EL5123 MidTerm Solution (2013 Fall)

Q1:

(a) The human being perceives color through different types of cones in the retina, sensitive to different parts of the visible spectrum, corresponding to red, green and blue colors primarily. The human brain fuses the received signals from these cones and gives sensations of different colors depending on the combinations of the responses from these cones.

(b) The color mixing theory tells us that any color can be obtained by mixing three primary colors in an appropriate proportion.

(c) To capture a color image, we use different sensors that are sensitive to each of three primary colors.

(d) The three attributes of a color are: intensity (or luminance), hue and saturation.

(e) One reason is to enable us to process (modify, enhance, etc.) the luminance and chorominance components separately to achieve the desired effects. For example, we may do equalization on the luminance only to enhance the image contrast. We may change the chromiance components in a certain way to achieve a desired color altering effect. Another reason is to enable more efficient representation of the image to reduce the cost for transmission or storage. This is because the chrominance components have narrower bandwidth so that we can sample it at a lower rate, and that the human eye is less sensitive to the chrominzance components so that we can use fewer bits to specify each chrominance sample.

Q2:

Solution:

(a) Uniform quantizer: Stepsize=256/3=85.33; decision levels: 0, 85.33, 170.67, 256; reconstruction levels: 42.67, 128, 213.33. And

$$MSE = 2\left(\int_{0}^{85.33}(f-42.67)^2\, p(f)df + \int_{85.33}^{128}(f-128)^2\, p(f)df\right)$$

where

$$p(f) = 1/128 \quad - f/128^2$$

Note: 2 pts for decision levels, 2 pts for reconstruction levels, 1 pt for MSE (if equation is correct).

(b) Non-uniform quantizer:

Based on the symmetry with respect to the center 128, the decision levels can be represented as: *0, b, 256-b, 256*; reconstruction levels can be represented as *a, 128, 256-a*.

Using the optimality condition for minimal MSE quantizer, we can derive the following relations:

$$Nearest\ Neighbor\ Condition: \quad b = \frac{a+128}{2};$$

$$Centroid\ condition: \ a = \frac{\int_0^b f\,p(f)df}{\int_0^b p(f)df} = \frac{\frac{1}{128}\int_0^b f\left(1-\frac{f}{128}\right)df}{\frac{1}{128}\int_0^b \left(1-\frac{f}{128}\right)df} = \frac{384b-2b^2}{768-3b}.$$

Note that the nearest neighbor condition implies that the decision level between two reconstruction levels should be in the midway between the two reconstruction levels.

Solving the above two equations together yields:

$$a = 128(2-\sqrt{3})=34.3,\ b=64(3-\sqrt{3})=81.1$$

As a rough check, because f is more concentrated near the lower values in the first half, you should expect that with the MMSE quantizer, a and b should both be lower than their corresponding values in the uniform quantizer, which are 42.67 and 85.33, respectively.

The MSE is

$$MSE = 2\left(\int_0^b (f-a)^2\,p(f)df + \int_b^{128} (f-128)^2\,p(f)df\right)$$

Many students wrote the MSE as a function of decision levels and reconstruction levels, and then took derivative of MSE with respect to the decision levels and reconstruction levels, to derive the equations that the decision levels and reconstruction levels must satisfy. When you did every thing right, you should get to the nearest neighbor condition and centroid condition. If you correctly identified that, based on symmetry, you only need to determine two unknowns, and you derived the relations (by setting derivatives to zero) that these two unknowns must satisfy, you get most of the points. If you derived the general relations without making use of the symmetry, you get 4 points.

Q3:

Original Image				
Gray level f	0	1	2	3
Histogram h(f)	6/16	5/16	3/16	2/16

(5 Points. 1 For Each)

f_A	$p_F(A)$	$C_F(A)$		$C_F(B)$	$p_F(B)$	f_B
0	6/16	6/16		2/16	2/16	0
1	5/16	11/16		7/16	5/16	1
2	3/16	14/16		14/16	7/16	2
3	2/16	16/16		16/16	2/16	3

Transform Image				
Gray level f	0	1	2	3
Histogram h(f)	0/16	6/16	8/16	2/16

(5 Points. 1 For Each)

1	1	2	3
1	2	2	3
2	2	2	1
2	2	1	1

(5 Points.)

