4F8 Dr N Kingsbury ngk@eng.cam.ac.uk

Module 4F8 – Image Processing and Image Coding: April 2006

Solutions

1. (a) An expression relating the true image to the observed image is

$$y(u_1, u_2) = \int\int h(v_1, v_2)x(u_1 - v_1, u_2 - v_2)dv_1 dv_2 + d(u_1, u_2) \qquad (1)$$

where $h(v_1, v_2)$ is the point-spread function of the distorting system (if the distortion is assumed linear). [15%]

(b) We can write equation 1 in discrete form as

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

If we then neglect the noise, 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:

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

$$\therefore X(\omega_1, \omega_2) = \frac{Y(\omega_1, \omega_2)}{H(\omega_1, \omega_2)}$$

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 $1/H$. [20%]

(c) 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 (or indeed infinite if there are zeros) 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

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

or

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

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. [15%]

(d) The Wiener filter is optimal in the sense that it is the linear, spatially invariant filter which minimises the expectation of the squared error between the true and the reconstructed images.

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*:

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

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

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

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

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

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 $\frac{d^2}{\sigma^2}$ term in the 1-D gaussian – if N is diagonal then N^{-1} will be diagonal with elements $\frac{1}{\sigma_i^2}$.

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

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

which one must maximise wrt $\mathbf{x}$ to obtain the reconstruction. [35%]

(e) The Wiener filter is easy to calculate and has known reconstruction errors. However, it is certainly not the best filter for real problems. It depends on the *assumption of gaussianity* and knowledge of the covariance structure *a priori* – often gaussian assumptions are not correct and we can only make a poor guess at the covariance structure. We are therefore forced to consider alternative *priors*. One such prior which has been widely and successfully used is the *entropy prior*.

$$P(\mathbf{x}) \propto e^{\alpha S}$$

where the *entropy* S of the image (this form is often referred to as the *cross entropy*) is given by

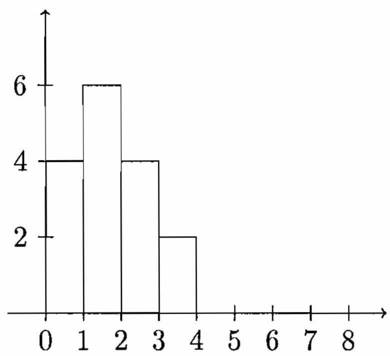
$$S(\mathbf{x}, \mathbf{m}) = \sum_i \left[x_i - m_i - x_i \ln \left(\frac{x_i}{m_i} \right) \right]$$

where $\mathbf{m}$ is the *measure* on an image space (*the model*) to which the image $\mathbf{x}$ defaults in the absence of data. (By differentiating term by term, we can see that the global maximum of S occurs at $\mathbf{x} = \mathbf{m}$.) [15%]

2. (a) (i) A histogram plot of frequency of occurrence of grey levels in an image against grey level will tell us how much the available grey levels are used. An intuitively appealing idea would be to apply a transformation or mapping to the image pixels in such a way that the probability of occurrence of the various grey levels should be constant i.e. all grey levels are equiprobable which would correspond to a constant amplitude histogram. This process is called *histogram equalisation*.

Histogram equalisation is often useful in bringing out detail in images which make poor use of the available grey levels – this may occur due to poor illumination of the scene or non-linearity in the imaging system. [15%]

(ii) The histogram of the image in figure 1 is shown below. We can see that the grey levels used are concentrated around the lower end of the range 1-8.



[10%]

(iii) 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
$H(i)$	4	6	4	2	0	0	0	0
$C(i)$	4	10	14	16	16	16	16	16

The transformed levels are given by

$$y_k = \sum_{i=1}^{k} L \frac{N_i}{NM}, \qquad k = 1...8$$

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, $L = 8$, $NM = 16$ and

$$y_k = \frac{L}{NM} \sum_{i=1}^{k} N_i = \frac{1}{2} \sum_{i=1}^{k} N_i = \frac{1}{2} C(k), \qquad k = 1...8$$

