Version JL/1

EGT3
ENGINEERING TRIPOS PART IIB: SOLUTIONS

Tuesday 6 May 2025   2 to 3.40

Module 4F8

IMAGE PROCESSING AND IMAGE CODING

*Answer not more than **three** questions.*

All questions carry the same number of marks.

*The **approximate** percentage of marks allocated to each part of a question is indicated in the right margin.*

Answers to questions in each section should be tied together and handed in separately.

*Write your candidate number **not** your name on the cover sheet.*

STATIONERY REQUIREMENTS
Single-sided script paper

SPECIAL REQUIREMENTS TO BE SUPPLIED FOR THIS EXAM
CUED approved calculator allowed
Engineering Data Book

10 minutes reading time is allowed for this paper at the start of the exam.

You may not start to read the questions printed on the subsequent pages of this question paper until instructed to do so.

You may not remove any stationery from the Examination Room.

1 (a) Perception in images is very much concerned with lines and edges. Phase is strongly related to lines and edges. If the phase response of a filtering operation disturbs the phase, lines and edges will be blurred. To illustrate this we can FT an image then a) set the phases to zero (so there is only amplitude information) and inverse FT, and b) set all amplitudes to 1 (so there is only phase information) and inverse FT. We will see that it is the inverse FT of the phase image that picks out lines and edges. [10%]

(b) Since $u_i' = \alpha u_i$, the 2D Fourier Transform of the scaled function is:

$$
\begin{aligned}
G'(\omega_1, \omega_2) &= \int\int g(\alpha u_1, \alpha u_2) \exp\left(-j\left(\omega_1 u_1 + \omega_2 u_2\right)\right) du_1 du_2 \\
&= \int\int g(u_1', u_2') \exp\left(-j\left(\frac{\omega_1}{\alpha}u_1' + \frac{\omega_2}{\alpha}u_2'\right)\right) \frac{du_1'}{\alpha}\frac{du_2'}{\alpha}
\end{aligned}
$$

thus

$$
G'(\omega_1, \omega_2) = \frac{1}{\alpha^2} G\left(\frac{\omega_1}{\alpha}, \frac{\omega_2}{\alpha}\right)
$$

[20%]

(c) We have

$$
u_1 = r\cos\phi, \quad u_2 = r\sin\phi, \quad \omega_1 = \omega\cos\theta, \quad \omega_2 = \omega\sin\theta
$$

Changing our FT integral to polar coordinates (and recalling that $du_1 du_2 = r dr d\phi$) gives:

$$
\begin{aligned}
G(\omega, \theta) = G(\omega_1, \omega_2) &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(u_1, u_2) \exp\left(-j\left(\omega_1 u_1 + \omega_2 u_2\right)\right) du_1 du_2 \\
&= \int_0^{2\pi}\int_0^{\infty} g(r, \phi) \exp\left(-j\omega r\left(\cos\theta\cos\phi + \sin\theta\sin\phi\right)\right) r dr d\phi
\end{aligned}
$$

therefore

$$
f(r, \phi, \omega, \theta) = r\exp\left(-j\omega r\cos(\phi - \theta)\right)
$$

[20%]

(d) We want the integral

$$
\int_0^{2\pi}\int_0^{\infty} g(r, \phi) \exp\left(-j\omega r\cos(\phi - \theta)\right) r dr d\phi
$$

Let us do the r integral first, and use the information given in the question. As we can take $g(r, \phi) = \epsilon a_{\max}\delta(r - r_0)$, the r integral will pick out the value of the integrand at $r = r_0$:

$$
\int_0^{\infty} g(r, \phi) \exp\left(-j\omega r\cos(\phi - \theta)\right) r dr = \epsilon a_{\max} \exp\left(-j\omega r_0 \cos(\phi - \theta)\right) r_0
$$

Then we can do the ϕ integral. Noting that ϕ goes from 0 to 2π, we can see that the value of θ is irrelevant and we can just form

