COMMUNICATING PICTURES: TUTORIAL SOLUTIONS

Please note: These solutions are provided as a guide and aid for readers of the book Communicating Pictures. While every effort has been made to ensure their correctness, course organizers are advised to verify them independently before use in examination material.

Chapter 1: Introduction

Q1.1 What is the primary purpose of standardisation in video compression? List two other advantages of standardisation.

Outline solution

- Standards are essential for interoperability, enabling material from different sources to be processed and transmitted over a wide range of networks or stored on a wide range of devices.
- This interoperability underpins an enormous market for video equipment, which can exploit the advantages of volume manufacturing, reducing equipment and product costs while also providing the widest possible range of services for users.

——oooo——

Q1.2 Using the example of a DVB-T2 terrestrial broadcast system transmitting HDTV video content to the home, explain why digital video compression is needed.

Outline solution

- The basic bit rate of an 8 bit HDTV 1080i30 video, without chrominance subsampling, is approximately 1.5Gbps whereas the typical bandwidth of a UK DVB-T2 link is 40Mbps, currently shared between 4 TV programmes. Chrominance subsampling using a 4:2:0 format will reduce the required bit rate by a factor of 2, but even then a compression ratio of 75:1 is needed.

——oooo——

Q1.3 Consider the case of 4K UHDTV, with the original video in 4:2:2 (a luma signal of 3840×2160 and 2 chroma signals of 1920×2160) format at 10 bits and a frame rate of 50fps. Calculate the compression ratio if this video is to be transmitted over a DVB-T2 link with an average bandwidth of 15Mb/s.

Outline solution

- The 4:2:2 colour subsampling method has, on average, the equivalent of two 10 bit samples for each pixel (see Chapter 4). Thus, in its uncompressed form the bit rate is calculated as follows:

$$R = 3840(\mathrm{H}) \times 2160(\mathrm{V}) \times 2(\mathrm{samples/pixel}) \times 10(\mathrm{bits}) \times 50(\mathrm{fps}) = 8.2944 \times 10^9\ \mathrm{b/s}$$

- Assuming this is transmitted over a DVB-T2 link at 15 Mbps then the compression ratio required would be 553:1. i.e.

$$\mathrm{CR} = \frac{8294400000}{15000000} \approx 553$$

———oooo———

Chapter 2: The human visual system

Q2.1 Assuming that the field of view within the fovea is 2 degrees, compute the number of pixels that fall horizontally within the foveated visual field. Assume a 1m wide screen with 1920 horizontal pixels viewed at a distance of $3H$.

Outline solution

- Assuming a 16:9 screen aspect ratio, the height H of the screen is 9/16m.
- The viewing distance of $3H = 1.6875$m.
- The amount of the screen foveated at 1.6875m is $x = 2 \times 1.6875 \times \tan(1^o) = 0.0589$m
- The number of horizontal pixels that fall within the foveated field of view is thus $1920 \times 0.0589 = 113$

——oooo——

Q2.3 Calculate the normalised contrast sensitivity of the human visual system for a luminance-only stimulus at a spatial frequency of 10 cycles per degree.

Outline solution

- An approximation for the normalised CSF is given in equation 2.6:

$$C(f) = 2.6\,(0.0192 + 0.114f)\,e^{-(0.114f)^{1.1}}$$

- Evaluating this equation at a frequency of 10 cycles per degree, the normalised CSF is 0.9495

——oooo——

Q2.2 The following table lists a number of important features of the human visual system (HVS). Complete the table by describing the influence each feature has on the design of a digital video compression system.