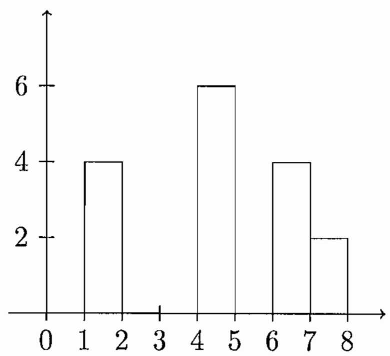
We can now add an extra line to our table to show the transformed values:

i	1	2	3	4	5	6	7	8
$H(i)$	4	6	4	2	0	0	0	0
$C(i)$	4	10	14	16	16	16	16	16
$y(i)$	2	5	7	8	8	8	8	8

From this table it is now easy to draw the new image and sketch the new histogram

2	2	5	5
7	5	8	7
5	8	7	5
5	7	2	2

Figure 1:



We can see from the new histogram that the process has succeeded in spreading out the grey levels more evenly across the scale but that 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. [25%]

(b) (i) We are told that black= 0 and white= A. The 2d fourier transform of the image in fig. 2 is given by

$$G(\omega_1,\omega_2) = \int\int g(u_1,u_2)e^{-j(\omega_1 u_1+\omega_2 u_2)}du_1 du_2$$

Since g is only non-zero, and constant, in the diamond-shaped region, S, we can write the above as

$$G(\omega_1,\omega_2) = A\int\int_S e^{-j(\omega_1 u_1+\omega_2 u_2)}du_1 du_2$$

In order to perform the integration easily we try to transform the diamond to a square or rectangle in some different coordinate system (u_1', u_2'). In the 4 quadrants the border of the shape are given by the equations:

1st quadrant: $u_1 + \alpha u_2 = a_1$
2nd quadrant: $u_1 - \alpha u_2 = -a_1$
3rd quadrant: $u_1 + \alpha u_2 = -a_1$
4th quadrant: $u_1 - \alpha u_2 = a_1$

where $\alpha = a_1/a_2$. Thus, consider the coordinate change:

$$u_1' = u_1 - \alpha u_2, \quad u_2' = u_1 + \alpha u_2$$

The diamond therefore maps onto a square of side length $2a_1$ centred on the origin. The FT integral therefore becomes

$$G(\omega_1,\omega_2) = A\int_{u_1'=-a_1}^{a_1}\int_{u_2'=-a_1}^{a_1} e^{-\frac{j}{2}\omega_1(u_1'+u_2')}\ e^{-\frac{j}{2\alpha}\omega_2(u_2'-u_1')}\ |J|\ du_1'\ du_2'$$

where $|J|$ is the Jacobian of the transformation. From the given coordinate transformation we have that

$$\frac{1}{|J|} = \begin{vmatrix} \frac{\partial u_1'}{\partial u_1} & \frac{\partial u_1'}{\partial u_2} \\ \frac{\partial u_2'}{\partial u_1} & \frac{\partial u_2'}{\partial u_2} \end{vmatrix} = \begin{vmatrix} 1 & -\alpha \\ 1 & \alpha \end{vmatrix} = 2\alpha$$

Therefore

$$G(\omega_1,\omega_2) = \frac{A}{2\alpha}\int_{u_1'=-a_1}^{a_1}\int_{u_2'=-a_1}^{a_1} e^{-\frac{j}{2}u_1'(\omega_1-\frac{1}{\alpha}\omega_2)}\ e^{-\frac{j}{2}u_2'(\omega_1+\frac{1}{\alpha}\omega_2)}\ du_1'\ du_2'$$

which can be written as

$$G(\omega_1,\omega_2) = \frac{A}{2\alpha}\int_{u_1'=-a_1}^{a_1}\int_{u_2'=-a_1}^{a_1} e^{-ju_1'\omega_1'}\ e^{-ju_2'\omega_2'}\ du_1'\ du_2'$$

where $\omega_1' = \frac{1}{2}(\omega_1 - \frac{1}{\alpha}\omega_2)$ and $\omega_2' = \frac{1}{2}(\omega_1 + \frac{1}{\alpha}\omega_2)$