Note that for the most accurate results, you should try to match the CDF of A and CDF of B, when trying to find the mapping between A and B. Some students converted the CDF to image values by multiplying the CDF by 3 and rounding the result to nearest integers; and then find the mapping between A and B based on these rounded integer values. If you did this way correctly, 2pt will be deducted.

Q4:

(a) Not separable(1 Point). The functionality of this filter is **Edge Detection, because the coefficients sum to 0**. (1 Point)

(b) Separable(1 Point). Horizontal Filter is [1 2 1]; Vertical Filter is $[1\ 2\ 1]^T$. The functionality of this filter is **Image Smoothing, because the coefficients sum to 1 and all coefficients are positive**. (1 Point)

(c) $H_2(x,y) = \frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} = \frac{1}{16}\begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}\begin{bmatrix} 1 & 2 & 1 \end{bmatrix}$

$$Hy(v) = F\{h_y\} = (1 \cdot e^{-j2\pi v(-1)} + 2 \cdot e^{-j2\pi v(0)} + 1 \cdot e^{-j2\pi v(1)} = 2 + e^{j2\pi v} + e^{-j2\pi v})/4$$
$$= 0.5 + 0.5cos2\pi v$$

(2 Points)

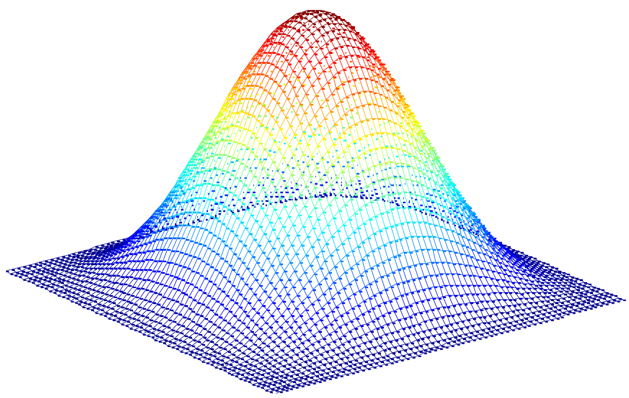
$$Hx(u) = F\{h_x\} = (1 \cdot e^{-j2\pi u(-1)} + 2 \cdot e^{-j2\pi u(0)} + 1 \cdot e^{-j2\pi u(1)} = 2 + e^{j2\pi u} + e^{-j2\pi u})/4$$
$$= 0.5 + 0.5cos2\pi u$$

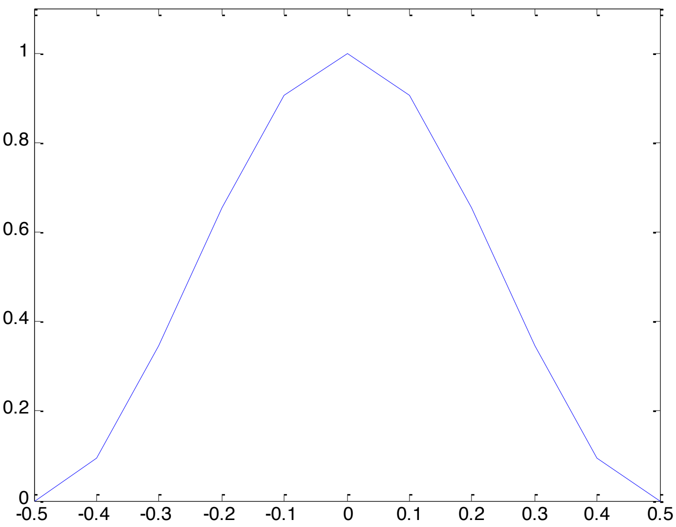
(2 Points) Note that because horizontal filter is the same as the vertical filter, you don't have to derive this. You could just use H_x(u)=H_y(u).