Outline solution

HVS characteristic	**Implication for compression**
HVS more sensitive to high contrast image regions than low contrast regions.	Edges should be preserved. Artificial edge artefacts introduced by quantisation are very noticeable and should be avoided. Spatial (contrast) masking of quantisation noise can be exploited in textured regions.
HVS is more sensitive to luminance than chrominance information.	Visual signal representations can be based on colour spaces that split luminance and chrominance. Chrominance channels can be allocated lower bandwidth than the luminance channel.
HVS is more sensitive to lower spatial frequencies than higher spatial frequencies.	Frequencies beyond the upper range need not be processed. Higher frequencies can be coded more coarsely with implications for quantisation. The lower frequency luminance roll-off is not normally exploited.
In order to achieve a smooth appearance of motion, the HVS must be presented with image frames above a certain minimum rate (and this rate depends on ambient light levels).	Frame rates for acquisition and display should normally be at least 50 Hz. Due to flicker effects, higher frame refresh rates are required for larger screens and closer viewing.
HVS responses vary from individual to individual.	Subjective quality assessment experiments should be based on results from a large number of subjects (typically $>$ 20).

——oooo——

Chapter 3: Discrete time analysis for images and video

Q3.1 Plot the sinusoidal signal $x(t) = \cos(20\pi t)$. Compute the frequency spectrum of this sinusoid and plot its magnitude spectrum.

Outline solution

- $x(t) = \cos(20\pi t)$ represents a time domain sinusoidal signal at a frequency of $f_0 =$ 10Hz. Thus the period $T_0 = 0.1s$ and $\omega_0 = 20\pi$.
- We can compute the spectral components using the Fourier series representation, thus:

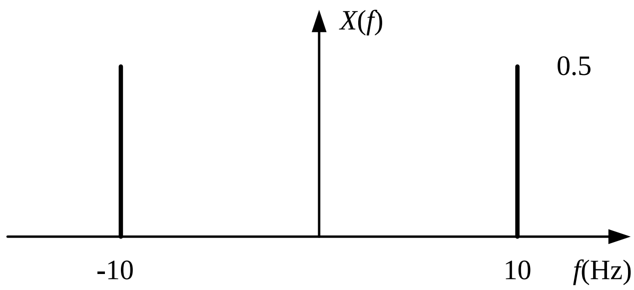
$$C_k = \frac{1}{T_0}\int_T x(t)\, e^{-j\omega_0 kt} dt$$

$$= \frac{1}{T_0}\int_T \cos(\omega_0 t)\, e^{-j\omega_0 kt} dt$$

$$= \frac{1}{2T_0}\int_T \left(e^{j\omega_0 kt} + e^{-j\omega_0 kt}\right) e^{-j\omega_0 kt} dt$$

$$= \frac{1}{2T_0}\int_T e^{j\omega_0 t(1-k)} dt + \frac{1}{2T_0}\int_T e^{-j\omega_0 t(1+k)} dt$$

- The first integral term is equal to zero except when k=1. Thus $C_1 = 0.5$. Similarly, the second integral term is zero except when k=-1, giving $C_{-1} = 0.5$.
- Hence the spectrum for this function is:



———oooo———

Q3.2 Assume that the sinusoidal signal in Q3.1 is sampled with $T = 0.1$s. Sketch the magnitude spectrum of the sampled signal and comment on any aliasing issues.

Outline solution

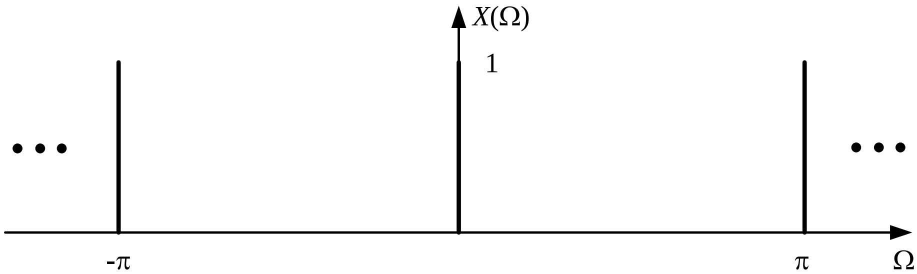
- Using the DFT we can compute the spectral characteristics of the sampled signal. Because the signal is sampled at a frequency of 10Hz (the same as its fundamental frequency) there is only one sample per cycle. Aliasing will thus occur where the primary alias (i.e. the reconstructed signal after low pass filtering) will be at DC.
- Using the DFT:

$$X(k) = \sum_{n=0}^{N-1} x[n]\, e^{-j\frac{2\pi}{N}kn}$$

- Since N=1:

$$X(0) = \sum_{n=0}^{0} x[0]\, e^{-j\frac{2\pi}{1}0} = 1.e^{-j\frac{2\pi}{1}0} = 1$$

- Similarly for other values of k. Hence the spectrum is:



———oooo———

Q3.3 If an HDTV (full HD) screen with aspect ratio 16:9 has a width of 1.5m and is viewed at at distance of $4H$, what is the angle subtended by each pixel at the retina?