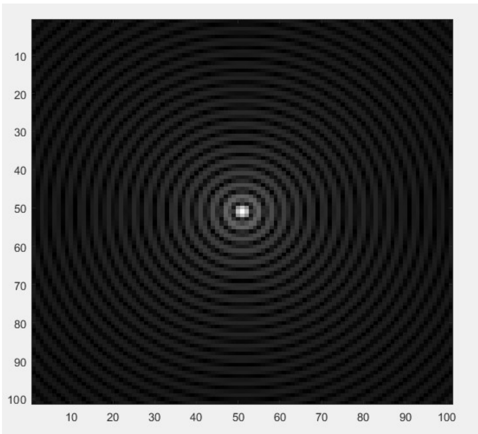
$$\epsilon a_{\max} r_0 \int_0^{2\pi} \exp\left(-j\omega r_0 \cos\phi\right) d\phi$$

giving an overall answer of

$$G(\omega, \theta) = 2\pi a_{\max} r_0 \epsilon J_0(r_0 \omega)$$

[20%]

(e) Sketch (origin in the centre and absolute values shown) is shown below. From part (c) and fig.1 we can see that the modulus of G is circularly symmetric and oscillates between high and low values as we increase in radial distance - gradually decreasing the overall value for high r. [10%]



(f) It is clear that the phase of G is zero everywhere. If the circle is translated so that the origin lies at $(c, 0)$, we know that we acquire a phase factor of $e^{-jc\omega_1}$, so that there is a phase ramp as we move horizontally across our Fourier plane. [10%]

(g) For these, we multiply by a phase factor of $\exp(-jc\omega_1)$ for the circle shifted by positive c and $\exp(jc\omega_1)$ for the circle shifted by negative c, with the amplitudes remaining unchanged (and of course still centred at the origin).
When added together, however, we have a sum

$$\exp(-jc\omega_1) + \exp(+jc\omega_1) = 2\cos(c\omega_1)$$

so that the phase is zero, but the amplitude which we obtained earlier is directly modulated.

[10%]

2 (a) Sampling on a rectangular grid:

(i) The FT of the sampled image G_s (with sampling intervals of Δ_1, Δ_2) is given by

$$G_s(\omega_1, \omega_2) = \frac{1}{\Delta_1\,\Delta_2} \sum_{p_1=-\infty}^{\infty} \sum_{p_2=-\infty}^{\infty} G(\omega_1 - p_1\Omega_1, \omega_2 - p_2\Omega_2)$$

where G is the FT of the unsampled image. [15%]

(ii) Refer to Fig.1 – we see that the amplitude of the Bessel function drops from 1 at $x = 0$ to approx 0.1 around $x = 64$. We know from part (a)(i) that the spectrum repeats every interval of the sampling frequencies – which is therefore at gridpoints $(2\pi/\Delta_1, 2\pi/\Delta_2)$. Circular symmetry implies we need the same Δ_1 and Δ_2.
If we consider the frequency in ω_1 direction, the FT has amplitude $kJ_0(\omega_1)$, so at $\omega_1 = 64$ the amplitude has dropped to 0.1 of its value at $\omega_1 = 0$ – thus we want our sampling frequency $2\pi/\Delta_1$ to be at least 2×64.
Which tells us that $\Delta_1 < \pi/64$. [20%]

(b) In image filtering we convolve with an IR (or PSF):

(i) When we design a filter with desired zero-phase 2D frequency response, the spatial domain behaviour is to convolve the image with the IFT of this filter. This IFT will in general be of infinite extent – we therefore multiply this by a window function to force it to be of finite extent. In the frequency domain this corresponds to convolving the ideal frequency response with the FT of the window function – the ideal frequency response is therefore distorted. [10%]

(ii) The two most common forms of forming 2D window functions from 1D window functions (of which there are many) are:

The *product method*: $w(n_1, n_2) = w_1(n_1)w_2(n_2)$. In the Fourier domain this implies that $W(\omega_1, \omega_2) = W_1(\omega_1)W_2(\omega_2)$.