This is now easy to integrate, so:

$$\begin{aligned} G(\omega_1,\omega_2) &= \frac{A}{2\alpha}\left[\frac{e^{-ju_1'\omega_1'}}{-j\omega_1'}\right]_{-a_1}^{a_1}\left[\frac{e^{-ju_2'\omega_2'}}{-j\omega_2'}\right]_{-a_1}^{a_1} \\ &= \frac{Aa_2}{2a_1}\left[\frac{e^{ja_1\omega_1'} - e^{-ja_1\omega_1'}}{j\omega_1'}\right]\left[\frac{e^{ja_1\omega_2'} - e^{-ja_1\omega_2'}}{j\omega_2'}\right] \\ &= 2Aa_1a_2\ \mathrm{sinc}(a_1\omega_1')\ \mathrm{sinc}(a_1\omega_2') \\ &= 2Aa_1a_2\ \mathrm{sinc}(\frac{a_1\omega_1 - a_2\omega_2}{2})\ \mathrm{sinc}(\frac{a_1\omega_1 + a_2\omega_2}{2}) \end{aligned}$$

Thus we see that this case is indeed very similar to the case of a rectangle centred with sides parallel to the u_1, u_2 axes – we simply 'redefine' our frequency axes. [35%]

(ii) The first zeros of the sinc functions occur at $\omega_1' = \frac{\pi}{a_1}$ and $\omega_2' = \frac{\pi}{a_1}$; one method of measuring *bandwidth* would be to take these main lobes of the sinc functions. Thus, the bandwidths can be taken to be $\frac{\pi}{a_1}$ in both the ω_1' and ω_2' directions (noting that these are rotated and scaled compared to the ω_1 and ω_2 directions).

Given the above measures of bandwidth, we can say that we should sample the image (in the ω_1' and ω_2' directions) at

$$\Omega_1' > 2\frac{\pi}{a_1} \qquad \text{and} \qquad \Omega_2' > 2\frac{\pi}{a_1}$$

to avoid aliasing effects. [15%]

3. (a) When $Y = TXT^T$, the left-hand pre-multiplication by T operates on the columns of matrix X, while the right-hand post-multiplication by T^T operates on the rows of the result.

Looking first at the rows of Y, we can rewrite it as

$$Y = \frac{1}{2}\begin{bmatrix} (a+c)+(b+d) & (a+c)-(b+d) \\ (a-c)+(b-d) & (a-c)-(b-d) \end{bmatrix} = \frac{1}{2}\begin{bmatrix} (a+c) & (b+d) \\ (a-c) & (b-d) \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

Then, looking at the columns of the result:

$$Y = \frac{1}{2}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = T\ X\ T^T$$

where

$$T = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

The Haar transform is simple because it only involves four additions and four subtractions for the complete 2-D transform process on 4 pixels. It is effective because there is strong positive correlation both vertically and horizontally between adjacent pairs of pixels, so we find that most of the image energy is concentrated in the top left coefficient of each transformed block and much less energy is in the other three coefficients (especially the lower right one). Hence the transform achieves good compression of energy. Even greater compression can be achieved if the transform is applied again (and again) to the top left subband. [30%]

(b) For a matrix to be orthonormal, the dot product between any pair of distinct rows must be zero, and the energy (L_2-norm) of each row must be unity. For T:

$$(\text{row 1}) \,.\, (\text{row 2}) = \tfrac{1}{2}(1\,.\,1 + 1\,.\,(-1)) = 0$$

and

$$(\text{row 1}) \,.\, (\text{row 1}) = \tfrac{1}{2}(1\,.\,1 + 1\,.\,1) = 1$$

and

$$(\text{row 2}) \,.\, (\text{row 2}) = \tfrac{1}{2}(1\,.\,1 + (-1)\,.\,(-1)) = 1$$

Hence T is orthonormal.

This is an important property because it means that $T^{-1} = T^T$, so that the inverse of the transform may be easily calculated. In addition energy is preserved between the image domain and the transform domain, which means that techniques which minimise the mean squared error from quantisation in the transform domain will also minimise the mean squared error in the image domain. [20%]

(c) For the 4×4 transform to be orthonormal:

By inspection, the dot product of any pair of distinct rows is zero for any value of p, q, r, because of the particular arrangement of positive and negative terms in the given matrix which means that every positive product is cancelled out by an equivalent negative product.

In order for rows 1 and 3 to have unit energy, $4p^2 = 1$. Therefore $p = \frac{1}{2}$.

