Version JL/2

EGT3
ENGINEERING TRIPOS PART IIB: SOLUTIONS

Monday 18 April 2016 9:30 to 11:00

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.

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

1 (a) (i) The sampling grid $s(u_1,u_2)$ can be expressed as a sum of two rectangular sampling grids, s_1 and s_2, as follows

$$s(u_1,u_2) = s_1(u_1,u_2) + s_2(u_1,u_2)$$

$$= \sum_{n_1=-\infty}^{\infty}\sum_{n_2=-\infty}^{\infty} \delta[u_1 - n_1\Delta_1, u_2 - n_2\Delta_2] + \delta[u_1 - (n_1 + \frac{3}{4})\Delta_1, u_2 - (n_2 + \frac{1}{2})\Delta_2]$$

with $\Delta_1 = \Delta$ and $\Delta_2 = \Delta/2$.

The Fourier series expressions for s_1 and s_2 are

$$s_1(u_1,u_2) = \sum_{p_1=-\infty}^{\infty}\sum_{p_2=-\infty}^{\infty} c_1(p_1,p_2)\, e^{j(p_1\Omega_1 u_1 + p_2\Omega_2 u_2)}$$

and

$$s_2(u_1,u_2) = \sum_{p_1=-\infty}^{\infty}\sum_{p_2=-\infty}^{\infty} c_2(p_1,p_2)\, e^{j(p_1\Omega_1 u_1 + p_2\Omega_2 u_2)}$$

where $\Omega_1 = \frac{2\pi}{\Delta_1} = \frac{2\pi}{\Delta}$ and $\Omega_2 = \frac{2\pi}{\Delta_2} = \frac{4\pi}{\Delta}$. As s_1 is simply a rectangular grid centred on the origin, we know that the Fourier coefficients are given by

$$c_1(p_1,p_2) = \frac{1}{\Delta_1\Delta_2} = \frac{2}{\Delta^2}$$

Grid s_2 is simply a translated s_1. The shift theorem tells us that $s_1(u_1 - a_1, u_2 - a_2)$ will acquire a factor of $\mathrm{e}^{-j(a_1 p_1\Omega_1 + a_2 p_2 \Omega_2)}$ relative to the Fourier coefficients of $s_1(u_1,u_2)$. Thus,

$$c_2(p_1,p_2) = c_1(p_1,p_2)\mathrm{e}^{-j(\frac{3\Delta}{4}p_1\Omega_1 + \frac{\Delta}{4}p_2\Omega_2)} = c_1(p_1,p_2)\mathrm{e}^{-j(\frac{3\pi}{2}p_1 + \pi p_2)}$$

Therefore our final Fourier series expressions for s_1 and s_2 are

$$s_1(u_1,u_2) = \frac{2}{\Delta^2}\sum_{p_1=-\infty}^{\infty}\sum_{p_2=-\infty}^{\infty} e^{j(\frac{2\pi}{\Delta}p_1 u_1 + p_2\frac{4\pi}{\Delta}u_2)}$$

and

$$s_2(u_1,u_2) = \frac{2}{\Delta^2}\sum_{p_1=-\infty}^{\infty}\sum_{p_2=-\infty}^{\infty} \mathrm{e}^{-j(\frac{3\pi}{2}p_1 + \pi p_2)} e^{j(\frac{2\pi}{\Delta}p_1 u_1 + p_2\frac{4\pi}{\Delta}u_2)}$$

[30%]

(ii) The frequency shift theorem enables us to write the Fourier transform of a signal sampled on a rectangular grid (centred on origin, and sampled at Δ_1, Δ_2) as

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

Thus for our case $G_s(\omega_1, \omega_2)$ takes the form

$$G_s(\omega_1, \omega_2) = \frac{2}{\Delta^2} \sum_{p_1=-\infty}^{\infty} \sum_{p_2=-\infty}^{\infty} G\left(\omega_1 - \left[\frac{2\pi}{\Delta}\right] p_1, \omega_2 - \left[\frac{4\pi}{\Delta}\right] p_2\right) [1 + \mathrm{e}^{-j(\frac{3\pi}{2}p_1 + \pi p_2)}]$$

so that

$$f(\Delta) = \frac{2}{\Delta^2}$$
$$\beta_1 = \frac{2\pi}{\Delta}$$
$$\beta_2 = \frac{4\pi}{\Delta}$$
$$W = \mathrm{e}^{-j(\frac{3\pi}{2}p_1 + \pi p_2)}$$