The *rotation method*: In this method we firstly rotate a 1D continuous window $w(u_1, u_2) = w_1(u)|_{u=\sqrt{u_1^2+u_2^2}}$. Then we sample the 2D continuous window: $w(n_1, n_2) = w(u_1, u_2)|_{u_1=n_1\Delta_1, u_2=n_2\Delta_2}$. [10%]

(c) (i) If we assume that our noise is Gaussian and that our image has Gaussian statistics, the standard form of the Wiener filter is:

$$G(\omega) = \frac{H^*(\omega)(P_{xx}(\omega))}{|H(\omega)|^2(P_{xx}(\omega)) + (P_{dd}(\omega))}$$

 (cont.

where the true image x is obtained by filtering the observed image y with a filter g, and G is the FT of g. H is the FT of the distorting function h and P_{xx} and P_{dd} are the power spectra of the image and noise respectively.

Fine if the above answer is given, but it is more likely that the following form will be described:

Under the assumptions as above (gaussianity), a Bayesian derivation of the Wiener filter produces the following form

$$W = (C^{-1} + L^T N^{-1} L)^{-1} L^T N^{-1} \quad = \quad \text{"}\frac{L^T N^{-1}}{C^{-1} + L^T N^{-1} L}\text{"}$$

where $C = E[\mathbf{x}\mathbf{x}^T]$ is the covariance matrix for the image, $N = E[\mathbf{d}\mathbf{d}^T]$ is the covariance of the noise, and L is the distorting function. [10%]

(ii) If there is no convolution, our expression for $P(\mathbf{x}|\mathbf{y})$ is, under our assumptions of gaussianity, given by

$$P(\mathbf{x}|\mathbf{y}) \propto P(\mathbf{y}|\mathbf{x})P(\mathbf{x}) = \exp\left(-\frac{1}{2}((\mathbf{y}-\mathbf{x})^T N^{-1}(\mathbf{y}-\mathbf{x}) + \mathbf{x}^T C^{-1}\mathbf{x}\right)$$

With the given prior, this reduces to

$$P(\mathbf{x}|\mathbf{y}) \propto \exp\left(-\frac{1}{2}\left[\frac{1}{\sigma^2}\sum_i (y_i - x_i)^2 + \frac{1}{\lambda}\sum_i x_i^2\right]\right)$$

where the sum is over pixel values.

Differentiating the RHS wrt x_j gives

$$\left[\frac{-1}{\sigma^2}(y_j - x_j) + \frac{1}{\lambda}x_j\right]\exp\left(-\frac{1}{2}\left[\frac{1}{\sigma^2}\sum_i (y_i - x_i)^2 + \frac{1}{\lambda}\sum_i x_i^2\right]\right)$$

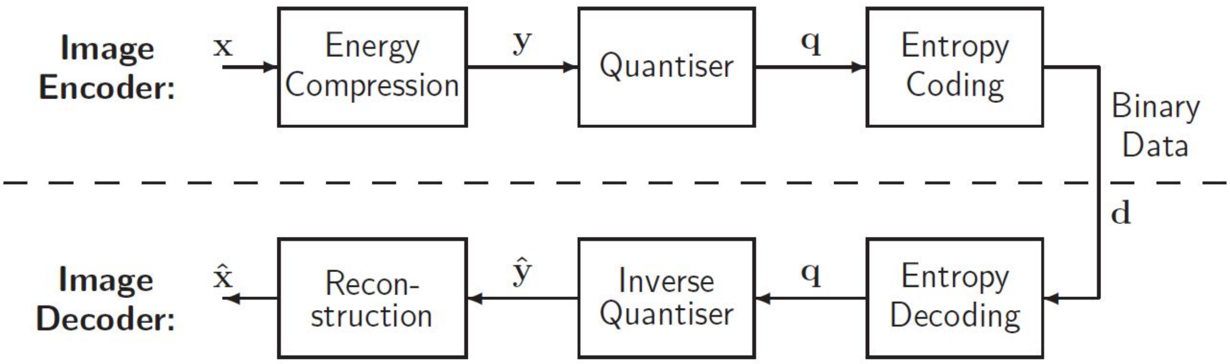
Setting this to zero gives (for all j)

$$\hat{x}_j = \frac{y_j}{1 + \sigma^2/\lambda}$$

[25%]

