Polytechnic Institute of NYU Fall 2012
EL5123/BE6223 --- DIGITAL IMAGE PROCESSING Yao Wang

Midterm Exam (10/24, 3:00-5:30PM)
Closed book, 1 sheet of notes (double sided) allowed.
No peeking into neighbors or unauthorized notes. Cheating will result in getting an F on the course.
Write your answers on this problem sheet for problems where space is provided. Otherwise write you answer in the blue book.
Submit both the blue book and this problem set.

Your name: ____SOLUTION___________________________________

1. (10pt) Briefly answer the following questions: a) how does the human visual system perceives color? (3pt) b) What does the color mixing theory tells us? (2pt) How do we use this theory to capture color images (1pt), display color images (1pt), and print color images (1pt), respectively? c) What are the three attributes of a color? (2pt)

(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. To capture a color image, we use different sensors that are sensitive to each of three primary colors; To display a color image, we excite three types of phosphors at each screen location that emits three separate primary colors; To print a color image, we use three types of color inks. For capture and display, we use primary colors for illuminating light sources, mainly red, green and blue; For printing, we use primary colors for reflecting ligh sources, namely cyan, magenta, and yellow.
(c) The three attributes of a color are: intensity (or luminance), hue and saturation.

2. (15 pt) A conventional color image using the RGB coordinate requires 8 bits per color component or 24 bits per pixel. One way to reduce the bit requirement is by converting the RGB to YCbCr representation, and representing Cb and Cr components using fewer bits than the Y component, because the human eye is less sensitive to the chrominance. Suppose we use 6 bits for Y, and 3 bits each for the Cb and Cr components respectively, and use uniform quantizer in the range of 0-256 for each component. (a) What is the total number of bits per pixel? (1pt) (b) Illustrate the quantizer function for the Y and Cb component respectively. (2pt+2pt) (c) Assume a pixel has the following RGB values: R=205, G=113, B=81. What are the corresponding Y, Cb, and Cr values? (2pt) (d) What are the quantized Y, Cb, and Cr values? (2+2+2pt) (e) What are the reconstructed R,G,B values? The RGB to YCbCr conversion matrix and the inverse conversion matrix are given below. (2pt)

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1.164 & 0.000 & 1.596 \\ 1.164 & -0.392 & -0.813 \\ 1.164 & 2.017 & 0.000 \end{bmatrix} \begin{bmatrix} Y-16 \\ C_b-128 \\ C_r-128 \end{bmatrix} \qquad \begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix} = \begin{bmatrix} 0.257 & 0.504 & 0.098 \\ -0.148 & -0.291 & 0.439 \\ 0.439 & -0.368 & -0.071 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix}$$

(a) Total number of bits per pixel = 6 + 3 + 3 = 12 bits. (1pt)
(b) The number of reconstruction levels of 6 bits uniform quantizer is 2^6=64. Quantization interval q = (256 – 0)/64 = 4.

$$Q(Y) = \left\lfloor \frac{Y-Y_{min}}{q} \right\rfloor \times q + \frac{q}{2} + Y_{min} = \left\lfloor \frac{Y}{4} \right\rfloor \times 4 + 2 \quad \text{(2pt)}$$

The number of reconstruction levels of 3 bits uniform quantizer is 2^3=8. Quantization interval q = (256 – 0)/8 = 32.

$$Q(Cb) = \left\lfloor \frac{Cb-Cb_{min}}{q} \right\rfloor \times q + \frac{q}{2} + Cb_{min} = \left\lfloor \frac{Cb}{32} \right\rfloor \times 32 + 16 \quad \text{(2pt)}$$

(c) The YCbCr valuses corresponding to the given RGB values can be obtained by

$$\begin{bmatrix} 0.257 & 0.504 & 0.098 \\ -0.148 & -0.291 & 0.439 \\ 0.439 & -0.368 & -0.071 \end{bmatrix} \begin{bmatrix} 205 \\ 113 \\ 81 \end{bmatrix} + \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} = \begin{bmatrix} 133.575 \\ 100.336 \\ 170.66 \end{bmatrix} \quad \text{(2pt)}$$