[20%]

(b) (i) Perception of images is very much concerned with lines and edges. It can be shown that if we discard the amplitude information present in the 2D FT of an image, we can still reconstruct a recognisable image due to the fact that edge information is retained in the phases of the FT.
If a filter phase response is non-linear, then the various frequency components which contribute to an edge in an image will be phase-shifted with respect to each other in such a way that they no longer add up to produce a sharp edge – i.e. dispersion takes place. It is often simplest to enforce the *zero-phase* condition, i.e. insisting that the frequency response is purely real, so that

$$H(\omega_1, \omega_2) = H^*(\omega_1, \omega_2)$$

Thus, ensuring that our filters are zero-phase will ensure that we preserve edges – crucial for image recognition. [15%]

(ii) Can first of all do this via straightforward FTs. Firstly we can write the frequency response as

$$H(\omega_1, \omega_2) = H_0 - H_1 H_2$$

where

$$H_0(\omega_1,\omega_2) = \begin{cases} 1 & \text{if } |\omega_1| < \Omega_{U1} \text{ and } |\omega_2| < \Omega_{U2} \\ 0 & \text{otherwise} \end{cases}$$

$$H_1(\omega_1,\omega_2) = \begin{cases} 1 & \text{if } \Omega_{L1} < |\omega_1| < \Omega_{U1} \\ 0 & \text{otherwise} \end{cases}$$

$$H_2(\omega_1,\omega_2) = \begin{cases} 1 & \text{if } \Omega_{L2} < |\omega_2| < \Omega_{U2} \\ 0 & \text{otherwise} \end{cases}$$

Taking the IFT of $H(\omega_1,\omega_2)$ gives us

$$h(n_1,n_2) = \frac{\Delta_1\Delta_2}{(2\pi)^2}\int_{-\pi/\Delta_2}^{\pi/\Delta_2}\int_{-\pi/\Delta_1}^{\pi/\Delta_1}[H_0 - H_1H_2]e^{j(\omega_1 n_1\Delta_1+\omega_2 n_2\Delta_2)}\,d\omega_1\,d\omega_2$$

$$= \frac{\Delta_1\Delta_2}{(2\pi)^2}\int_{-\Omega_{U2}}^{\Omega_{U2}}\int_{-\Omega_{U1}}^{\Omega_{U1}} e^{j(\omega_1 n_1\Delta_1+\omega_2 n_2\Delta_2)}\,d\omega_2\,d\omega_1$$

$$-\frac{\Delta_1\Delta_2}{(2\pi)^2}\left[\int_{-\Omega_{U1}}^{-\Omega_{L1}} e^{j\omega_1 n_1\Delta_1}d\omega_1 + \int_{\Omega_{L1}}^{\Omega_{U1}} e^{j\omega_1 n_1\Delta_1}d\omega_1\right]\left[\int_{-\Omega_{U2}}^{-\Omega_{L2}} e^{j\omega_2 n_2\Delta_2}d\omega_2 + \int_{\Omega_{L2}}^{\Omega_{U2}} e^{j\omega_2 n_2\Delta_2}d\omega_2\right]$$

Evaluating these integrals gives

$$\frac{\Delta_1\Delta_2}{(2\pi)^2}\left\{\left[\frac{e^{j\omega_1 n_1\Delta_1}}{jn_1\Delta_1}\right]_{-\Omega_{U1}}^{\Omega_{U1}}\left[\frac{e^{j\omega_2 n_2\Delta_2}}{jn_2\Delta_2}\right]_{-\Omega_{U2}}^{\Omega_{U2}}\right\}$$

$$-\frac{\Delta_1\Delta_2}{(2\pi)^2}\left\{\left[\left[\frac{e^{j\omega_1 n_1\Delta_1}}{jn_1\Delta_1}\right]_{-\Omega_{U1}}^{-\Omega_{L1}} + \left[\frac{e^{j\omega_1 n_1\Delta_1}}{jn_1\Delta_1}\right]_{\Omega_{L1}}^{\Omega_{U1}}\right]\left[\left[\frac{e^{j\omega_2 n_2\Delta_2}}{jn_2\Delta_2}\right]_{-\Omega_{U2}}^{-\Omega_{L2}} + \left[\frac{e^{j\omega_2 n_2\Delta_2}}{jn_2\Delta_2}\right]_{\Omega_{L2}}^{\Omega_{U2}}\right]\right\}$$

