E4.08
SO14
ISE4.33

IMPERIAL COLLEGE LONDON

DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING
EXAMINATIONS 2007

MSc and EEE/ISE PART IV: MEng and ACGI

**DIGITAL IMAGE PROCESSING**

Wednesday, 16 May 10:00 am

Time allowed: 3:00 hours

**There are FOUR questions on this paper.**

**Answer THREE questions.**

*All questions carry equal marks*

CORRECTED COPY

**Any special instructions for invigilators and information for candidates are on page 1.**

| | | |
|---|---|---|
| Examiners responsible | First Marker(s) : | P.T. Stathaki |
| | Second Marker(s) : | P.L. Dragotti |

**Special instructions for invigilators:** None

**Information for candidates:** None

1. (a) Consider an $M \times N$-pixel gray level image $f(x,y)$ which is zero outside $0 \le x \le M-1$ and $0 \le y \le N-1$. The image intensity is given by the following relationship

$$f(x,y) = \begin{cases} c, & x = x_0,\ 0 \le y \le N-1 \\ 0, & \text{otherwise} \end{cases}$$

where $c$ is a constant value between 1 and 255 and $x_0$ is a constant value between 0 and $M-1$.

(i) Plot the image intensity. **[3]**

(ii) Find the $M \times N$-point Discrete Fourier Transform (DFT) of $f(x,y)$. Plot its amplitude response. **[7]**

(iii) Compare the plots found in (i) and (ii). **[3]**

Hint:

The following result holds: $\sum_{k=0}^{N-1} a^k = \frac{1-a^N}{1-a}, a \ne 1$.

(b) Consider the population of vectors $\underline{f}$ of the form

$$\underline{f} = \begin{bmatrix} f_1(x,y) \\ f_2(x,y) \\ f_3(x,y) \end{bmatrix}.$$

Each component $f_i(x,y), i = 1,2,3$ represents an image. The population arises from the formation of the vectors across the entire collection of pixels.

Consider now a population of vectors $\underline{g}$ of the form

$$\underline{g} = \begin{bmatrix} g_1(x,y) \\ g_2(x,y) \\ g_3(x,y) \end{bmatrix}$$

where the vectors $\underline{g}$ are the Karhunen-Loeve transforms of the vectors $\underline{f}$.

The covariance matrix of the population $\underline{f}$ calculated as part of the transform is

$$\underline{C}_{\underline{f}} = \begin{bmatrix} a & 0 & b^2 \\ 0 & a & b^2 \\ b^2 & b^2 & a \end{bmatrix}$$

Suppose that a credible job could be done of reconstructing approximations to the three original images by using one or two principal component images. What would be the mean square error incurred in doing so in each case?

**[7]**

2. (a) An old movie is to be restored. Frames of the movie contain black spots which we wish to remove by median filtering. Figure 1 shows a representative example of a small region of a frame of the movie.

(i) Apply median filtering using a $3\times3$ window to the image area of Figure 1. Assume that pixel values are zero outside the image area of Figure 1. **[3]**

| | | | | |
|---|---|---|---|---|
| 12 | 11 | 10 | 12 | 15 |
| 12 | 11 | 0 | 11 | 14 |
| 13 | 12 | 12 | 12 | 10 |
| 12 | 10 | 13 | 12 | 11 |

Figure 1

(ii) An alternative method of processing first applies a median filter with a window of 3 columns and 1 row, and then applies to the result of this, median filtering with a 1 column by 3 row window. What is the result if this method is applied to the above image area of Figure 1? **[3]**

(iii) If the original image frames contain rectangular objects with sharp corners, which of the two methods given in (i) and (ii) above is best? Explain your answer. **[3]**

(iv) If the black spots in the image frames are $2\times2$ pixels in size, which of the methods given in (i) and (ii) is best? Explain your answer. **[3]**