Outline solution

- For a 16:9 screen of width 1.5m, the screen height H is $1.5 \times 9 \div 16$
- The viewing distance is therefore $4H$=3.275m
- The width of a pixel is $1.5 \div 1920 = 0.00078125$m
- The angle subtended by a pixel under these conditions is thus

$$\theta = 2\tan^{-1}\left(\frac{0.00078125}{3.375 \times 2}\right) \approx 0.0133^{\circ}$$

———oooo———

Q3.4 The impulse response for the Le Gall high pass analysis filter is $h_1[n] = \{\ 0.25 \quad -0.5 \quad 0.25\ \}$. Compute the output from this filter for the input sequence: $x[n] = \{\ 1 \quad 2 \quad 3 \quad 0 \quad 0 \quad \cdots\ \}$.

Outline solution

n	0	1	2	3	4	5
$y_1[n]$	0.25	-0.5	0.25	0	0	0
$y_2[n]$	0	0.5	-1	0.5	0	0
$y_3[n]$	0	0	0.75	-1.5	0.75	0
$y[n]$	0.25	0	0	-1	0.75	0

———oooo———

Q3.5 Compute the z-plane pole zero plots and the frequency responses for the following filter pair: $H_0(z) = 1 + z^{-1}$ and $H_1(z) = 1 - z^{-1}$.

Outline solution

$$H_0(z) = 1 + z^{-1} = \frac{z+1}{z}$$

- This implies that the function has a zero at z=-1 and a pole at z=0.
- The frequency response is given by:

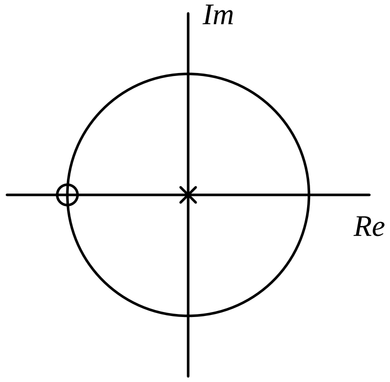
$$H(\Omega) = H(z)\left.\right|_{z=e^{j\Omega}} = 1 + e^{-j\Omega}$$

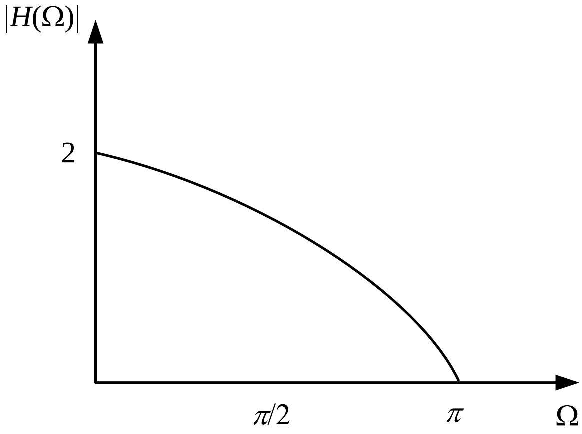
- Considering the magnitude response only:

$$\Omega = 0 \implies |H(\Omega)| = 2$$

$$\Omega = \pi \implies |H(\Omega)| = 0$$

$$\Omega = \pi/2 \implies |H(\Omega)| = \sqrt{2}$$





- A similar calculation can be performed for $H_1(z)$

———oooo———

Q3.6 Perform median filtering on the following input sequence using a 5 tap median filter and comment on the result.

$$x[n] = \{1, 3, 5, 13, 9, 11, 6, 15, 17, 19, 29\}$$

Outline solution

$$x[n] = \{1, 3, 5, 13, 9, 11, 6, 15, 17, 19, 29\}$$

- The output of the median filter is given by:

$$y[n] = \{0, 0, 1, 3, 5, 9, 9, 11, 11, 15, 17, 17, 17, 0, 0\}$$

- The input signal is linear in n with impulsive noise added. The median filter smooths this out and helps to preserve 'good' values without introducing the distortion that would occur with a linear filter. There is a delay of 2 samples through the filter and the first two and last two samples represent transient effects.