$$= \frac{\Delta_1\Delta_2}{(2\pi)^2}\left\{2\Omega_{U1}2\Omega_{U2}\mathrm{sinc}(n_1\Delta_1\Omega_{U1})\mathrm{sinc}(n_2\Delta_2\Omega_{U2})\right\}$$

$$-\frac{\Delta_1\Delta_2}{(2\pi)^2}\left\{\left[2\Omega_{U1}\mathrm{sinc}(n_1\Delta_1\Omega_{U1}) - 2\Omega_{L1}\mathrm{sinc}(n_1\Delta_1\Omega_{L1})\right]\left[2\Omega_{U2}\mathrm{sinc}(n_2\Delta_2\Omega_{U2}) - 2\Omega_{L2}\mathrm{sinc}(n_2\Delta_2\Omega_{L2})\right]\right\}$$

$$= \frac{\Delta_1\Delta_2}{(\pi)^2}\left\{\Omega_{U1}\Omega_{L2}\mathrm{sinc}(n_1\Delta_1\Omega_{U1})\mathrm{sinc}(n_2\Delta_2\Omega_{L2}) + \right.$$

$$\left.\Omega_{L1}\Omega_{U2}\mathrm{sinc}(n_1\Delta_1\Omega_{L1})\mathrm{sinc}(n_2\Delta_2\Omega_{U2}) - \Omega_{L1}\Omega_{L2}\mathrm{sinc}(n_1\Delta_1\Omega_{L1})\mathrm{sinc}(n_2\Delta_2\Omega_{L2})\right\}$$

 (cont.

It is also possible to arrive at the above by using standard results for a rectangular lowpass and bandpass filters.
Standard result for a lowpass filter (H_0) is:

$$h(n_1\Delta_1, n_2\Delta_2) = \frac{\Delta_1\Delta_2}{\pi^2}[\Omega_{U2}\,\Omega_{U1}\,\mathrm{sinc}(\Omega_{U2}n_2\Delta_2)\,\mathrm{sinc}(\Omega_{U1}n_1\Delta_1)]$$

Standard result for a separable bandpass filter (H_1H_2) is

$$h(n_1\Delta_1, n_2\Delta_2) =$$
$$\frac{\Delta_1\Delta_2}{\pi^2}\,[\Omega_{U1}\mathrm{sinc}(\Omega_{U1}n_1\Delta_1) - \Omega_{L1}\mathrm{sinc}(\Omega_{L1}n_1\Delta_1)]\,[\Omega_{U2}\mathrm{sinc}(\Omega_{U2}n_2\Delta_2) - \Omega_{L2}\mathrm{sinc}(\Omega_{L2}n_2\Delta_2)]$$

As well as taking $(H_0 - H_1H_2)$ we can also treat the shaded region as the sum of lowpass filters ($|\omega_1| < \Omega_{U1}$ and $|\omega_2| < \Omega_{L2}$, $|\omega_1| < \Omega_{L1}$ and $|\omega_2| < \Omega_{U2}$) minus another lowpass filter ($|\omega_1| < \Omega_{L1}$ and $|\omega_2| < \Omega_{L2}$.

[35%]

2 (a) (i) An expression relating the true image to the observed image in the continuous case is

$$y(u_1,u_2) = \int\int h(v_1,v_2)x(u_1-v_1,u_2-v_2)dv_1dv_2 + d(u_1,u_2) \qquad (1)$$

where $h(v_1,v_2)$ is the point-spread function of the distorting system (if the distortion is assumed linear).
We can then write this equation in discrete form as

$$y(n_1,n_2) = \sum_{m_1}\sum_{m_2} h(m_1,m_2)x(n_1-m_1,n_2-m_2) + d(n_1,n_2)$$

[5%]