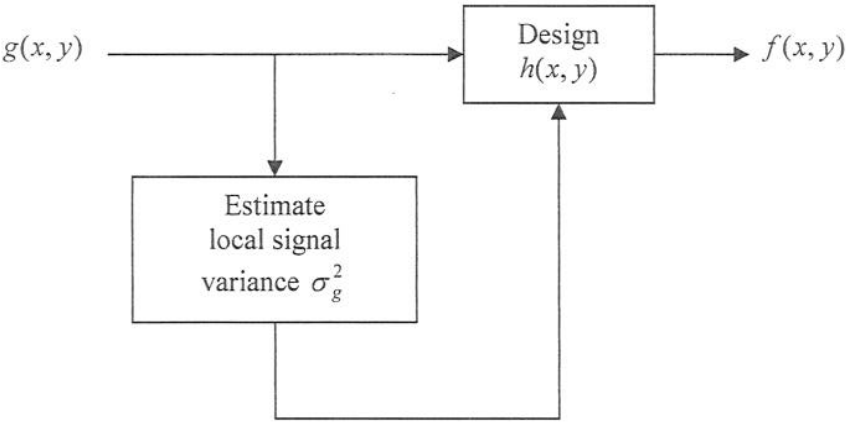
(v) Even in regions without black spots median filtering will change the exact values of many pixels. Propose a method based on a $3\times3$ window in which pixels are left unchanged unless there is a black spot. **[3]**

(b) We wish to apply a local threshold to an image $f(x,y)$ in order to find pixels which are significantly brighter than their surroundings. We create an output $g(x,y)$ such that

$$g(x,y)=\begin{cases}1, & f(x,y)>\text{mean}\{f(x,y)\}+T\\ 0, & \text{otherwise}\end{cases}$$

where $\text{mean}\{f(x,y)\}$ is the local mean averaged over a $5\times5$ pixel region, and $T$ is an input parameter. Explain how you would implement the above process using standard linear convolution followed by thresholding. What is the convolution filter kernel that should be used? **[5]**

3. We are given the noisy version $g(x,y)$ of an image $f(x,y)$. Consider the adaptive image restoration system sketched in the following figure.



In this system, $h(x,y)$ is designed at each pixel based on the local signal variance $\sigma_g^2$ estimated from the degraded image $g(x,y)$. The filter $h(x,y)$ is assumed to have the form

$$h(x,y) = k_1 e^{-k_2(x^2+y^2)} w(x,y), k_1, k_2 > 0$$

where $w(x,y)$ is a $5 \times 5$-point rectangular window placed symmetrically around the origin, i.e.,

$$w(x,y) = \begin{cases} 1, & -2 \le x \le 2, \ -2 \le y \le 2 \\ 0, & \text{otherwise} \end{cases}$$

We require that

$$\sum_{x=-\infty}^{\infty} \sum_{y=-\infty}^{\infty} h(x,y) = 1$$

(i) Derive an expression for $h(x,y)$ in the limit as $k_2$ tends to 0. [4]

(ii) Derive an expression for $h(x,y)$ in the limit as $k_2$ tends to $+\infty$. [4]

(iii) Based on the observation that random noise is typically less visible to human viewers in image regions of high detail, such as edge regions, than in image regions of low detail, such as uniform background regions, sketch one reasonable choice of $k_2$ as a function of $\sigma_g^2$. [4]

(iv) Denote your choice in (iii) as $k_2(\sigma_g^2)$. Determine $k_1$. [4]

(v) The image restoration system discussed here can exploit the observation stated in (iii). The system, however, cannot exploit the observation that random noise is typically less visible to human viewers in bright areas than in dark areas. How would you modify the image restoration system so that this additional observation can be exploited? [4]

4. (a) Consider an image with intensity $f(x,y)$ that can be modeled as a sample obtained from the probability density function sketched in the Figure 1 below.