$$H(u,v) = \frac{1}{16}H(u)H(v) = \frac{1}{16}(2+2\cos 2\pi u)(2+2\cos 2\pi v) = \frac{1}{4}(1+\cos 2\pi u)(1+\cos 2\pi v)$$

(2 Points)

(d) The curve should be approximately like the following: (3 Points) (You should mark center with amplitude 1, locations of zeros when $u, v = \pm 0.5$. The range of horizontal/vertical frequency is from -0.5 to 0.5. If the shape is correct, but the actual position where it goes to zero is wrong, then you get 2pts. If you only draw the 1D plot correctly as below, you get 2pts.





(e) Based on the curve and/or 1D projection, the filter is a low-pass in both u and v direction. (2 Points)

Q5:

(a) $(\mathrm{M}+\mathrm{K}-1)\times(\mathrm{N}+\mathrm{L}-1)$ (2 Points)

(b) $(\mathrm{M}+\mathrm{K}-1)\times(\mathrm{N}+\mathrm{L}-1)\times KL$ (2 Points)

(c) Assuming the filter is separable, horizontal filtering for each row requires $(N+L-1)L$ multiplications. Since you have to apply this operation to M rows, the total operation for horizontal filtering is $M(N+L-1)L$. This will result an intermediate image of size $M(N+L-1)$. Next you apply vertical filtering to each column. Vertical filtering for each column requires *(M+K-1)K*, and applying it to all *(N+L-1)* columns requires $(M+K-1)(N+L-1)K$.

The overall number of multiplications is

$$M(N+L-1)L+(M+K-1)(N+L-1)K=(N+L-1)(ML+MK+K^2-K)$$ (3 Points)

OR: if you do vertical filtering first and then horizontal filtering, you should have

$$(M+K-1)(NK+NL+L^2-L)$$

(d) $S \geq M+K-1, T \geq \mathrm{N}+\mathrm{L}-1$(2 Points)

(e) 1D-DFT to each row requires $T\log_2 T$. 1D-DFT on each of the S rows requires $S\cdot T\log_2 T$ multiplications. Then another 1D-DFT to each column of intermediate image requires $T \cdot S \log_2 S$. Thus, the total number of multiplications is

$$ST \cdot (\log_2 S + \log_2 T) = ST \log_2 ST \quad \text{(3 Points)}$$

(f) DFT of original image costs $ST \cdot \log_2 ST$ multiplications. IDFT costs $ST \cdot \log_2 ST$ multiplications. Frequency domain multiplications between DFT[Image] and DFT[Filter] cost $S \cdot T$. Thus, the total number of multiplications required to implement the filtering using the FFT is $2ST \cdot \log_2 ST + ST$ (3 Points)

Q6:

(a) (2pt) Recall that for N-pt DFT, the center frequency indices around *N/2* are high frequency, the low frequency indices near 0 and high indices near *N-1* are low frequency. Because *P(k)* is non-zero only for *k=0,1,255*, *p(k)* is a low pass 1D filter. *H(k,l)* is a low pass 2D filter, passing only very low frequency. Therefore, the image *g(m,n)* should be a severely blurred version of the original image.

(b) (4pt) Because *H(k,l)* is separable, the corresponding spatial domain filter *h(m,n)* is also separable and can be written as *h(m,n)=p(m) p(n),* where *p(m)* is the inverse DFT of *P(k):*

$$p(n) = IDFT\big(P(k)\big) = \frac{1}{\sqrt{256}}\left(1 + e^{\frac{j2\pi n}{256}} + e^{j2\pi\, 255\frac{n}{256}}\right)$$
$$= \frac{1}{\sqrt{256}}\left(1 + e^{\frac{j2\pi n}{256}} + e^{-j2\pi\frac{n}{256}}\right)$$
$$= \frac{1}{\sqrt{256}}\left(1 + 2\cos\frac{2\pi n}{256}\right)$$

$$\textit{Note that in the above we made use of the fact that}$$

$$e^{j2\pi\, 255\frac{n}{256}} = e^{j2\pi\,(256-1)\frac{n}{256}} = e^{j2\pi n}e^{-j2\pi\frac{n}{256}} = e^{\frac{-j2\pi n}{256}}$$

$$Therefore\!:\ h(m,n) = p(m)p(n) = \frac{1}{256}\left(1 + 2\cos\frac{2\pi m}{256}\right)\left(1 + 2\cos\frac{2\pi n}{256}\right)$$