(ii) If we then neglect the noise in the equation of part (i), we are left with

$$y(n_1,n_2) = \sum_{m_1}\sum_{m_2} h(m_1,m_2)x(n_1-m_1,n_2-m_2)$$

Since the relationship between x and y is a 2-D convolution, a straightforward approach to the problem of reconstruction is to take the Fourier transform of each side of the above to give:

$$Y(\omega_1,\omega_2) = H(\omega_1,\omega_2)X(\omega_1,\omega_2)$$

where: $H(\omega_1,\omega_2) = \sum_{n_2=-\infty}^{\infty}\sum_{n_1=-\infty}^{\infty} h(n_1,n_2)e^{-j(\omega_1 n_1+\omega_2 n_2)}$

$$\therefore X(\omega_1,\omega_2) = \frac{Y(\omega_1,\omega_2)}{H(\omega_1,\omega_2)} \text{ and } x(n_1,n_2) = \frac{1}{(2\pi)^2}\int_{-\pi}^{\pi}\int_{-\pi}^{\pi} X(\omega_1,\omega_2)e^{j(\omega_1 n_1+\omega_2 n_2)}d\omega_1 d\omega_2$$

Thus, if we neglect noise and know the psf, h, we can estimate our true image by a process known as *inverse filtering*, which, as we see above, involves dividing the fourier transform of the observed image by the fourier transform of h – the inverse filter is therefore $1/H$.

[10%]

(iii) If the transfer function $H(\omega_1,\omega_2)$ has zeros then the inverse filter, $1/H$, will have infinite gain. i.e. when $H(\omega_1,\omega_2)$ is very small, $1/H(\omega_1,\omega_2)$ is very large and therefore, small noise in the regions of the frequency plane where $1/H(\omega_1,\omega_2)$ is very large, can be hugely amplified. In practice a method of lessening this sensitivity to noise is to threshold the frequency response, leading to the so-called, *pseudo-inverse* or *generalised inverse filter* $H_g(\omega_1,\omega_2)$. This is given by

$$H_g(\omega_1,\omega_2) = \begin{cases} \frac{1}{H(\omega_1,\omega_2)} & \frac{1}{|H(\omega_1,\omega_2|} < \gamma \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

or

$$H_g(\omega_1,\omega_2) = \begin{cases} \frac{1}{H(\omega_1,\omega_2)} & \frac{1}{|H(\omega_1,\omega_2|} < \gamma \\ \gamma\frac{|H(\omega_1,\omega_2|}{H(\omega_1,\omega_2)} & \text{otherwise} \end{cases} \tag{3}$$

Clearly for $\frac{1}{|H(\omega_1,\omega_2|} \geq \gamma$ in equation 3, the modulus of the filter is set as γ, whereas in equation 2 it is set as 0.

[5%]

(iv) Our observed image, y, the original image, x, the linear distortion L, and the noise d, are related by

$$y(\mathbf{n}) = Lx(\mathbf{n}) + d(\mathbf{n})$$

Writing this equation in vector form – i.e. we write $\mathbf{x}$ for the vector of original image values etc.

$$\mathbf{y} = L\mathbf{x} + \mathbf{d}$$

For simplicity we assume that $E[\mathbf{x}] = 0$ and $E[\mathbf{d}] = 0$, i.e. that both the signal and the noise are zero mean. To find an estimate of $\mathbf{x}$, we maximise $P(\mathbf{x}|\mathbf{y})$, i.e. the *probability* of the original image *given* the observed data. When dealing with *conditional probabilities* we use *Bayes' Theorem*:

$$P(\mathbf{x}|\mathbf{y}) = \frac{1}{P(\mathbf{y})}P(\mathbf{y}|\mathbf{x})P(\mathbf{x}) \tag{4}$$

at the simplest level we regard $P(\mathbf{y})$, the probability of the data, simply as a normalising factor, which therefore implies that we wish to maximise

$$P(\mathbf{x}|\mathbf{y}) \propto P(\mathbf{y}|\mathbf{x})P(\mathbf{x}) \tag{5}$$

If we assume that the noise is gaussian distributed we can write the probability of the noise, which is proportional to the likelihood as

$$P(\mathbf{y}|\mathbf{x}) \propto e^{-\frac{1}{2}\mathbf{d}^T N^{-1}\mathbf{d}} = e^{-\frac{1}{2}(\mathbf{y}-L\mathbf{x})^T N^{-1}(\mathbf{y}-L\mathbf{x})}$$

where $N = E[\mathbf{d}\mathbf{d}^T]$ is the noise covariance matrix. The $\mathbf{d}^T N^{-1}\mathbf{d}$ term is the vector equivalent of the $\frac{1}{\sigma}$ term in the 1d gaussian – if N is diagonal then N^{-1} will be diagonal with elements $\frac{1}{\sigma_i}$.
We now have to decide on the assignment of the *prior* probability $P(\mathbf{x})$ – this probability incorporates any *prior* knowledge we may have about the distribution of the data.