(iii) Note that this is telling us the $\hat{x}_j = W y_j$ where $W = \frac{(1/\sigma^2)}{1/\lambda + 1/\sigma^2}$ corresponds to our Wiener filter form in part (c)(ii). [10%]

3 (a) The figure below shows the main blocks in any *image coding* system. $\mathbf{x}$ is a monochrome) image. Elements of $\mathbf{x}$ are $x(\mathbf{n}) \equiv x(n_1, n_2)$, where n_1 runs over rows and n_2 runs over columns. The *Binary Data* is the transmitted data. Energy should only be lost in the quantiser stages.



(b) To apply the 2×2 Haar transform, $\mathbf{T}$, to a $2n \times 2n$ image, X, we proceed by splitting the image into $n \times n$ (2×2) blocks and applying $\mathbf{T}$ to each of these blocks in a 2-sided manner to form $n \times n$ transform blocks, each 2×2, giving the transformed image where each block is formed via $Y = \mathbf{T}X\mathbf{T}^T$.

For each of the $n \times n$ blocks of Y, we take the top LH pixel (keeping them in the horizontal and vertical order that they appear in Y) and amalgamate them into an $n \times n$ subimage that we place in the top left of a $2n \times 2n$ array. We then do the same with the top RH/bottom LH/bottom RH pixels, placing the resulting $n \times n$ subimages in the top RH/bottom LH/bottom RH of the $2 \times 2n$ array.

[*students will probably explain all this via a diagram – which is sensible*]

The 4 resulting $n \times n$ subimages have the following properties:

(i) Top Left: Horizontal low-pass and vertical low-pass (Lo-Lo) filtering

(ii) Top Right: Horizontal high-pass and vertical low-pass (Hi-Lo) filtering

(iii) Bottom Left: Horizontal low-pass and vertical high-pass (Lo-Hi) filtering

(iv) Bottom Right: Horizontal high-pass and vertical High-pass (Hi-Hi) filtering

[15%]

(c) (i) If we apply the Haar transform to each of the 2×2 sub-blocks we get

$$TX_{11}T^T = TX_{21}T^T = \frac{1}{2}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}\begin{bmatrix} 1 & 2 \\ 1 & 2 \end{bmatrix}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = \begin{bmatrix} 3 & -1 \\ 0 & 0 \end{bmatrix}$$

$$TX_{12}T^T = TX_{22}T^T = \frac{1}{2}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

Therefore Y is given below and we rearrange to give Y':

$$Y = \begin{bmatrix} 3 & -1 & 1 & 0 \\ 0 & 0 & 0 & -1 \\ 3 & -1 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}, \quad Y' = \begin{bmatrix} 3 & 1 & -1 & 0 \\ 3 & 1 & -1 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & -1 \end{bmatrix}$$

As in part (b), for Y', top LH subimage is Lo-Lo, top RH is Hi-Lo, bottom LH is Lo-Hi and bottom RH is Lo-Lo. [20%]

(ii) If we apply a second level Haar transform to the Lo-Lo image of Y we obtain:

$$TY'_{11}T^T = \frac{1}{2}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}\begin{bmatrix} 3 & 1 \\ 3 & 1 \end{bmatrix}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = \begin{bmatrix} 4 & 2 \\ 0 & 0 \end{bmatrix}$$