———oooo———

Q3.7 Compute the basis functions for the 2-point DFT.

Outline solution

- The basis functions of the DFT are given by $W = e^{-j2\pi/N}$. For the case of N=2 we have:

$$W = \begin{bmatrix} e^{-j2\pi 0/2} & e^{-j2\pi 0/2} \\ e^{-j2\pi 0/2} & e^{-j2\pi 1/2} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

- i.e. the basis functions are effectively sum and difference operators similar to the 2-pt DCT and the 2-pt KLT.

———oooo———

Q3.8 Consider the two 1-D digital filters:

$$\mathbf{h}_1 = [\ 1 \quad 2 \quad 2 \quad 1]^{\mathrm{T}} \ ;$$
$$\mathbf{h}_2 = [\ 1 \quad -3 \quad -3 \quad 1]^{\mathrm{T}}$$

Compute the equivalent 2-D digital filter where $\mathbf{h}_1$ performs horizontal filtering and $\mathbf{h}_2$ performs vertical filtering.

Outline solution

- The equivalent 2-D digital filter is given by:

$$\mathbf{h} = \mathbf{h}_1 \mathbf{h}_2^T = \begin{bmatrix} 1 \\ 2 \\ 2 \\ 1 \end{bmatrix} \begin{bmatrix} 1 & -3 & -3 & 1 \end{bmatrix}$$

$$\mathbf{h} = \begin{bmatrix} 1 & -3 & -3 & 1 \\ 2 & -6 & -6 & 2 \\ 2 & -6 & -6 & 2 \\ 1 & -3 & -3 & 1 \end{bmatrix}$$

———oooo———

Q3.9 Compute biased and unbiased correlation estimates for the following sequence:

$$x[n] = \{1, 2, 5, -1, 3, 6, -4, -1\}$$

Outline solution

- The biased estimate is given by:

$$R_{xx}[m] = \frac{1}{N} \sum_{n=0}^{N-|m|-1} x[n]\, x[n+m]$$

- giving a solution for the above sequence, of:

$$\mathbf{r} = [11.625, 0.25, -0.75, 4.5, -0.5, -0.875, -0.75, -0.125]$$

- The unbiased estimate is given by:

$$R_{xx}[m] = \frac{1}{N-|m|} \sum_{n=0}^{N-|m|-1} x[n]\, x[n+m]$$

- giving a solution for the above sequence, of:

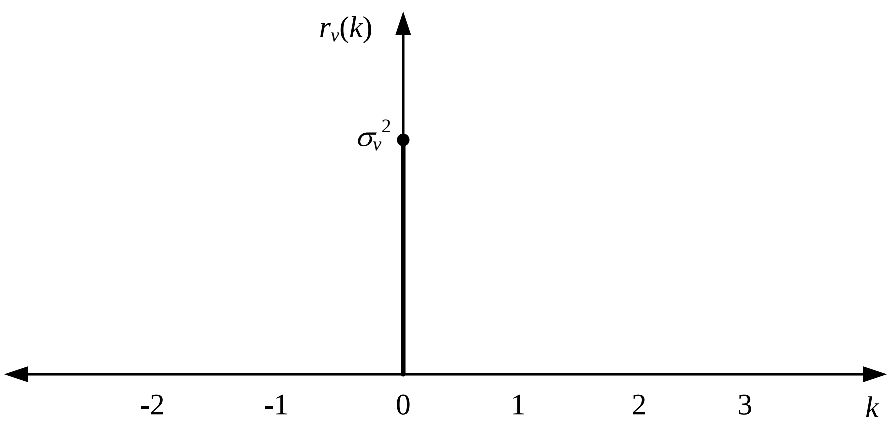
$$\mathbf{r} = [11.625, 0.2857, -1, 7.2, -1, -2.333, -3, -1]$$

——oooo——

Q3.10 Plot the autocorrelation function, $r_v(k)$ for a white noise sequence $v[n]$ with variance σ_v^2. Form the autocorrelation matrix for this sequence for lags up to ± 3. What is the inverse of this autocorrelation matrix?