In order for rows 2 and 4 to have unit energy, $2q^2 + 2r^2 = 1$. Hence if $r/q = \tan(\pi/8)$:

$$2q^2(1 + \tan^2(\frac{\pi}{8})) = \frac{2q^2}{\cos^2(\frac{\pi}{8})} = 1$$

So

$$q = \frac{1}{\sqrt{2}}\cos(\frac{\pi}{8}) = 0.6533 \qquad \text{and} \qquad r = q\tan(\frac{\pi}{8}) = \frac{1}{\sqrt{2}}\sin(\frac{\pi}{8}) = 0.2706$$

Row 2 of the transform can be expressed as:

$$[q \quad r \quad -r \quad -q] = \frac{1}{\sqrt{2}}[\cos(\frac{\pi}{8}) \quad \cos(\frac{3\pi}{8}) \quad \cos(\frac{5\pi}{8}) \quad \cos(\frac{7\pi}{8})]$$

and row 4 can be expressed as:

$$[r \quad -q \quad q \quad -r] = \frac{1}{\sqrt{2}}[\cos(\frac{3\pi}{8}) \quad \cos(\frac{9\pi}{8}) \quad \cos(\frac{15\pi}{8}) \quad \cos(\frac{21\pi}{8})]$$

Hence row 2 represents samples from half a cycle of a cosine wave, and row 4 represents samples from 1.5 cycles of a cosine wave. Similarly row 3 represents a whole cycle and row 1 represents a constant value (i.e. a cosine of zero frequency). Thus the transform is a discrete cosine transform or DCT. The ratio r/q was specified as $\tan(\pi/8)$ so that the second and first terms of row 2 would be in the ratio $\cos(3\pi/8)/\cos(\pi/8)$, and thus be correct for a DCT. [30%]

(d) It was demonstrated in the lecture course that an 8×8 DCT produced better energy compression and hence lower entropy than a 4×4 or 2×2 DCT. For a given bit rate, the visual quality of the 8×8 DCT is also best.

It was also shown that a 16×16 DCT could produce even better energy compression and lower entropy, compared with the 8×8 DCT, but only by a small amount. However for a given bit rate, the visual quality of the 16×16 DCT is not quite as good as the 8×8 DCT, because the blocks are larger and discontinuities at their boundaries due to quantisation are more noticeable. Hence 8×8 was selected as the optimum transform size for JPEG coding.

Wavelet-based coders employ the multi-scale discrete wavelet transform (DWT), which uses small regions for representing the high frequency components of the image and progressively larger regions for representing the lower frequency components. Hence the region size automatically adapts to the level of detail present in the image and there is no need to define a fixed block size. In addition the wavelet basis functions decay smoothly to zero at their extremities, so boundary discontinuities do not occur. [20%]

4. (a)

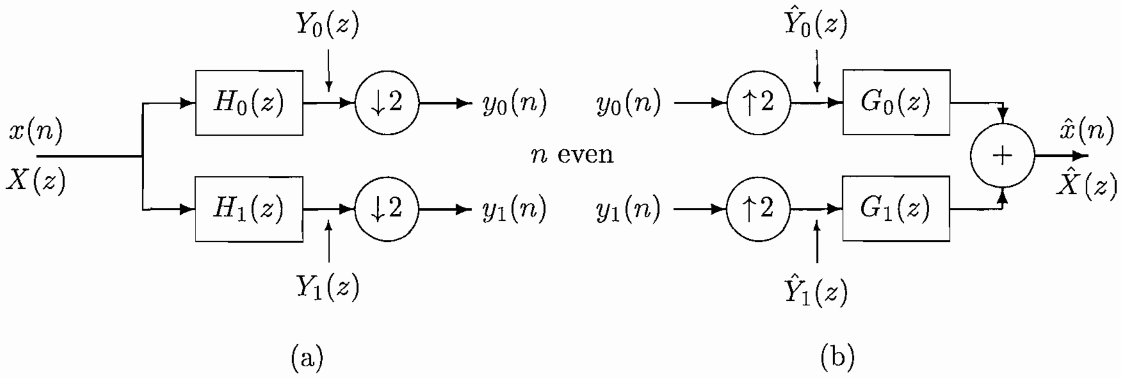


Fig 4.1: Two-band filter banks for analysis (a) and reconstruction (b).

The above figure from the lecture notes shows the required filter banks. To produce a 2-D wavelet transform, they must be used on the rows and columns of the image, separably, and also be used over more than one level of filtering – i.e. the Lo-Lo band from the first level of 2-D filtering must be used as the input to a second level of filtering, and so on, typically over several levels. This is shown below for just two levels (also from the lecture notes).