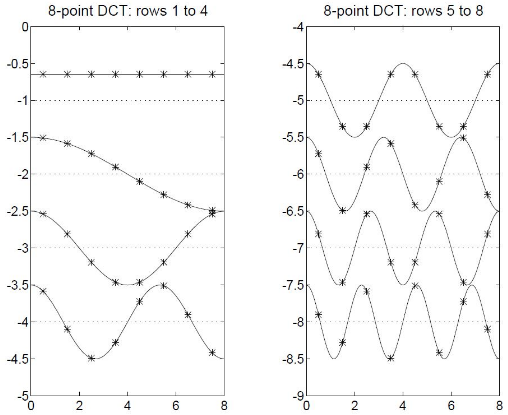
Therefore, our 2-level Haar transform results in Y''

$$Y'' = \begin{bmatrix} 4 & 2 & -1 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & -1 \end{bmatrix}$$

[10%]

(iii) The total energy in X is the sum of the squared pixels, which is $8 \times 1^2 + 2 \times 2^2 = 24$. The total energy in Y'' is similarly $4^2 + 2^2 + 4 \times (-1)^2 = 24$. So total energy has indeed been preserved. We also see that most of the energy has been compressed into two coefficients, so the compression has been successful. [10%]

(d) (i) The rows of the DCT are sketched below:



From this we can see that odd rows are symmetric about the centre point and even rows are antisymmetric. These symmetry properties are useful in constructing faster algorithms for computing the DCT. [15%]

(ii) We can form the total mean entropy in each subimage from the formula:
$(i + j)$ can sum to 2 through 16.
We have 1 subimage that sums to 2 (H_{11}) and 1 subimage that sums to 16 (H_{88}).
We have 2 subimages that sum to 3 ($H_{12,21}$) and 2 subimages that sum to 15 ($H_{87,78}$).
We have 3 subimages that sum to 4 ($H_{13,31,22}$) and 3 subimages that sum to 14 ($H_{86,68,77}$).
We have 4 subimages that sum to 5 ($H_{14,41,23,32}$) and 4 subimages that sum to 13 ($H_{85,58,76,67}$).
We have 5 subimages that sum to 6 ($H_{15,51,24,42,33}$) and 5 subimages that sum to 12 ($H_{84,48,75,57,66}$).
We have 6 subimages that sum to 7 ($H_{16,61,25,52,34,43}$) and 6 subimages that sum to 11 ($H_{83,38,74,47,65,56}$).
We have 7 subimages that sum to 8 ($H_{17,71,26,62,35,53,44}$) and 7 subimages that sum to 10 ($H_{82,28,73,37,64,46,55}$).
We have 8 subimages that sum to 9 ($H_{18,81,27,72,36,63,45,54}$)
Therefore our total entropy (as each subimage has 64 pixels) is given by $S1 + S2$ where

$$S1 = 64 * (1\times(6/1)+2\times(6/2)+3\times(6/3)+4\times(6/4)+5\times(6/5)+6\times(6/6)+7\times(6/7)+8\times(6/8))$$

$$= 64 \times 6 \times [1 + 3 + \ldots + 8] = 64 \times 48$$

and

$$S2 = 64 \times 6 \times [1/15 + 2/14 + 3/13 + 4/12 + 5/11 + 6/10 + 7/9] = 64 \times 15.6$$

 (cont.

So the total number of bits needed is $64 \times 63.6 \approx 4073$, with the biggest contributions coming from the subbands with lower i, j as expected. [15%]

4 (a) The figure shows a) which is downsampling-filtering-upsampling, b) which is filtering (z^2)-downsampling-upsampling and c) downsampling-upsampling-filtering (z^2). We want to show that these are all equivalent.