(d) The quantized YCbCr values are

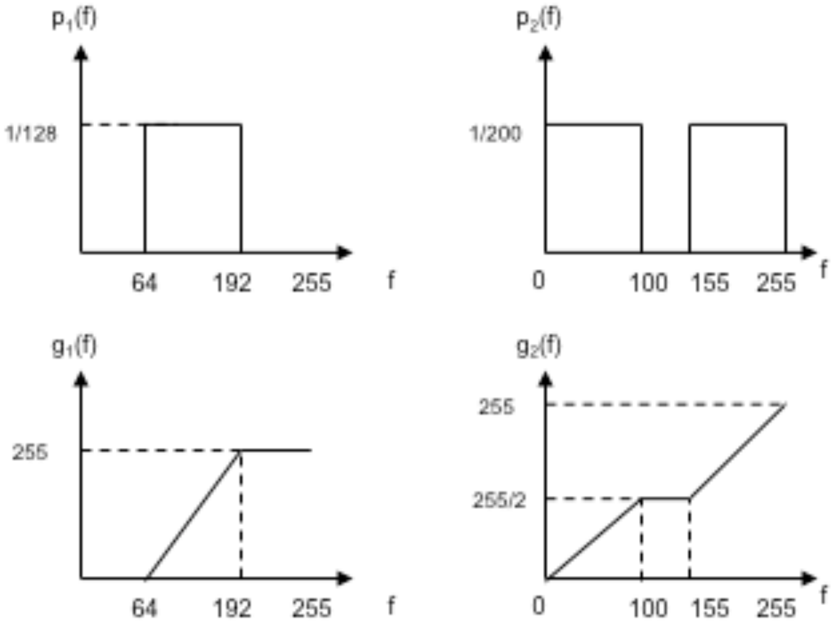
$$Y = \left\lfloor \frac{133.575}{4} \right\rfloor \times 4 + 2 = 134,\ Cb = \left\lfloor \frac{100.336}{32} \right\rfloor \times 32 + 16 = 112,\ Cr = \left\lfloor \frac{170.66}{32} \right\rfloor \times 32 + 16 = 176 \quad \text{(2+2+2pt)}$$

(e) The reconstructed RGB values from quantized YCbCr values before rounding to integers are

$$\begin{bmatrix} 1.164 & 0 & 1.596 \\ 1.164 & -0.392 & -0.813 \\ 1.164 & 2.017 & 0 \end{bmatrix} \begin{bmatrix} 134-16 \\ 112-128 \\ 176-128 \end{bmatrix} = \begin{bmatrix} 213.96 \\ 104.6 \\ 105.08 \end{bmatrix}$$ (2pt)

After rounding, the integer RGB values are R=214, G=105, B=105. (no points will be taken whether you did rounding or no).

3. (10pt) The histograms of two images are illustrated below. Sketch a transformation function for each image that will make the image has a better contrast. Use the axis provided below to sketch your transformation functions. (5pt+5pt) If shape is correct, but specifics (vertical values, horizontal transitions) are wrong, deduct a small amount.



(5pt+5pt)

The vertical value of 255/2 should be 128 since it needs to be an integer. If a student uses 127 or 255/2, it is also OK. If you use 1 and ½ in the vertical axis, deduct 2 pt.

4. (10 pt) For the image shown in Fig. A, find a transformation function (i.e. a look-up-table) that will change its histogram to match the one shown in Table A. Draw the transformed image in Fig. B. Assume that the processed images can only take integer values between 0 and 3 (including 0 and 3), and give the histograms of the original and processed images in Table B and Table C, respectively

0	0	1	1	2
0	1	1	2	2
1	1	2	3	1
1	2	3	1	1
2	3	1	0	0

Fig. A

0	0	0	0	3
0	0	0	3	3
0	0	3	3	0
0	3	3	0	0
3	3	0	0	0

Fig. B (2pt)

Table A: Desired histogram

Gray level f	0	1	2	3
Histogram h(f)	15/25	0	0	10/25

Table B: Original image histogram (2pt)

Gray level f	0	1	2	3
Histogram h(f)	5/25	11/25	6/25	3/25

Table C: Transformed image histogram (2pt)

Gray level f	0	1	2	3
Histogram h(f)	16/25	0	0	9/25

Process of deriving solutions (4pt)

Original value (f)	Histogram of f	CDF of f	CDF of z	Desired Histogram of z	Transformed value (z)
0	5/25	5/25 (map to 0)	15/25	15/25	0
1	11/25	16/25 (map to 0)	15/25	0	1
2	6/25	22/25 (map to 3)	15/25	0	2
3	3/25	1 (map to 3)	1	10/25	3

5. (5pt) For each filter given below, answer the following questions. a) Is it a separable filter? If yes, present the horizontal and vertical filters. b) What is the functionality of the filter? Explain your reasoning.

$$H_1 = \frac{1}{4}\begin{bmatrix} -1 & -2 & -1 \\ -2 & 16 & -2 \\ -1 & -2 & -1 \end{bmatrix}, \quad H_2 = \begin{bmatrix} -1 & -3 & -1 \\ 0 & 0 & 0 \\ 1 & 3 & 1 \end{bmatrix}$$