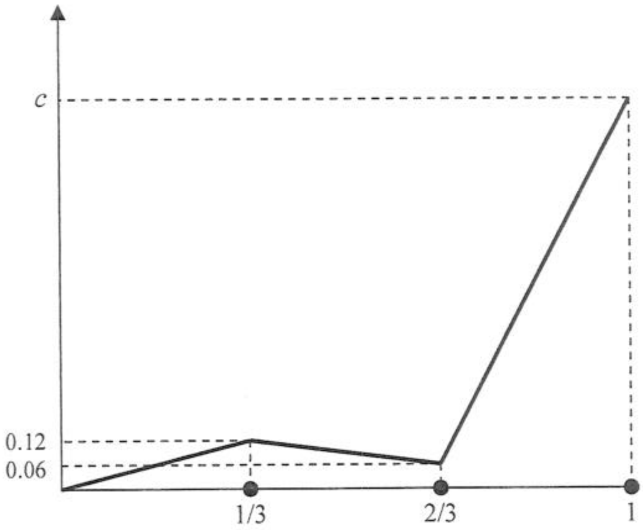


Figure 1

(i) Determine the constant $c$ shown in Figure 1. **[6]**

(ii) Suppose three reconstruction levels are assigned to quantize the intensity $f(x,y)$. Determine these reconstruction levels using a uniform quantizer. **[1]**

(iii) Determine the codeword to be assigned to each of the three reconstruction levels (symbols) using Huffman coding. For your codeword assignment, determine the average number of bits required to represent the image intensity. **[3]**

(iv) Determine the entropy, the redundancy and the coding efficiency of the Huffman code for this example. Comment on the efficiency of Huffman code for this particular set of symbols. **[3]**

(b) In the above set of symbols apply the extended-by-two Huffman coding.

(i) Explain the motivation for using extended Huffman code in the given set of symbols. **[3]**

(ii) Determine the redundancy and the coding efficiency of the extended by two Huffman code for this example. Comment on its efficiency. **[4]**

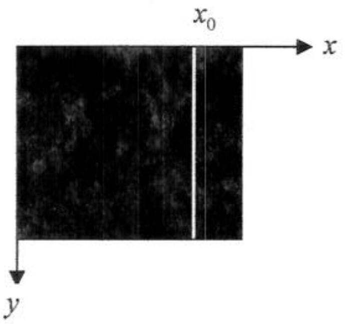
SOLUTIONS - E4.08

DIGITAL IMAGE PROCESSING ISE 4.33

2017 SO14.

1. (a)

(i) Plot the image intensity.



(ii) Find the $M \times N$ -point Discrete Fourier Transform (DFT) of $f(x,y)$. Plot its amplitude response.

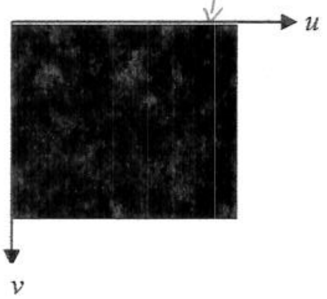
$$F(u,v)=\frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)e^{-j2\pi(ux/M+vy/N)}=\frac{1}{MN}\sum_{y=0}^{N-1} f(x_0,y)e^{-j2\pi(ux_0/M+vy/N)}$$

$$=ce^{-j2\pi ux_0/M}\sum_{y=0}^{N-1} e^{-j2\pi vy/N}=ce^{-j2\pi ux_0/M}\,\frac{1-(e^{-j2\pi vy/N})^N}{1-e^{-j2\pi vy/N}}=ce^{-j2\pi ux_0/M}\,\frac{1-e^{-j2\pi vy}}{1-e^{-j2\pi vy/N}}$$

$$F(u,v)=\begin{cases} cNe^{-j2\pi ux_0/M}, & v=0 \\ 0, & \text{otherwise}\end{cases}$$

$$|F(u,v)|=\begin{cases} cN, & v=0 \\ 0, & \text{otherwise}\end{cases}$$

a non-black line



(iii) As seen a straight line in space implies a straight line perpendicular to the original one in frequency.

(b) The eigenvalues of the covariance matrix $\underline{C}_f$ are obtained by solving the equation $\det\left|\lambda I-\underline{C}_{\underline{f}}\right|=0 \Rightarrow (\lambda-a)^3-2b^4(\lambda-a)=0 \Rightarrow \lambda=a$, $\lambda=a\pm\sqrt{2}b^2$. The eigenvalues are sorted as follows: $a+\sqrt{2}b^2 \geq a \geq a-\sqrt{2}b^2$. Therefore, if we keep one image the error will be $2a-\sqrt{2}b^2$ and if we keep two images the error will be $a-\sqrt{2}b^2$.