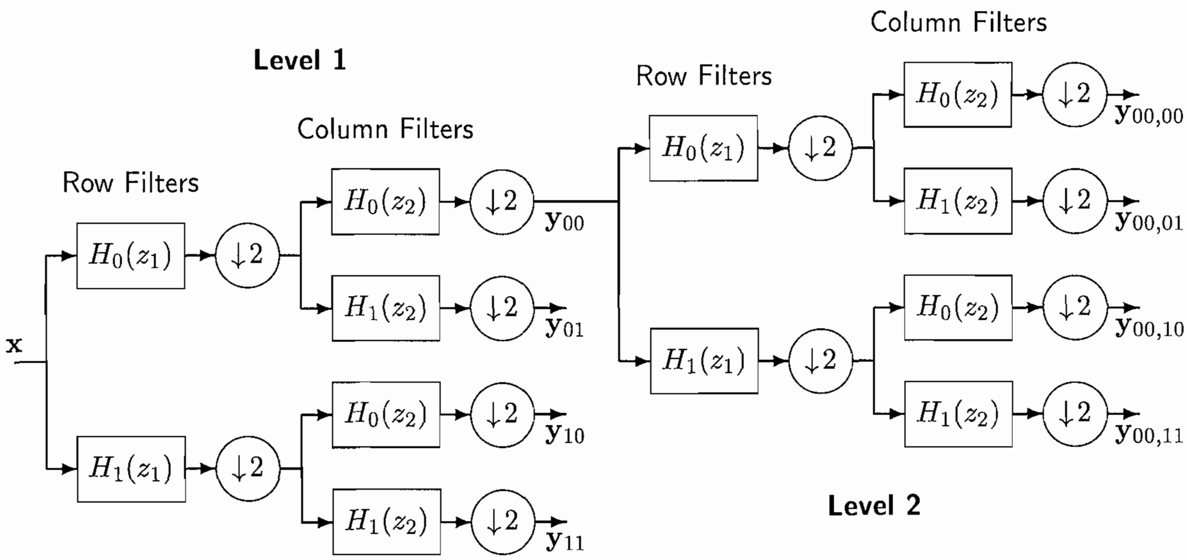


Fig 4.13: Two levels of a 2-D filter tree, formed from 1-D lowpass (H_0) and highpass (H_1) filters.

The reconstruction tree is just the reverse of this process with the G filters replacing the H filters. [25%]

(b) Perfect Reconstruction:

In fig 4.1, the combined downsampling and upsampling operation may be analysed as:

$$\hat{y}_0(n) = y_0(n) \text{ for } n \text{ even}, \qquad \hat{y}_0(n) = 0 \text{ for } n \text{ odd.}$$

Therefore $\hat{Y}_0(z)$ is a polynomial in z, comprising *only* the terms in even powers of z from $Y_0(z)$. This may be written as:

$$\hat{Y}_0(z) = \sum_{\text{even } n} y_0(n)\ z^{-n} = \sum_{\text{all } n} \tfrac{1}{2}[y_0(n)\ z^{-n} + y_0(n)\ (-z)^{-n}] = \tfrac{1}{2}[Y_0(z) + Y_0(-z)]$$

Similarly

$$\hat{Y}_1(z) = \tfrac{1}{2}[Y_1(z) + Y_1(-z)]$$

(These two results could be just quoted.)

Applying these to the system of fig 4.1 gives:

$$\begin{aligned}\hat{X}(z) &= \tfrac{1}{2}G_0(z)[Y_0(z) + Y_0(-z)] + \tfrac{1}{2}G_1(z)[Y_1(z) + Y_1(-z)] \\ &= \tfrac{1}{2}G_0(z)H_0(z)X(z) + \tfrac{1}{2}G_0(z)H_0(-z)X(-z) \\ &\quad + \tfrac{1}{2}G_1(z)H_1(z)X(z) + \tfrac{1}{2}G_1(z)H_1(-z)X(-z) \\ &= \tfrac{1}{2}X(z)[G_0(z)H_0(z) + G_1(z)H_1(z)] \\ &\quad + \tfrac{1}{2}X(-z)[G_0(z)H_0(-z) + G_1(z)H_1(-z)]\end{aligned}$$