Assume first an ideal world in which $\mathbf{x}$ is a gaussian random variable, described by a *known* covariance matrix $C = E[\mathbf{x}\mathbf{x}^T]$ (including all cross-correlations etc.) so that

$$P(\mathbf{x}) \propto \mathrm{e}^{-\frac{1}{2}\mathbf{x}^T C^{-1}\mathbf{x}}$$

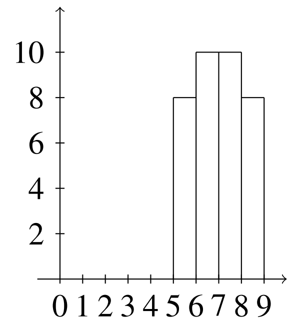
Thus we can now write the posterior probability as

$$P(\mathbf{x}|\mathbf{y}) \propto P(\mathbf{y}|\mathbf{x})P(\mathbf{x}) \propto \mathrm{e}^{-\frac{1}{2}[(\mathbf{y}-L\mathbf{x})^T N^{-1}(\mathbf{y}-L\mathbf{x})+\mathbf{x}^T C^{-1}\mathbf{x}]} \tag{6}$$

which one must maximise wrt $\mathbf{x}$ to obtain the reconstruction.

[30%]

(b) (i) The histogram of the image is shown below. We can see that the grey levels used are concentrated around the upper end of the range 1-9.



[10%]

(ii) It often helps to draw up a table when performing histogram equalisation: below let $H(i)$ be the frequency values and $C(i)$ be the cumulative frequency values

i	1	2	3	4	5	6	7	8	9
$H(i)$	0	0	0	0	0	8	10	10	8
$C(i)$	0	0	0	0	0	8	18	28	36

The transformed levels are given by

$$y_k = \sum_{i=1}^{k} L\frac{N_i}{NM}, \quad k = 1 \ldots 9$$

where $N \times M$ are the dimensions of the image, N_i is the number of pixels in grey level i (equivalent to $H(i)$ above) and L is the range in grey level space. Therefore, $L = 9$, $NM = 36$ and

$$y_k = \frac{L}{NM}\sum_{i=1}^{k} N_i = \frac{1}{4}\sum_{i=1}^{k} N_i = \frac{1}{4}C(k), \quad k = 1 \ldots 9$$

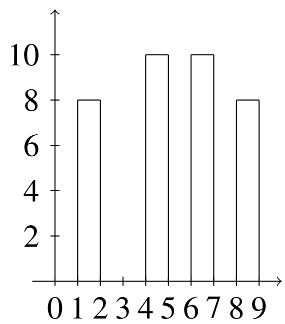
We can now add an extra line to our table to show the transformed values:

i	1	2	3	4	5	6	7	8	9
$H(i)$	0	0	0	0	0	8	10	10	8
$C(i)$	0	0	0	0	0	8	18	28	36
$y(i)$	0	0	0	0	0	2	4.5	7	9

From this table it is now easy to draw the new image and sketch the new histogram (round up the 4.5 value to 5)

2	2	5	7	9	9
2	2	5	7	9	9
5	5	5	7	7	7
7	7	7	5	5	5
9	9	7	5	2	2
9	9	7	5	2	2

Fig. 1



The process has succeeded in spreading out the grey levels more evenly across the scale but the distribution is far from being uniform. The discreteness of the problem means that the equalisation process tries to do the best job it can according to the rules prescribed. This could now be improved by interpolation.

[30%]

(iii) The original image is effectively two peaks – in the top right and bottom left corners, but the difference between these peaks and the intervening 'valley' is not pronounced. With the new histogram equalised image, the contrast between the peaks and the separating valley is much more pronounced.

[10%]

3 (a) **Orthogonality**: First consider

$$\mathbf{t}_1 \cdot \mathbf{t}_k = \frac{\sqrt{2}}{n} \sum_{i=1}^{n} \cos\left(\frac{\pi(2i-1)(k-1)}{2n}\right)$$

for $2 \le k \le n$.