Some of you performed 2D IDFT on the original filter *H(k,l)* directly. As long as your calculation is correct, you get the full point. If your original equation is correct, but you were not able to simplify the result correctly, you get partial credits.

(c) *g(m,n)* is the 256x256-point circular convolution of *h(m,n)* and *f(m,n)*.

(d) *z(m,n)* has a size of 511x511. If the origin of the filter *h(m,n)* is at the top left corner, then the flipped version of *h(m,n)* will fit inside *f(m,n)* completely only when the filter origin is shifted to the lower right corner, i.e. *m*=*n*=255, of the original image. That is, *z(m,n)* and *g(m,n)* are equal only at the point *m=255* and *n=255*. Note that here we assume that the filtered image *z(m,n)* has a range of 0 to 511. Therefore, the point (256,256) is at the center of the *z(m,n)* image.

If you assume the origin of the filter is at its center, e.g. m=127,n=127, then the flipped version of *h(m,n)* will fit inside *f(m,n)* completely only when the filter origin is shifted to the point m=128, n=128. That is, *z(m,n)* at the point m=128 and n=128 equal to g(m,n) at point (255,255). Note that here we assume that the filtered image *z(m,n)* has a range of

-128 to 255+128=383. Therefore, the point (128,128) is still at the center of the *z(m,n)* image.

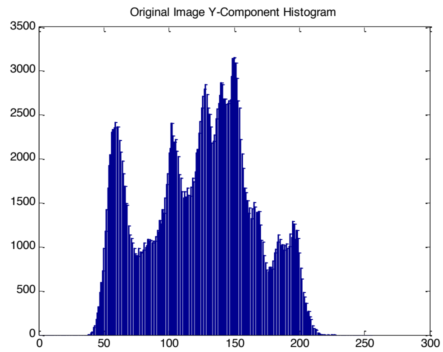
If you gave a correct answer based on either assumption, you get the full point.

Q7:

```
% Read in image and Display image -- 2pt
image = imread('lena_c.tiff');
figure; imshow(image);axis off;title('Original Image');

% Convert to YCbCr Format -- 1pt
image_YCbCr = RGB2YCBCR(image);

% Y Component Extraction -- 1pt
image_Y = image_YCbCr(:,:,1);

% Y image histogram calculation -- 3pt
h=zeros(256,1);
for l=0:255,
    h(l+1)=sum(sum(image_Y==l));
end

% Histogram Plot -- 1pt
figure; bar(h);title('Original Image Y-Component Histogram');

% Computing the transformation function -- 3pt
[Row Col] = size(image_Y);
H = h/(Row*Col);
for k=1:256
    C(k) = uint8(sum(H(1:k))*255);
end

% Perform mapping / Apply the transformation function on Y image -- 2pt
histeq_image_Y = C(image_Y);

% Y image histogram calculation -- 3pt
hh=zeros(256,1);
for l=0:255,
    hh(l+1)=sum(sum(histeq_image_Y==l));
end

% Histogram Plot -- 1pt
figure; bar(hh);title('Equalized Image Y-Component Histogram');

% Insert enhanced Y image -- 1pt
```

```
output_image_YCbCr = image_YCbCr;
output_image_YCbCr(:,:,1) = histeq_image_Y;

% Convert to RGB format and Display image -- 1pt
output_image = YCBCR2RGB(output_image_YCbCr);
figure; imshow(output_image);axis off;title('Enhanced Image');

% Save enhanced image -- 1pt
imwrite(output_image,'outimg.jpg');
```

Original Image



Enhanced Image