If we require $\hat{X}(z) \equiv X(z)$ — *the Perfect Reconstruction (PR) condition* — then:

$$G_0(z)H_0(z) + G_1(z)H_1(z) = 2 \qquad \text{and} \qquad G_0(z)H_0(-z) + G_1(z)H_1(-z) = 0$$

If $G_1(z) = zH_0(-z)$, then using the second of the above 2 results:

$$G_0(z)H_0(-z) = -G_1(z)H_1(-z) = -zH_0(-z)H_1(-z)$$

So, assuming that $H_0(-z) \neq 0$:

$$G_0(z) = -zH_1(-z)$$

Changing $-z$ for z, this may also be written as $zH_1(z) = G_0(-z)$.

Now, returning to the first of the above PR results:

$$G_0(z)H_0(z) + G_1(z)H_1(z) = G_0(z)H_0(z) + zH_0(-z)\ z^{-1}G_0(-z) = P(z) + P(-z)$$

Hence

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

In this expression the odd-order terms of $P(-z)$ cancel out those of $P(z)$, so there are no constraints on the odd-order terms.

The even-order terms of $P(-z)$ equal those of $P(z)$ and add up on the LHS. Since there are no powers of z on the RHS of the above expression, all the even powers of z in $P(z)$ must be zero, apart from the term in z^0 which must be unity to give 2 on the RHS. [30%]

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

Since the transformation is symmetrical in positive and negative powers of z, any polynomial in non-negative powers of Z will result in a symmetric polynomial in z – i.e. the coefficient of z^{-k} will equal that of z^k. Hence $P(z)$ will be symmetric about z^0 and be of finite length, and therefore produce a zero-phase FIR filter.

For image processing this means that edges will be treated symmetrically and not be shifted to the left or right, which gives minimum distortion for a given amplitude response in the frequency domain.

We note that there are only odd integer powers of z in the transformation. Hence the odd powers of Z will only produce odd powers of z and even powers of Z will only produce even powers of z. Thus all the even powers of Z in $P_t(Z)$ must be zero, apart from the Z^0 term, if all the even powers of z are to be zero in $P(z)$. Furthermore, the Z^0 term in $P_t(Z)$ must be unity, because there are no other terms in even powers of Z which could potentially contribute to the z^0 term in $P(z)$. The odd powers of Z in $P_t(Z)$ are unconstrained as they cannot affect the terms in even powers of z. [15%]

(d) If $P_t(Z)$ is of the form

$$P_t(Z) = (1 + Z)^3 \; R(Z)$$

we require that $R(Z)\ (1 + 3Z + 3Z^2 + Z^3)$ has no terms in even powers of Z other than a constant term of unity. To eliminate the term in Z^2, R must be at least of the form $(a + bZ)$, but this will introduce a non-zero term in Z^4. To also eliminate this term, R must be of the form $(a + bZ + cZ^2)$. This will produce a non-zero term in Z^5 but this is permitted.

To find $\{a, b, c\}$, we find the coefs of Z^0, Z^2 and Z^4:

Coef of $Z^0 = a \,.\, 1 = a = 1$

Coef of $Z^2 = 3a + 3b + c = 0$

Coef of $Z^4 = b + 3c = 0$

Hence $b = -3c$ and $a = 1$, so $3 - 9c + c = 0$.

Therefore $c = \frac{3}{8}$, $b = -\frac{9}{8}$ and $a = 1$. [15%]

(e) To derive H_0 and G_0, we first obtain P(z) by substituting $Z = \frac{1}{2}(z + z^{-1})$ into $P_t(Z)$. Then we factorise $P(z)$ and allocate some of its factors to the $H_0(z)$ filter and the remaining factors to $G_0(z)$. This may be done in a variety of ways, but usually we try to ensure that a similar number of roots at or near $z = -1$ are allocated to each filter and that the other factors are grouped in conjugate pairs (if the roots are complex) so that both filters have real coefficients and have quite similar lowpass frequency responses.

The roots at or near $z = -1$ ensure smoothness of impulse response and low gain at frequencies in the left half of the z-plane. If one filter impulse response is smoother than the other, then that filter should be used for $G_0(z)$ to get the best quality image coder.

Often, the factorisation problem can be simplified by first factorising $P_t(Z)$ and then transforming from Z to z in each factor separately. [15%]