ECE 253, Homework 5 Solutions

1) Scalar Quantization

(a) The centroid condition says that b should be the centroid of the probability that is in that region. Looking at positive x

$$b = \frac{\int_a^\infty x \frac{\lambda}{2} e^{-\lambda x} dx}{\int_a^\infty \frac{\lambda}{2} e^{-\lambda x}}$$

After eliminating the $\lambda/2$ on top and bottom, the numerator is the expression you were given in the problem statement with $c = a$, and the denominator is $(e^{-\lambda a})/\lambda$. So we end up with:

$$b = a + \frac{1}{\lambda}$$

(b) The other Lloyd condition says that the decision level must be halfway between the reconstruction levels. So: $a = b/2$. Therefore, the 3-level quantizer is characterized by

$$a = \frac{1}{\lambda} \qquad \text{and} \qquad b = \frac{2}{\lambda}$$

2) Lloyd Algorithm for Quantizer Design

A 3-level quantizer is to be designed to minimize mean squared error using the Lloyd algorithm with the training set

$$T = \{1, 2, 3, 4, 8, 9, 12\}$$

With the initial codebook $\mathcal{C}_1 = \{2.0, 6.0, 10.0\}$ we get the first set of decision points:

$$\mathcal{D}_1 = \left\{ \frac{2+6}{2}, \frac{6+10}{2} \right\} = \{4, 8\}$$

Here we have to enunciate a tie-breaking rule, since two of the training points lie exactly on the decision boundaries. We can arbitrarily use the rule that the points belong to the region to the left, or use the rule that they belong to the region to the right. Tie-breaking is not something we have to worry about in the case of continuous pdf's. If we assign points to the left, we get the following new codebooks and partitions:

$$\mathcal{C}_2 = \left\{ \frac{1+2+3+4}{4}, 8, \frac{9+12}{2} \right\} = \{2.5, 8, 10.5\}$$

$$\mathcal{D}_2 = \{5.25, 9.25\}$$

$$\mathcal{C}_3 = \{2.5, 8.5, 12\}$$

$$\mathcal{D}_3 = \{5.5, 10.25\}$$

and this last pair represents convergence. Whereas, if we adopted the rule of assigning to the right, we would get:

$$\mathcal{C}_2 = \left\{\frac{1+2+3}{3}, 4, \frac{8+9+12}{3}\right\} = \{2, 4, 9.7\}$$

$$\mathcal{D}_2 = \{3, 6.8\}$$

$$\mathcal{C}_3 = \{1.5, 3.5, 9.7\}$$

$$\mathcal{D}_3 = \{2.5, 6.6\}$$

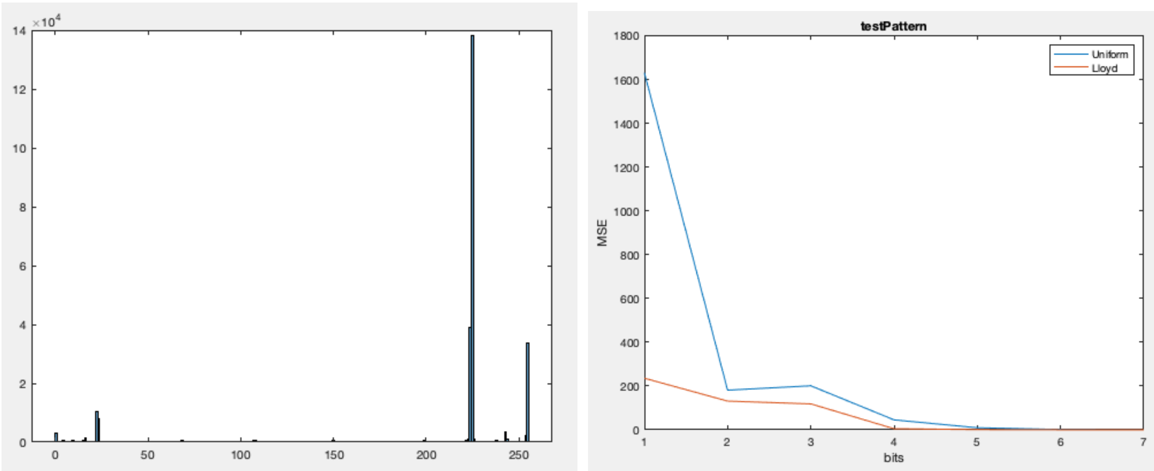
3) **Comparing quantization for images**

a) Here is a function that takes as inputs an 8-bit image and a scalar $s \in [1, 7]$ and performs uniform quantization so that the output is quantized to a s-bit image:

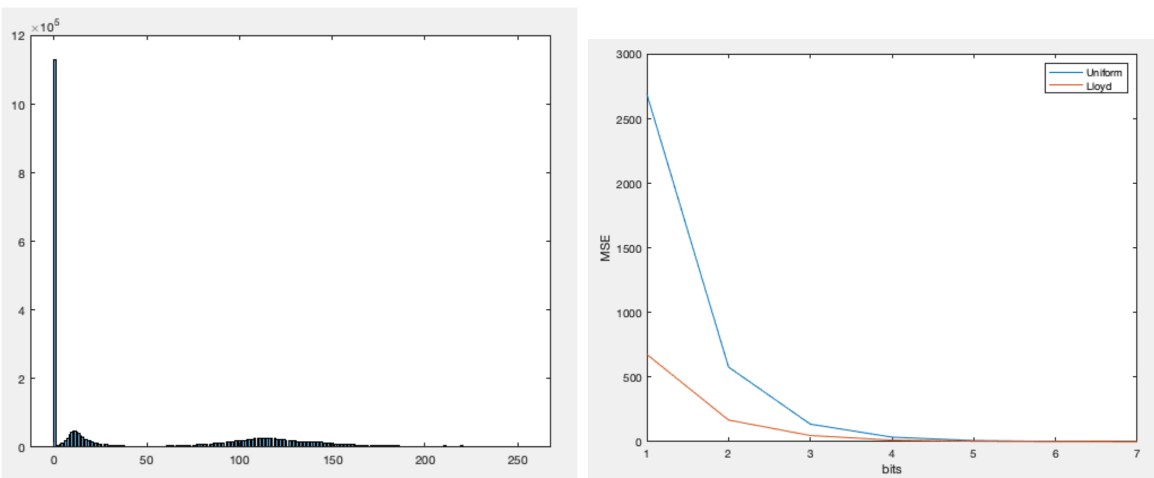
```
function [out] = uniform_quant(Im,bits)
index = 256/(2^bits);
P = 0:index:256;
P1 = P(1:end-1);
P2 = P(2:end);
Y = mean([P1;P2]);
out = Im;
for i = 1:(2^bits)
out((Im >= P(i)) & (Im < P(i+1))) = Y(i);
end
end
```

Here, Im is a grayscale image from 0-255, bits is in the range of 1 to 7, P is a vector of partitions, and Y is the codebook.

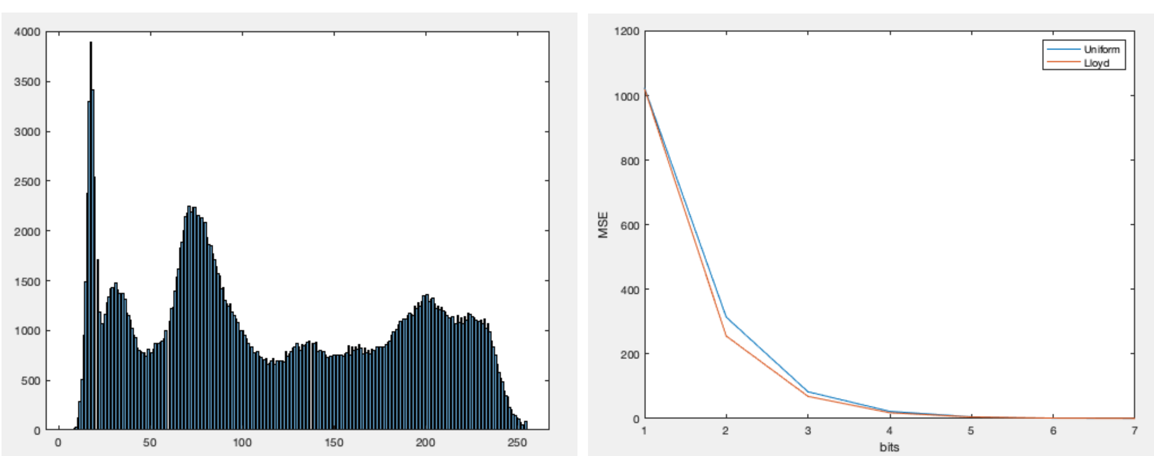
b) Here are the histogram and quantizer plot for the testpattern512 image:



Here they are for astronaut:



and for vase:



The Lloyd quantizer does better in general because it is trying to place the quantization levels optimally, where the probability mass function is large, whereas the uniform quantizer just spaces the quantization levels out evenly, regardless of the distribution.

The performance ratio is smallest for the vase image, because we can see that vase is by far the closest to having a uniform histogram. For a truly uniform histogram, uniform quantization and optimal quantization would be the same thing.

The performance ratio is largest for the test pattern, because that is an artificial image that has very few histogram bins occupied. The Lloyd quantizer is able to place quantization levels precisely where the occupied bins are.

Here is the code for making the quantization comparison:

```
function [uniform_mse,lloyd_mse] = compare(Im)
[N,M] = size(Im);
training = reshape(Im,N*M,1);
uniform_mse = zeros(1,7);
lloyd_mse = zeros(1,7);
for i = 1:7
UQ = uniform_quant(Im,i);
uniform_mse(i) = immse(Im,UQ);
[P,Y] = lloyds(double(training),2^(i));
LQ = apply_quant(round(P),round(Y),Im,i);
lloyd_mse(i) = immse(Im,LQ);
end
end

function [out] = apply_quant(P,Y,Im,bits)
out = Im;
P = [-1 P 256];
for i = 1:(2^bits)
out((Im > P(i)) & (Im <= P(i+1))) = Y(i);
end
end
```

c) Lastly we consider whether quantizers can get worse distortion with increasing bit rate.
- For the test pattern, the uniform quantization happens to do worse when the bit rate is increased from 2 bits per pixel to 3. A 2-bit uniform quantizer places the quantization levels at 32, 96, 160, and 224. The background of the test pattern happens to have a value of 224-225, so the background is represented with extremely low MSE by this quantizer. The 3-bit quantizer moves off of these particular values, so it actually does a little bit worse.

- Although for these images we don't see the Lloyd quantizer getting worse with increasing bit rate, it is possible that this could happen with the Lloyd algorithm because the iterative process can fall into a local minimum, and it is possible that it could be a relatively poor local minimum for a quantizer with N levels, and could be an excellent local minimum or be the global minimum for a quantizier with fewer levels.