The angles $\frac{\pi(2i-1)(k-1)}{2n}$ correspond to the centres of n sectors, uniformly spaced around the unit circle, covering the range 0 to $(k-1)\pi$.

Hence the cos terms are proportional to the projections of the sector centres of gravity onto the real axis. From symmetry considerations, these projections onto the real axis will sum to zero (pairs will cancel out) for all integer $(k-1)$ as long as $(k-1)$ is non-zero and not an integer multiple of $2n$.

For example, these sectors are shown here for $k = 2$ and $n = 6$:

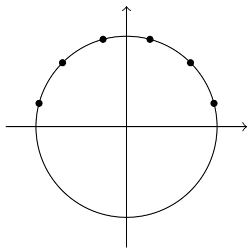


Fig. 2

Other proofs are possible, but are less intuitive.

Now consider the inner product of rows excluding the first:

$$\mathbf{t}_l \cdot \mathbf{t}_k = \frac{2}{n} \sum_{i=1}^{n} \cos\left(\frac{\pi(2i-1)(l-1)}{2n}\right) \cos\left(\frac{\pi(2i-1)(k-1)}{2n}\right)$$

We can expand each term as the sum of two cosines ($\cos(A+B)$, $\cos(A-B)$). As long as $(l+k-2)$ is not zero and not a multiple of $2n$, the first sum will be zero.

As long as $(l-k)$ is not zero and not a multiple of $2n$, the second sum will be zero.

But l and k only go from 2 to n, and $l \neq k$, so all rows are orthogonal.

Orthonormality:

$\sum_{i=1}^{n} t_{1i}^2 = n.\frac{1}{n} = 1$, therefore row 1 is normalised (has unit magnitude).

Also

$$\sum_{i=1}^{n} t_{ki}^2 = \frac{2}{n} \sum_{i=1}^{n} \cos^2\left(\frac{\pi(2i-1)(k-1)}{2n}\right) = \frac{1}{n} \sum_{i=1}^{n} \left[1 + \cos\left(\frac{2\pi(2i-1)(k-1)}{2n}\right)\right]$$

$$= n.\frac{1}{n} + 0 = 1$$

 (cont.

which tells us that rows 2 to n are also normalised.

Because each row of T times any other row of T sums to zero, and each row times itself sums to 1, we have $TT^T = I$, which tells us that $T^T = T^{-1}$ so that $TT^T = TT^{-1} = I$ [20%]

(b) The energy of X is the sum of squares of all elements of X, and similarly for Y. Consider $\mathbf{z} = T\mathbf{x}$, where $\mathbf{z}$ and $\mathbf{x}$ are n-element column vectors from Z and X.

The energy of Z is $\mathbf{z}^T\mathbf{z} = \mathbf{x}^T T^T T\mathbf{x} = \mathbf{x}^T\mathbf{x}$, which therefore equals the energy of X.

Therefore, if $Z = TX$, where Z and X are $n \times n$ matrices, then the energy of Z is equal to the energy of X.

Similarly, if $Y = ZT^T$, we also find that the energy of Y is equal to the energy of Z, since row energies of Z are preserved in Y.

Therefore Energy of $X =$ Energy of Y. [15%]

(c) For the 16 subimages, $i = 1...4$ and $j = 1...4$, and so we have the following energies, where α is a constant of proportionality, ie we assume that $E_{ij} = \alpha/(i+j-1)^2$:

(i,j)	energy per subimage
$(1,1)$	α
$(1,2),(2,1)$	$\alpha/4$
$(1,3),(2,2),(3,1)$	$\alpha/9$
$(1,4),(2,3),(3,2),(4,1)$	$\alpha/16$
$(2,4),(3,3),(4,2)$	$\alpha/25$
$(3,4),(4,3)$	$\alpha/36$
$(4,4)$	$\alpha/49$

Total energy is therefore:

$$= \alpha + \frac{2\alpha}{4} + \frac{3\alpha}{9} + \frac{4\alpha}{16} + \frac{3\alpha}{25} + \frac{2\alpha}{36} + \frac{\alpha}{49} = \alpha\left[1 + \frac{2}{4} + \frac{3}{9} + \frac{4}{16} + \frac{3}{25} + \frac{2}{36} + \frac{1}{49}\right] = 2.2793\alpha$$

Since energy is preserved in Y, $2.2793\alpha = \sigma^2$, so $\alpha = 0.4387\sigma^2$.

