Midterm Exam
10/31/05, 3:35 – 5:50 PM

Your Name: SOLUTION

Student ID: ________________________________

Write your solutions in the space provided next to the questions (use both sides if necessary). If you need extra space, you are probably on the wrong track!

Closed book, one sheet of notes (both sides) allowed.

<table>
<thead>
<tr>
<th>PROBLEM NO.</th>
<th>FULL POINTS</th>
<th>YOUR POINTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>Total Points</td>
<td>100</td>
<td></td>
</tr>
</tbody>
</table>

Good luck and Happy Halloween! 😊
1. (15pt) The histograms of three images are illustrated below. For each image, sketch a transformation function in the figure below that will help to equalize the histogram. Explain briefly your reasoning.

Fig. 1(a)

Fig. 1(b)

Fig. 1(c)
2. (15pt) For the image shown in Fig. 2(a), find a transformation function that will approximately equalize its histogram, and draw the transformed image in Fig. 2(b), and give the histograms of the processed image. Assume that the processed images can only take integer values between 0 and 7 (including 0 and 7),

<table>
<thead>
<tr>
<th>f</th>
<th>P(f)</th>
<th>C(f)</th>
<th>g=int(c(f)*7)</th>
<th>g</th>
<th>P(g)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>3/25</td>
<td>3/25</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>7/25</td>
<td>10/25</td>
<td>3</td>
<td>1</td>
<td>3/25</td>
</tr>
<tr>
<td>2</td>
<td>3/25</td>
<td>13/25</td>
<td>4</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>3/25</td>
<td>16/25</td>
<td>4</td>
<td>3</td>
<td>7/25</td>
</tr>
<tr>
<td>4</td>
<td>3/25</td>
<td>19/25</td>
<td>5</td>
<td>4</td>
<td>6/25</td>
</tr>
<tr>
<td>5</td>
<td>3/25</td>
<td>22/25</td>
<td>6</td>
<td>5</td>
<td>3/25</td>
</tr>
<tr>
<td>6</td>
<td>2/25</td>
<td>24/25</td>
<td>7</td>
<td>6</td>
<td>3/25</td>
</tr>
<tr>
<td>7</td>
<td>1/25</td>
<td>1</td>
<td>7</td>
<td>7</td>
<td>3/25</td>
</tr>
</tbody>
</table>

Fig.2 (a)  
Fig.2 (b)
3. (15pt) A 2D filter $H$ is given below, where the center position corresponds to $m=n=0$:

$$
H = \begin{bmatrix}
-1 & 2 & -1 \\
-2 & 4 & -2 \\
-1 & 2 & -1
\end{bmatrix}
$$

a) Is the filter separable? If so, give the one dimensional filters in horizontal and vertical directions.

b) Determine the DTFT $H(u,v)$ of the filter, and sketch the one dimensional profiles $H(u,0)$, $H(u,1/2)$, $H(0,v)$, and $H(1/2,v)$.

c) Explain the function of the filter in horizontal and vertical directions.

Solution:

a) yes. 

$$
H = \begin{bmatrix}
-1 & 2 & -1 \\
-2 & 4 & -2 \\
-1 & 2 & -1
\end{bmatrix} = \begin{bmatrix} 1 \\
2 \\
1 \end{bmatrix} \begin{bmatrix}
-1 & 2 & -1 \\
-1 & 2 & -1
\end{bmatrix} 
$$

Horizontal filter is $H_y = [-1,2,-1]$, vertical filter is $H_x = [1,2,1]$

b) $H(u,v) = H_y(v) \cdot H_x(u)$

$H_y(v) = 2 - \exp(-j2\pi v) - \exp(j2\pi v) = 2 - 2\cos(2\pi v)$. This is a high pass filter, as $H_y(0) = 0$, $H_y(1/2) = 4$

$H_x(u) = 2 + \exp(-j2\pi u) + \exp(j2\pi u) = 2 + 2\cos(2\pi u)$. This is a low pass filter, as $H_x(0) = 4$, $H_x(1/2) = 0$

$$
H(u,v) = 4(1 - \cos(2\pi v)) (1 + \cos(2\pi u))
$$

$H(u,0) = 0$, $H(u,1/2) = 8 (1 + \cos(2\pi u))$, (low pass)

$H(0,v) = 8 (1 - \cos(2\pi v)$, $H(1/2,v) = 0$. (high pass)

The filter is low pass in vertical direction (smoothing) and high pass in the horizontal direction. This filter can be used for detecting vertical edges.
4. (15 pt) You are given two 1-D vectors below: $h_0 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix}, h_1 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix}$

(a) Show that they are orthonormal vectors.