H_1: non-separable (1pt), high emphasis because coefficients sum to 1, but has both positive and negative values (1pt)

H_2: separable (1pt), $H\_2 = \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix} [1 \quad 3 \quad 1]$, edge detection because coefficients sum to 0 (1pt), horizontally smoothing, vertically edge detection to detect horizontal edges (1pt)

6. (10 pt) For the H_2 filter in the previous problem, a) Determine the DTFT H(u,v) of the filter (assuming the origin is at the center), (3pt) and sketch the one dimensional profiles H(u,0) (2pt) and H(0,v) (2pt) . Note: you should assume u represent the vertical frequency and v the horizontal frequency. b) What is the function of this filter based on its frequency response?

a)

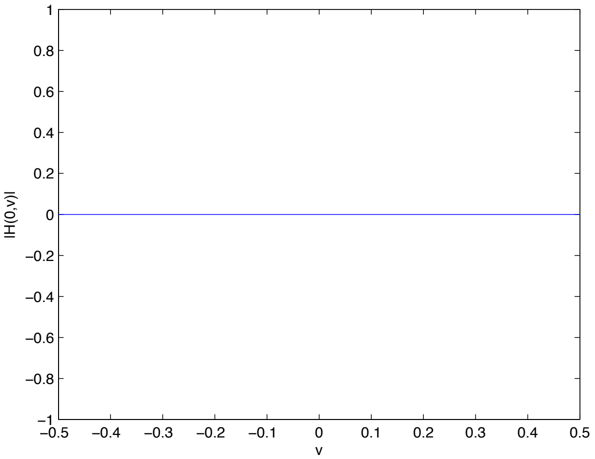
Horizontal filter: Hh(v) = $1 \times e^{-j2\pi(-1)v} + 3 \times e^{0} + 1 \times e^{-j2\pi(1)v} = 3 + 2\cos(2\pi v)$

Vertical filter: Hv(u) = $-1 \times e^{-j2\pi(-1)u} + 0 \times e^{0} + 1 \times e^{-j2\pi(1)u} = -2j\sin(2\pi u)$

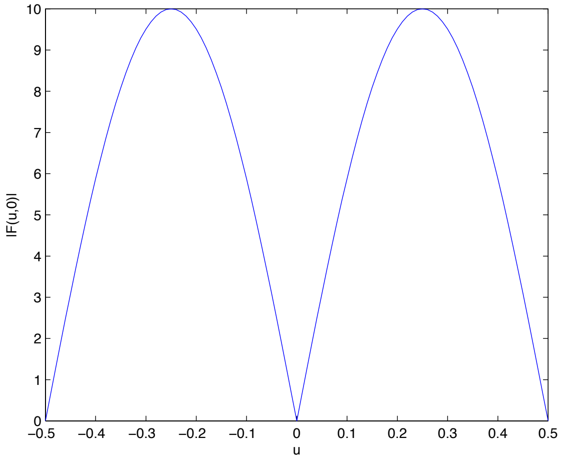
Total: H(u,v) = -2$j$sin $(2\pi u)(3 + 2\cos(2\pi v))$ (3pt)

H(0,v)=Hh(v) Hv(0)=0

H(u,0)= Hh(0) Hv(u) = $-10j\sin(2\pi u)$, |H(u,0)|=10 $|\sin(2\pi u)|$



(2pt)



(2pt)

b) horizontal low pass (1pt), vertical bandpass (1pt), overall detecting horizontal edges (1pt)

7. (10 pt) Consider linear convolution of an image of size N1xN1 with a filter of size N2xN2. We can either do the convolution directly, or using 2D fast DFT (FFT) of size N3xN3. (a) What is the number of multiplications needed if we do convolution directly? Your result should be expressed in terms of N1 and N2, and you should not assume the filter satisfy any symmetry property. (b) How should N3 be related to N1 and N2, for the result obtained using FFT to be the same as linear convolution? (c) What is the number of multiplications required by using the FFT? You should represent your results in terms of N3, which is in turn represented in terms of N1 and N2. (d) Assume N2= a*N1, where a is a fraction number between 0 and 1. For what range of a, will the FFT approach take less computation?
   (a) C1= (N1+N2-1)^2 *N2^2, (3pt) if answer N3^4, give (2pt)
   (b) N3>=N1+N2-1 (1pt)
   (c) If we assume that the DFT for the filter has been precalculated, we just need to compute DFT of the image (using separable FFT, with complexity 2 N3^2 logN3), point-wise multiplication of the DFT of the image and DFT of the filter (with complexity N2^2), and inverse DFT (complexity= 2 N3^2 logN3). The total complexity is C2= 2 N3^2 logN3 (FFT) +N3^2 (multiplication) +2 N3^2 logN3 (IFFT)= 4 N3^2 log N3 + N3^2 (3pt)
   If you assume that the DFT of the filter has to be calculated, then you will have C2= 6 N3^2 log N3 + N3^2