Hence the energies per coefficient of the 7 types of subband above are

$$0.4387\sigma^2,\ 0.1097\sigma^2,\ 0.0487\sigma^2,\ 0.0274\sigma^2,\ 0.0175\sigma^2,\ 0.0122\sigma^2,\ 0.0090\sigma^2,$$

[30%]

(d) Using the given formula for entropy and $Q = \sigma/2$, we get

$$H_{ij} = \frac{1}{2}\log_2\left(1 + \left[\frac{20}{\sigma^2/4}\right] E_{ij}\right)$$

where the E_{ij} are the values from part (c). Hence the 7 distinct H_{ij}s are given by

H_{11}	2.5869 bit
$H_{12} = H_{21}$	1.6445 bit
$H_{13} = H_{22} = H_{31}$	1.1464 bit
$H_{14} = H_{23} = H_{33} = H_{41}$	0.8376 bit
$H_{24} = H_{33} = H_{42}$	0.6327 bit
$H_{34} = H_{43}$	0.4909 bit
H_{44}	0.3896 bit

Each subband is of size $\frac{3072}{4} \times \frac{2048}{4} = 768 \times 512 = 393,216$ coefficients.
Therefore, the total number of bits needed to code the image is

$$= 393216[H_{11} + 2H_{12} + 3H_{13} + 4H_{14} + 3H_{24} + 2H_{34} + H_{44}] = 393216 \times 15.9351 = 6.266 \times 10^6 \text{ bits}$$

. [20%]

(e) In JPEG XR, the 4×4 DCTs on the pixels are followed by a second level of DCTs which are applied just to the subimage of the DC coefficients, Y_{11}. The other 15 subimages, $Y_{12}, ..., Y_{44}$, are left as they are from the level 1 DCTs. This gives good compression of the low frequency components of the image where the main energy is concentrated, since the 16 subimages from the level 2 DCT are approximately equivalent to the top-left 16 subimages of a single 16×16 point DCT system.

However JPEG XR avoids the main disadvantage of a 16×16 DCT, because the 'blockiness' is mainly confined to the 4×4 pixel blocks from level 1, rather than the 16×16 pixel blocks from level 2. Hence the blockiness is less visible in JPEG XR than either 8×8 JPEG or a 16×16 transform.

[15%]

4 (a) See Figure 3, where the left hand side is the Analysis filter bank and the right hand side is the Reconstruction filter bank.

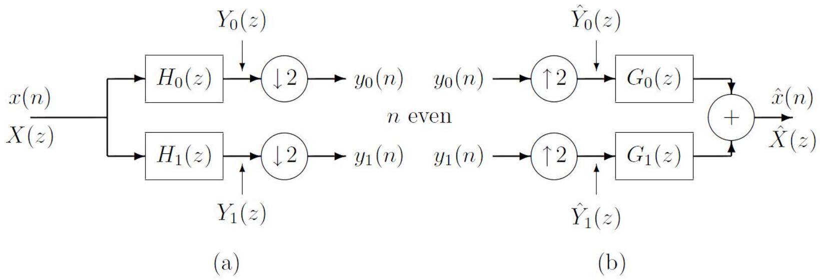


Fig. 3

$H_0(z)$ and $G_0(z)$ are lowpass filters and H_1 and G_1 are highpass filters.

Perfect Reconstruction means that we require that the reconstructed signal $\hat{X}(z)$ is identical to the input signal $X(z)$.

[20%]

(b) the figures below show the 2-level wavelet transforms for 1D signals and 2D signals;