(b) Derive 4 2-D basis images for a separable unitary transform using these 1-D vectors.

(c) Calculate the transform of the image $F = \begin{bmatrix} 1 & 2 \\ 3 & 5 \end{bmatrix}$ using the basis images derived in step (b).

(d) Find the reconstructed image $\hat{F}$ obtained with the largest coefficients (in magnitude). What can you say about the physical meaning of $\hat{F}$?

Note: If you were not able to solve step (b), please use the following basis images for solving steps (c) and (d).

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

(a) To show they are orthonormal, we just need to show that $(h_k, h_l) = (h_k)^T h_l = \delta_{k,l}$. This is easily verified: $(h_0, h_0) = 1, (h_0, h_1) = 0, (h_1, h_0) = 0, (h_1, h_1) = 1$.

(b) The 2-D basis images can be formed using $H_{k,l} = (h_k)(h_l)^T$. This yields

$$H_{00} = h_0^T h_0 = \frac{1}{2} \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}; \quad H_{01} = h_0^T h_1 = \frac{1}{2} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}; \quad H_{10} = h_1^T h_0 = \frac{1}{2} \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}; \quad H_{11} = h_1^T h_1 = \frac{1}{2} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix};$$

(c) The transform coefficients can be obtained by inner product of the image $F$ with each basis image. This yields

$$t_{00} = (F, H_{00}) = \frac{11}{2}; \quad t_{01} = (F, H_{01}) = -\frac{3}{2}; \quad t_{10} = (F, H_{10}) = -\frac{5}{2}; \quad t_{11} = (F, H_{11}) = \frac{1}{2};$$

(d) The largest coefficient is $t_{00} = \frac{11}{2}$; The reconstructed image is $\hat{F} = t_{00} H_{00} = \frac{11}{4} \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$.

Note that $1/4$ is the average value of the original image pixels. Therefore, reconstructing an image by the basis image $H_{00}$ (which is an all constant basis) is equivalent to approximate an image by its mean value.
5. (20pt) (a) Find the closing of the binary image, F, in Fig. (a) by the structuring element H in Fig. (b). You can use the grids in Fig. (c) and Fig. (d) to draw the intermediate and the final results.

(b) Find the gray scale opening of the function sketched in Fig. (e) with the structural element in Fig. (f). Sketch the intermediate and final result in Figs. (g) and (h).

Comment on the effect of closing and grayscale opening.

solution

The closing operation fills the small gaps and holes and the opening operation removes thin ridges and branches.
6. (20pt) (a) List the major steps needed for performing edge detection of an image by thresholding the gradient magnitude computed using Sobel operator (see below). For simplicity, assuming the threshold value is given in advance, denoted by $T$. (b) Write a MATLAB function (or C-routine if you prefer) that will implement these steps. You should not use the MATLAB built-in \texttt{conv2} function to perform convolution. Rather, your program should try to perform all required operations using a single loop of all pixel positions. That is, looping through all pixels in the given image, and for each pixel in the loop, perform all operations needed for determining whether this pixel is an edge and save the result into an edge map.

Assuming the image size is $W$ (width) x $H$ (height). Also you can ignore the boundary problem by performing edge detection only on non-boundary pixels.

Sobel operators: $H_x = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$; $H_y = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$

Major steps: (5 pt)

For every pixel: 1) compute vertical gradient $g_x$ by convolving with $H_x$, 1) compute horizontal gradient $g_y$ by convolving with $H_y$, 3) find the gradient magnitude using $A = \sqrt{g_x^2 + g_y^2}$. 4) if $A \geq T$, this pixel is an edge, otherwise, it is not an edge.

Matlab function: (15 pt)

```matlab
%Asssuming the luminance component of the image has been read into a matrix F of size H*W
edgemap=zeros(H,W);

for (m=2:H-1) for (n=2:W-1) %skipping boundary pixels
    gx=F(m+1,n-1)+2*F(m+1,n)+F(m+1,n+1)-F(m-1,n-1)-2*F(m-1,n)-F(m-1,n+1);
    gy=F(m-1,n+1)+2*F(m,n+1)+F(m+1,n+1)-F(m-1,n-1)-2*F(m,n-1)-F(m+1,n-1);
    A=sqrt(gx*gx+gy*gy)
    if (A>=T)
        edgemap(m,n)=255;
    end;
end; end;
```

Note: If you used a matlab program that has separate loops for performing convolution with $H_x$ and $H_y$ and another for computing the magnitude of the gradient, and another for comparing the gradient magnitude to the threshold, you will get 10 points, if the program is done correctly.