2. (a)

(i)

| | | | | |
|---|---|---|---|---|
| 0 | 10 | 10 | 10 | 0 |
| 11 | 12 | 11 | 12 | 11 |
| 11 | 12 | 12 | 12 | 11 |
| 0 | 12 | 12 | 11 | 0 |

(ii)

| | | | | |
|---|---|---|---|---|
| 11 | 11 | 11 | 11 | 11 |
| 11 | 11 | 11 | 12 | 11 |
| 11 | 12 | 12 | 12 | 11 |
| 10 | 12 | 12 | 12 | 10 |

(iii) For an image containing rectangular objects the second method is better. The first method 'rounds off corners' because then the 3x3 window is centred at a corner pixel, 4 pixels within the window are on the object and 5 on the background, hence the output pixel is given the value of the background. For the second method with 1x3 and 3x1 filtering a when the window is centred a corner pixel a majority of pixels (2 out of 3) is on the object hence the background has no effect.

(iv) Now the first method is better. For a window centred on one of the black pixels, 4 out of 9 pixels within the window cover the black spot, whereas a majority, 5 out of 9 are on the background, hence the output value is the background. In contrast for the second method a majority of window pixels are one the black spot and so the output is the same as the input and the black spot is not removed.

(v) Check for instance whether the smallest value in the window is less than 20% of the median value, if it is apply median filtering and replace the centre pixel with the median value. Otherwise just copy the input value to the output value.

(b) $f(x,y)\} = f(x,y) * \delta(x,y)$ and $\text{mean}\{f(x,y)\} = f(x,y) * w(x,y)$ . Therefore,

$$g(x,y) = \begin{cases} 1, & f(x,y) * (\delta(x,y) - w(x,y)) > T \\ 0, & \text{otherwise} \end{cases}$$

| | | | | |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |

Figure: $\delta(x,y)$

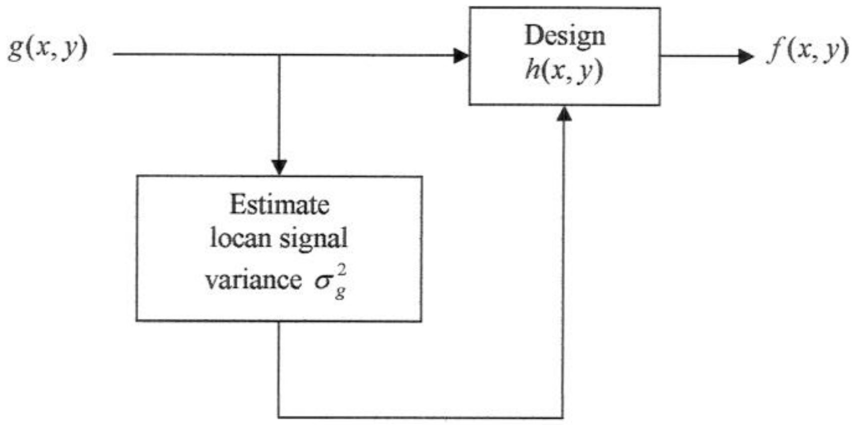
| | | | | |
|---|---|---|---|---|
| 1/25 | 1/25 | 1/25 | 1/25 | 1/25 |
| 1/25 | 1/25 | 1/25 | 1/25 | 1/25 |
| 1/25 | 1/25 | 1/25 | 1/25 | 1/25 |
| 1/25 | 1/25 | 1/25 | 1/25 | 1/25 |
| 1/25 | 1/25 | 1/25 | 1/25 | 1/25 |

Figure: $w(x,y)$

| | | | | |
|---|---|---|---|---|
| -1/25 | -1/25 | -1/25 | -1/25 | -1/25 |
| -1/25 | -1/25 | -1/25 | -1/25 | -1/25 |
| -1/25 | -1/25 | 24/25 | -1/25 | -1/25 |
| -1/25 | -1/25 | -1/25 | -1/25 | -1/25 |
| -1/25 | -1/25 | -1/25 | -1/25 | -1/25 |

Figure: $\delta(x,y) - w(x,y)$

3. We are given the noisy version $g(x,y)$ of an image $f(x,y)$. Consider the adaptive image restoration system sketched in the following figure.