Outline solution i)

$$r_v(k) = E\{v[n]\, v[n-k]\} = \sigma_v^2 \delta(k)$$



ii)

$$\mathbf{r}_{\mathrm{v}}(k) = E\left\{\mathbf{v}\mathbf{v}^{\mathrm{T}}\right\} = \begin{bmatrix} \sigma_v^2 & 0 & 0 & 0 \\ 0 & \sigma_v^2 & 0 & 0 \\ 0 & 0 & \sigma_v^2 & 0 \\ 0 & 0 & 0 & \sigma_v^2 \end{bmatrix} = \sigma_v^2 \mathbf{I}$$

iii) The inverse of the above matrix (since, by definition, the inverse of $\mathbf{I} = \mathbf{I}$) is given by:

$$\mathbf{r}_{\mathbf{v}}^{-\mathbf{1}}(k) = \left(\sigma_v^2 \mathbf{I}\right)^{-1} = \frac{1}{\sigma_v^2}\mathbf{I}$$

——oooo——

Q3.11 A feedback-based linear predictor with quantisation uses a predictor $P(z) = (z^{-1} + z^{-2})/2$. Assume an input sequence as given below:

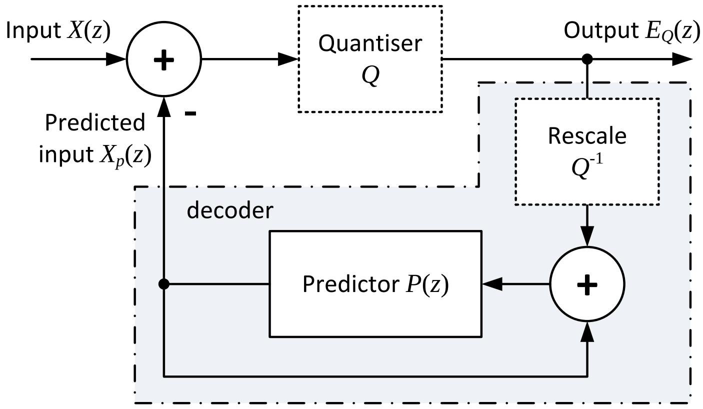
$$x[n] = \{1, 3, 4, 3, 5, 6 \cdots\}$$

and that quantisation is performed as follows, with rounding of 0.5 values toward zero:

$$e_Q[n] = \mathrm{rnd}\left(\frac{e[n]}{2}\right); \quad e_R[n] = 2e_Q[n]$$

Compute the predictor output sequence $e_Q[n]$ and the reconstructed output signal $y[n]$ from the decoder. Comment on your results in terms of numerical precision.

Outline solution



$x[n]$	1	3	4	3	5	6	$\cdots$
$x_p[n] = (y[n-1] + y[n-2])/2$	0	0	1	2.5	2.75	3.625	$\cdots$
$e[n] = x[n] - x_p[n]$	1	3	3	0.5	2.25	2.375	$\cdots$
$e_Q[n] = \text{rnd}(e[n]/2)$	0	1	1	0	1	1	$\cdots$
$e_R[n] = 2e_Q[n]$	0	2	2	0	2	2	$\cdots$
$y[n] = x_p[n] + e_R[n]$	0	2	3	2.5	4.75	5.625	$\cdots$

- Note: Internal numerical precision (consistency between encoder and decoder) is key to maintaining prediction accuracy without drift. In this case, for example, $y[n]$ could be rounded to integer values prior to prediction.

——oooo——

Q3.12 Compute the entropies of the sequences $y[n]$ and $e_Q[n]$ from Q3.11. Comment on your result.

