Homework- III 3 (i)

1. Generate a 256×256 matrix $\mathbf{A}$ of outcomes of a continuous amplitude Gaussian random variable with $\mu = 2$ and $\sigma^2 = 3$. Calculate its sample mean and variance. Note that $\mathbf{A}$ is not an image matrix. Normalize $\mathbf{A}$ to obtain $\mathbf{B}$. Calculate the sample mean, variance, probability mass function as well as the histogram of $\mathbf{B}$. Show $\mathbf{B}$ and all calculated quantities.

2. Using histogram modification, modify *your image* so that the resulting image has a histogram that matches $h_B(l)$ as in 1 above. Show your image, the modified image, their histograms and the matching point function. Briefly compare the modified image's histogram to $h_B(l)$.

3. Do the processing I did in the "undoing example" on your image.

4. Uniform quantize your image using $\Delta = 4, 8, 16, 64$. Show the quantized image, its histogram and MSQE in each case.

5. Compand your image using $\Delta = 4, 8, 16, 64$. Show the quantized image, its histogram and MSQE in each case. Compare the results to those obtained in 4.

6. $p_A(l) = [.1, 0, .3, .2, 0, 0, .3, .1]$ and $p_B(k) = [.2, 0, 0, .1, .4, .3]$ for two *images* $\mathbf{A}$ and $\mathbf{B}$. Calculate a point function $g(l)$ such that $C(i,j) = g(A(i,j))$ has histogram $h_C(l)$ that "matches" $h_B(l)$. Assume all images have a total of 10 pixels. Calculate the histogram $h_C(l)$.

Homework- III(ii)

1. Implement the Lloyd-Max quantizer for your image. Do everything I did between pages 9-13. Show all results as well as your original image and its sample probability mass function.

2. Convolve the 2-d sequences with nonzero portions shown, i.e., obtain $\mathbf{C} = \mathbf{A} \otimes \mathbf{B}$. Show your computations graphically for at least 5 different values of $\mathbf{C}$ similar to the earlier example.

$A(i,j)=$

	$j=0$	1	2
$i=0$	0	4	4
1	-2	1	3
2	5	1	1

, $B(i,j)=$

	$j=0$	1	2
$i=0$	-1	3	4
1	1	2	2
2	-3	2	6

3. Implement a convolution script in matlab taking note of the computational simplifications as discussed. Convolve your image with *itself*. Normalize the result and show it as an image. Comment on the result as well as the execution time. What is the dimension of the resulting image? Now do the same using the *conv*2 command in matlab. Make sure the results of your script and *conv*2 are the same. (You can do so by calculating an error image and summing its absolute value. The result should be 0. Experiment with small images till you get it right.)

4. Let $A(i,j) =$

	$j=0$	1	2
$i=0$	-1	0	1
1	-2	0	2
2	-1	0	1

Convolve your image with $\mathbf{A}$ (your script). Take the absolute value of the result, normalize and show. Comment on the result and the execution time. What is the size of the resulting image? Now do the same with *conv*2 as above.