In this system, $h(x,y)$ is designed at each pixel based on the local signal variance $\sigma_g^2$ estimated from the degraded image $g(x,y)$. The filter $h(x,y)$ is assumed to have the form

$$h(x,y) = k_1 e^{-k_2(x^2+y^2)} w(x,y)$$

where $w(x,y)$ is a $5 \times 5$ -point rectangular window placed symmetrically around the origin, i.e.,

$$w(x,y) = \begin{cases} 1, & -2 \le x \le 2,\ -2 \le y \le 2 \\ 0, & \text{otherwise} \end{cases}$$

We require that

$$\sum_{x=-\infty}^{\infty} \sum_{y=-\infty}^{\infty} h(x,y) = 1$$

(i) For a very small $k_2$ we can write $h(x,y) = k_1 w(x,y)$

(ii) For a very large $k_2$ we can write $h(x,y) = k_1 \delta(x,y)$

(iii) In image regions of high detail, such as edge regions, we don't want to do any filtering and therefore we want large $k_2$. In image regions of low detail, such as uniform background regions, we want to de-noise and therefore we want small $k_2$. Therefore, we can choose $k_2 = \sigma_g^2$

(iv) $k_1 = \dfrac{1}{\sum_{x=-2}^{2} \sum_{y=-2}^{2} e^{-\sigma_g^2(x^2+y^2)}}$

(v) A possible scenario is to filter both high detail and bright areas. Bright areas can be identified from the local mean $m_g$ (A large $m_g$ indicates a bright area). Therefore, we can normalize $\sigma_g^2$ and $m_g$ so that they occupy the same range of values (for example from 0 to 1) to obtain $\tilde{\sigma}_g^2$ and $\tilde{m}_g$ and then we can consider $k_2 = \tilde{\sigma}_g^2 + \tilde{m}_g$.

4. (a)

(i) The integral of the pdf between 0 and 1 must be 1 and according to this restriction we find $c = 5.64$.

(ii) The three reconstruction levels are 1/6, 3/6, 5/6.

(iii) Using Huffman code we get:

| Symbol | Probability | Code |
|---|---|---|
| $s_1 = 1/6$ | 0.02 | 11 |
| $s_2 = 3/6$ | 0.03 | 10 |
| $s_3 = 5/6$ | 0.95 | 0 |

Average number of bits per symbol $l_{avg} = 1.05$ bits/symbol.

(iv) Entropy $H(s) = n = 0.335$ bits/symbol.

Redundancy $l_{avg} - n = 1.05 - 0.335 = 0.715$ or $\frac{l_{avg} - n}{n}\% = 213\%$ of entropy! Huffman code exhibits a very high redundancy for the specific alphabet and therefore is not efficient enough.

(b)

(i) There is a strong motivation for the use of extended Huffman code since the redundancy obtained by using standard Huffman is too high. We now merge the symbols in groups of two symbols. In the next table the extended alphabet and corresponding probabilities and Huffman codewords are shown.

| Symbol | Probability | Code |
|---|---|---|
| $s_1 s_1$ | 0.0004 | 110011 |
| $s_1 s_2$ | 0.0006 | 110001 |
| $s_1 s_3$ | 0.019 | 1101 |
| $s_2 s_1$ | 0.0006 | 110010 |
| $s_2 s_2$ | 0.0009 | 110000 |
| $s_2 s_3$ | 0.0285 | 101 |
| $s_3 s_1$ | 0.019 | 111 |
| $s_3 s_2$ | 0.0285 | 100 |
| $s_3 s_3$ | 0.9025 | 0 |

**Table:** The extended alphabet and corresponding Huffman code

(ii) For the new extended alphabet we have
$l_{avg} = 1.222$ bits/**new symbol** or $l_{avg} = 0.611$ bits/original symbol.

Redundancy $\frac{l_{avg} - n}{n}\% = 72\%$ of entropy. Huffman code exhibits a very high redundancy for the specific alphabet and therefore is not efficient enough.