Outline solution

- Assuming that (very crudely) the probabilities of the sample values for both the input sequence and the quantised error sequence are given by their relative frequencies of occurrence, then these are:
- For $x[n]$:

$$P(1) = 1/6;\ P(3) = 2/6;\ P(4) = 1/6;\ P(5) = 1/6;\ P(6) = 1/6$$

- Hence:

$$H_x = -\sum_{i=1}^{N} P(s_i)\log_2(P(s_i)) = 4 \times (1/6)\log_2(1/6) + (1/3)\log_2(1/3) = 2.2516\text{bits}$$

- and for $e_Q[n]$:

$$P(0) = 4/6;\ P(1) = 2/6$$

- Hence:

$$H_e = -\sum_{i=1}^{N} P(s_i) \log_2 (P(s_i)) = (2/3) \log_2 (2/3) + (1/3) \log_2 (1/3) = 0.9183\text{bits}$$

- So, on the basis of these simple assumptions, the average wordlength required to code the quantised error signal is significantly smaller that that for the original signal.

——oooo——

Chapter 4: Digital picture formats and representations

Q4.1 In video coding schemes it is usual to code the colour components in the form Y, C_b, C_r rather than R, G, B. Explain why this approach is justified, its benefits in terms of compression and how it is exploited in image sampling. Explain how pictures can be efficiently stored or transmitted using a 4:2:0 format.

Outline solution

- **Justification**: the human visual system has a relatively poor response to colour information. The effective resolution of our colour vision is about half that of luminance information.
- **How exploited**: separation of luma and chroma components thus means that we can sub-sample the chroma information by a factor of two without affecting the luma resolution or the perceived image quality.
- **Benefits**: this provides an immediate compression benefit as we are reducing the bit rate by 50% before we start compression proper. As well as having compatibility with monochrome displays, the availability of an independent luma signal is useful during compression. For example, we normally perform motion estimation only on the luma component and use the resulting vectors to compensate both luma and chroma components. We also often base objective quality assessments only on the luma component.
- **4:2:0**: This system is the most common format for broadcast delivery, internet streaming and consumer devices. It sub-samples the chroma signals by a factor of two in both horizontal and vertical directions, maximising the exploitation of perceptual colour redundancy.

——oooo——

Q4.2 Compute YUV and Y C_b C_r vectors for the following RGB vectors:

$$\begin{array}{l} \text{a) } [R,G,B] = \begin{bmatrix} 128 & 128 & 128 \end{bmatrix} \\ \text{b) } [R,G,B] = \begin{bmatrix} 255 & 255 & 255 \end{bmatrix} \\ \text{c) } [R,G,B] = \begin{bmatrix} 100 & 0 & 0 \end{bmatrix} \end{array}$$

Outline solution

- Assuming the RGB values are gamma corrected, we can use the following matrix-vector calculation:

$$\begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix} = \begin{bmatrix} 0.257 & 0.504 & 0.098 \\ -0.148 & -0.291 & 0.439 \\ 0.439 & -0.368 & -0.071 \end{bmatrix} \begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} + \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix}$$

- to give the following results:

a) $[Y, C_b, C_r] = \begin{bmatrix} 126 & 128 & 128 \end{bmatrix}$
b) $[Y, C_b, C_r] = \begin{bmatrix} 235 & 128 & 128 \end{bmatrix}$
c) $[Y, C_b, C_r] = \begin{bmatrix} 42 & 113 & 172 \end{bmatrix}$

——oooo——

Q4.3 If a colour movie of 100 minutes duration is represented using ITU-R.601 (720×576, 25fps@8bits, 4:2:0 format):

a) What hard disk capacity would be required to store the whole movie?

b) If the movie is encoded at a compression ratio CR=40:1 and transmitted over a satellite link with 50% channel coding overhead, what is the total bitrate required for the video signal?

Outline solution

(a) The bit rate is given by:

$$720 \times 576 \times 25 \times 8 \times \frac{3}{2} = 124,416,000\text{b/s} = 124\text{Mb/s}$$

- For a 100 minute movie, the total storage space required is:

$$100 \times 60 \times 124,416,000 = 746,496,000,000\text{bits} \approx 75\text{GB}$$

(b) With a compression ratio of 40:1, the bit rate would be approximately 1.866Mbps. With 50% coding overhead, this represents a total bit rate of approximately 2.799Mbps.

