut-off).