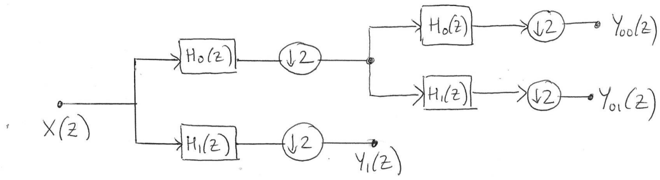


Fig. 4

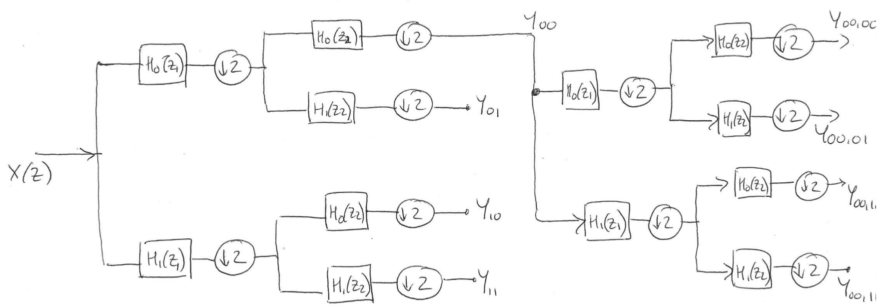


Fig. 5

For 2D filtering, the lowpass and highpass filters (H_0 and H_1) are applied first to the rows of the input image x, and then to the columns of the resulting pair of subimages, to give 4 subimages, $y_{00},...,y_{11}$, each one being one quarter of the size of the input image. Then the Lo-Lo subimage, y_{00}, is passed through a second level of filters in each direction.

[20%]

(c) Separable filters are much more efficient to compute than non-separable filters. If the 1D filters are of length k, then the level 1 row convolutions require a total of Nk multiply-and-add operations, where N is the number of pixels in the image. Similarly the level 1 column convolutions also require Nk operations ($Nk/4$ for each output subimage). So level 1 requires $2Nk$ operations in total.

If instead we used 4 separate 2D filters each of area $k \times k$ coefficients, then this would require Nk^2 operations in total ($Nk^2/4$ for each output subimage).

Hence using two 1D filtering processes require $\frac{2Nk}{Nk^2} = \frac{2}{k}$ times as much computation as a single 2D filtering process for each subimage, which provides a useful saving when k is around 10 to 20.

Level 2 wavelet subimages are only $1/4$ the size of level 1 subimages, so only $1/4$ of the computation is needed at level 2. Similarly, only $1/16$ of the computation of level 1 is needed at level 3 etc. So the total computation for a wavelet transform of many layers is

$$2Nk(1+1/4+1/16+1/64+...) < 2Nk\,\frac{4}{3}$$

Since $\sum_{n=1}^{\infty} \frac{1}{n^2} = 4/3$.

This is therefore very efficient and linear in N. [20%]

(d) Substituting the expressions for H_1 and G_1 in to the P-R equation gives

$$G_0(z)H_0(z) + H_0(-z)G_0(-z) = 2$$

or

$$P(z) + P(-z) = 2$$

where $P(z) = H_0(z)G_0(z)$. If $P(z)$ is a polynomial, the PR conditions therefore requires all terms in even powers of z to be zero (except the z^0 term which should be 1).

When z is replaced by $-z$ in $Z = \frac{1}{2}(z + z^{-1})$ we get $\frac{1}{2}(-z + -z^{-1}) = -Z$. Note that odd powers of Z produce only odd powers of z, and even powers of Z produce only even powers of z – essential for the PR conditions. By writing the PR equation in terms of Z

instead of z we are able to more easily solve a polynomial equation (no negative powers) in terms of Z.

Here the P-R equation, in terms of Z becomes:

$$(1+Z)(1+aZ)(1+Z)+(1-Z)(1-aZ)(1-Z)=2$$

Now

$$(1+Z)^2(1+aZ)=(1+2Z+Z^2)(1+aZ)=1+(2+a)Z+(1+2a)Z^2+aZ^3$$

When Z is replaced by $-Z$, the odd-order terms get negated, and therefore cancel out in the P-R equation, which then simplifies to

$$2[1+(1+2a)Z^2]=2$$

So that $(1+2a)=0$ and $a=-1/2$. [20%]

(e) When $z=-1$, $Z=(1/2)(z+z^{-1})=-1$ also. Hence each factor $(1+Z)$ produces 2 zeros at $z=-1$, because

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

.

If we want more zeros in Z at $z=-1$, for each factor of $(1+Z)$ in the transformed filters, then we need a higher-order substitution for Z. In order that odd powers of Z produce **only** odd powers of z in the filter (to satisfy the P-R condition) the next higher order subsitution for Z that is symmetric about z^0 is

$$Z=pz^3+(1/2-p)(z+z^{-1})+pz^{-3}$$

where p is a design parameter that can be chosen to put as many zeros as possible (typically 4) at $z=-1$, when $Z=-1$. [20%]

END OF PAPER

THIS PAGE IS BLANK