   (d) Let N3= (1+a)N -1 \approx = (1+a)N, (Let N=N1)
   C1=(1+a)^2 N^2 * a^2 N^2 = a^2 (1+a)^2 N^4
   C2=4 (1+a)^2 N^2 log (1+a)N +(1+a)^2 N^2 \approx = 4 (1+a)^2 N^2 log (1+a)N
   Let C2<C1 -> 4 (1+a)^2 N^2 log (1+a)N < a^2 (1+a)^2 N^4 -> 4 log (1+a)N < a^2 N^2
   Further approx, assume log (1+a)\approx 0, 4 log N < a^2 N^2 → a> 2 (log N)^(1/2) /N
   Eg. N=1024, a> 2 sqrt(10) /1024 (3pt)

8. (10 pt) You are given the following basis images for 2x2 image patterns:

$$H_{00}=\frac{1}{2}\begin{bmatrix}1 & 1\\ 1 & 1\end{bmatrix};\quad H_{01}=\frac{1}{2}\begin{bmatrix}1 & 1\\ -1 & -1\end{bmatrix};\quad H_{10}=\frac{1}{2}\begin{bmatrix}1 & -1\\ 1 & -1\end{bmatrix};\quad H_{11}=\frac{1}{2}\begin{bmatrix}1 & -1\\ -1 & 1\end{bmatrix};$$

   (a) Show that they form orthonormal basis images. 3pt
   (b) Calculate the transform coefficients of the image $F=\begin{bmatrix}4 & 7\\ 3 & 5\end{bmatrix}$ using these basis images. 3pt
   (c) Find the reconstructed image $\hat{F}$ obtained with the largest two coefficients (in magnitude). 2pt
   (d) By observing the appearance of the original and reconstructed images, explain what is the effect of not using the two coefficients with least values.

Solution:
   (a) The inner product of any 2 different H_ij is 0, and the norm of each H_ij = 1. (3pt)
   (b) $T_{00} = H_{00}\bullet F = 19/2$, $T_{01} = H_{01}\bullet F = 3/2$, $T_{10} = H_{10}\bullet F = -5/2$, $T_{11} = H_{01}\bullet F = -1/2$ (3pt)
   (c) The two largest coefficients (in magnitude) are T_{00} and T_{10}. The reconstructed image is $T_{00}\times H_{00} + T_{10}\times H_{10} = \begin{bmatrix}7/2 & 6\\ 7/2 & 6\end{bmatrix}$ (2pt)
   Some students used T_00 and T_01. If you resulting reconstructed image is correct based on T_00, T_01, I only deduct 1 pt. If answer to (d) is correct based on using T_00 and T_01, I did not deduct pt in (d).
   (d) Because coefficient T01 and T11 are discarded, the reconstructed image only contain the changes in the horizontal direction, the changes in vertical direction and diagonal direction are removed. (2pt)
   If you just said that by using the two largest coefficients, you minimize the error, you get 1pt.

9. (20pt) Write a MATLAB code to implement the following edge detection algorithm using the Sobel filters given below. Your program should i) Read in an image, ii) Perform filtering using Hx and Hy filters respectively to obtain gradient images Gx and Gy; iii) Do edge detection using the gradient images. Specifically you should calculate the gradient magnitude image by using $Gm=|Gx|+|Gy|$. For any pixel, if the gradient magnitude is greater than a threshold *T*, this pixel will be considered an edge pixel and assigned a value of 255 in the edge map. Otherwise, this pixel will be assigned a value of 0; iv) Display the original, filtered images Gx and Gy, the gradient magnitude image Gm, and the edge map Eimg; v) Save the edge map image as an image file. You should write a MATLAB function for filtering using a given filter. Your function can assume that the filter is non-zero in the region (-W,W) both horizontally and vertically. For simplicity, you only need to do filtering and edge detection over the image region which does not involve pixels outside the boundary. Your main program should call this function. You can assign the value of T and W in the main program.

$$H_x = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}, \quad H_y = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

Note that the origin of the filters *Hx* and *Hy* are both at the center.