First we address a): write the downsampling stage as $f(n) = x(2n)$ and then convolve with h:

$$\sum_i f(n-i)\ h(i) = \sum_i x(2(n-i))\ h(i) = \sum_i x(2n-2i)\ h(i)$$

Now upsample (fill in with zeros):

$$\begin{aligned} \hat{y}(n) &= \sum_i x(n-2i)\ h(i) \quad \text{for } n \text{ even} \\ &= 0 \qquad\qquad\qquad\quad\ \ \text{for } n \text{ odd} \end{aligned}$$

Now take z-transforms:

$$\hat{Y}(z) = \sum_n \hat{y}(n)\ z^{-n} = \sum_{\text{even } n} \sum_i x(n-2i)\ h(i)\ z^{-n}$$

Reverse the order of summation and let $m = n - 2i$:

$$\begin{aligned} \Longrightarrow \hat{Y}(z) &= \sum_i h(i) \left(\sum_{\text{even } m} x(m)\ z^{-m}\ z^{-2i} \right) \\ &= \left(\sum_i h(i)\ z^{-2i} \right) \left(\sum_{\text{even } m} x(m)\ z^{-m} \right) \end{aligned}$$

which is equivalent to

$$\hat{Y}(z) = H(z^2) \frac{1}{2} [X(z) + X(-z)]$$

ie to filtering with z^2 and then downsampling-upsampling (as in b)).

However we can also write the above as

$$\hat{Y}(z) = \frac{1}{2} [X(z) + X(-z)]\ H(z^2)$$

which is downsampling-upsampling followed by filtering ($H(z^2)$) – ie the operation in c).

[30%]

(b) Substituting in the antialiasing PR LHS, we have:

$$G_0(z)H_0(-z) + zH_0(-z)(-z)^{-1}G_0(z) = G_0(z)H_0(-z) - G_0(z)H_0(-z) = 0$$

as required. [10%]

(c) Using the constructions in part (b) we have

$$P(z) + P(-z) = G_0(z)H_0(z) + G_0(-z)H_0(-z) = G_0(z)H_0(z) + zH_1(z)z^{-1}G_1(z)$$

$$= G_0(z)H_0(z) + G_1(z)H_1(z) = 2$$

If this condition is satisfied, we require all terms in even powers of z in $P(z)$ to be zero except the z^0 term which should be 1. The odd powers of z can take any values since they cancel out. If in addition we want to make our filter zero phase, we require symmetry about the z^0 term. [20%]

(d) We take the LeGall;(3,5) lowpass wavelets as given. Under the given constructions we have:

$$G_1(z) = \frac{1}{2}z(-z + 2 - z^{-1}) = \frac{1}{2}(-z^2 + 2z - 1)$$

$$H_1(z) = \frac{1}{8}z^{-1}(-z^2 - 2z + 6 - 2z^{-1} - z^{-2})$$

However, these are not explicitly needed as we showed in part b) that one of the PR conditions is automatically satisfied. To show that the other condition (in part c)) is satisfied we just need to evaluate the filter $P_(z)$

$$P_(z) = \frac{1}{16}(z + 2 + z^{-1})(-z^2 + 2z + 6 + 2z^{-1} - z^{-2})$$

$$= \frac{1}{16}[-z^{-3} + 9z + 16 + 9z^{-1} - z^{-3}]$$

The z^0 component is 1 and there are no even powers, so PR conditions in part c) is satisfied (notice that it is also zero phase). [20%]

(e) From part a) we know that $H_{00}(z) = H_0(z)H_0(z^2)$ and $H_{01}(z) = H_0(z)H_1(z^2)$. We therefore just need to evaluate these:

$$H_{00}(z) = H_0(z)H_0(z^2) = \frac{1}{4}(z + 2 + z^{-1})(z^2 + 2 + z^{-2})$$

$$= \frac{1}{4}[z^3 - 2z^2 - z + 4 - z^{-1} - 2z^{-2} + z^{-3}]$$

and

$$H_{01}(z) = H_0(z)H_1(z^2) = \frac{1}{16}(z + 2 + z^{-1})(z^{-2})(-z^4 - 2z^2 + 6 - 2z^{-2} - z^{-4})$$

$$= \frac{1}{16}[-z^3 - 2z^2 - 3z - 4 + 4z^{-1} + 12z^{-2} + 4z^{-3} - 4z^{-4} - 3z^{-5} - 2z^{-6} - z^{-7}]$$

[20%]

END OF PAPER