Filtering function 10pt, all other steps are included in the other 10 pt (missing each step, or wrong in each step, up to 2 pt).

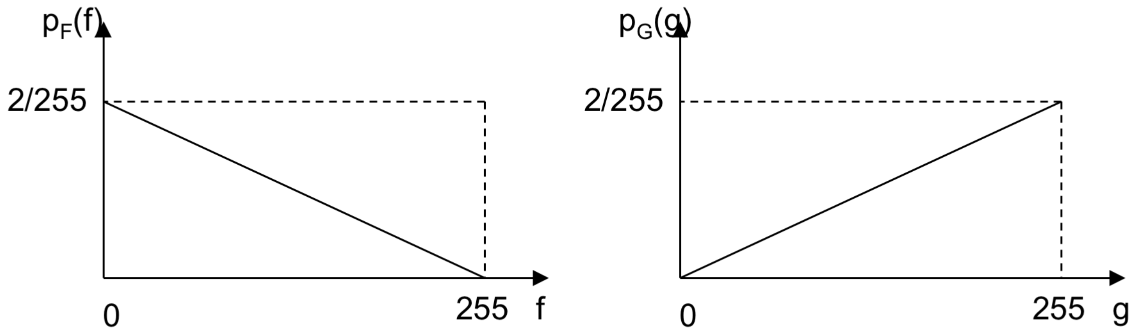
1. Read image
2. Perform filtering using Hx and Hy filters respectively to obtain gradient images Gx and Gy.
3. Do edge detection using the gradient images. $Gm=|Gx|+|Gy|$ and T
4. Display the original, filtered images Gx and Gy, the gradient magnitude image Gm, and the edge map Eimg.
5. Save the edge map image as an image file.

In the filter function:
Range of convolution: 2pt
Make it as a function: 2pt
Determine W: 2pt
Double and Uint8: 2pt
Rotate filter: 2pt

Sample program:

```
% Step 1
input_img = imread('image.jpg');

% Step 2
Hx = [-1 -2 -1; 0 0 0; 1 2 1];
Hy = [-1 0 1; -2 0 2; -1 0 1];
W = 1;

Gx = filter2d(input_img, Hx, W);
Gy = filter2d(input_img, Hy, W);

% Step 3
T = 128;
Gm = abs(Gx) + abs(Gy);

edge_img = uint8(zeros(size(Gm)));
edge_img(Gm>T) = 255;

% Step 4
figure(1);
subplot(2,2,1); imshow(Gx); subplot(2,2,2); imshow(Gy);
subplot(2,2,3); imshow(Gm); subplot(2,2,4); imshow(edge_img);

% Step 5
imwrite(edge_img, 'result.jpg', 'jpg');


function [ output_image ] = filter2d( input_image, filter, W )
%MFILTER Summary of this function goes here
%   Detailed explanation goes here
input_image = double(input_image);   2 pt

sizeImage = size(input_image);
sizeFilter = size(filter);
sizeOutput = sizeImage-sizeFilter+1;

flt_ctr = ceil(sizeFilter/2);
filter_inv = filter(flt_ctr(1)+W:-1:flt_ctr(1)-W, flt_ctr(2)+W:-1:flt_ctr(2)-W );   3 pt
output_image = zeros(sizeOutput);

for m = 1:sizeOutput(1)   5 pt
    for n = 1:sizeOutput(2)
        output_image(m,n) = sum(sum(input_image(m:m+sizeFilter(1)-1, n:n+sizeFilter(2)-
1).*filter_inv));
    end
end

end
```

10. (Bonus problem, 10 pt) Suppose an image has a probability density function as shown on the left. We would like to modify it so that it has a probability density function given on the right. Derive the transformation function g(f) that will accomplish this. For simplicity, assume both the original image and the modified image can take on gray levels in the continuous range of (0,255).



CDF of f (2pt) CDF of g (2pt), equate and solve (6pt)

$P_F(f) = \int_0^f p_F(f)df = \int_0^f \frac{2}{255}(1 - \frac{f}{255})df = \frac{2}{255}(f - \frac{1}{255} \times \frac{1}{2} f^2) = \frac{f}{255^2}(255\times2 - f)$ (2pt)

$P_G(g) = \int_0^g p_g(g)dg = \int_0^g \frac{2}{255^2} gdg = \frac{g^2}{255^2}$ (2pt)

$\frac{g^2}{255^2} = \frac{f}{255^2}(255\times2 - f) => g^2 = (510f - f^2)$

$g = \pm\sqrt{510f - f^2}$

g should be 0 ~ 255, so $g(f) = \sqrt{510f - f^2}$ (6pt)