——oooo——

Q4.4 Given a video sequence with a spatial resolution of 1920×1080 at 50fps using 10 bit colour sampling, compute the (uncompressed) bit rates of 4:4:4, 4:2:0 and 4:2:2 systems.

Outline solution

- **4:4:4**:$1920 \times 1080 \times 50 \times 10 \times 3 \approx 3.11\text{Gbps}$
- **4:2:2**: $1920 \times 1080 \times 50 \times 10 \times 2 \approx 2.07\text{Gbps}$
- **4:2:0**: $1920 \times 1080 \times 50 \times 10 \times 1.5 \approx 1.56\text{Gbps}$

——oooo——

Q4.5 If, for a given 1920×1080 I-frame in 4:2:0 format, the mean entropy of each coded 16×16 luminance block is 1.3 bits/sample and that for each corresponding chrominance block is 0.6 bits/sample, then estimate the total number of bits required to code this frame.

Outline solution

- The number of macroblocks in each frame is 8100 - each comprising 1 luma block and 2 chroma blocks.
- Each luma block has 256 samples whereas each chroma block has 64 samples.
- An estimate for the total number of bits required to code the frame is thus:

$$8100 \times (256 \times 1.3 + 64 \times 2 \times 0.6) = 3,317,760\text{bits}$$

———oooo———

Q4.6 Calculate the MAD between the following original image block $\mathbf{X}$ and its encoded and decoded version, $\tilde{\mathbf{X}}$:

$$\mathbf{X} = \begin{bmatrix} 2 & 4 & 4 & 6 \\ 3 & 6 & 6 & 6 \\ 5 & 7 & 7 & 8 \\ 3 & 7 & 7 & 8 \end{bmatrix}; \ \tilde{\mathbf{X}} = \begin{bmatrix} 2 & 5 & 5 & 5 \\ 4 & 4 & 6 & 6 \\ 5 & 5 & 7 & 8 \\ 4 & 5 & 7 & 7 \end{bmatrix}$$

Outline solution

- The absolute difference matrix is given by:

$$\left|\mathbf{X} - \tilde{\mathbf{X}}\right| = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 2 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 1 & 2 & 0 & 1 \end{bmatrix}$$

- Hence:

$$\text{MAD} = \frac{12}{16} = 0.75$$

———oooo———

Q4.7 Assuming a 5 bit digital image block, $\mathbf{X}$, the reconstruction after image compression is given by $\mathbf{Y}$. Calculate the PSNR for the reconstructed signal and provide an interpretation of the results in terms of error visibility.

$$\mathbf{X} = \begin{bmatrix} 3 & 8 & 1 & 8 \\ 7 & 0 & 5 & 0 \\ 2 & 6 & 0 & 5 \\ 4 & 1 & 10 & 2 \end{bmatrix}; \ \mathbf{Y} = \begin{bmatrix} 4 & 10 & 1 & 9 \\ 7 & 0 & 6 & 1 \\ 3 & 6 & 0 & 4 \\ 5 & 1 & 12 & 3 \end{bmatrix}$$

Outline solution

- We can see that $x_{\max}$=31 and A=16. The error matrix is:

$$|\mathbf{X}-\mathbf{Y}| = \begin{bmatrix} 1 & 2 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 2 & 1 \end{bmatrix}$$

- The MSE is thus given by:

$$\mathrm{MSE} = \frac{16}{16} = 1$$

- The PSNR for this block is given by:

$$\mathrm{PSNR} = 10 \cdot \log_{10}\left(\frac{16 \times 31^2}{16}\right) = 29.83\mathrm{dB}$$

———oooo———

Q4.8 Assuming an 8 bit digital image, **X**, the reconstructions due to 2 alternative coding schemes are given by $\mathbf{Y}_1$ and $\mathbf{Y}_2$ below. Calculate the Peak Signal to Noise Ratio (PSNR) for each of the reconstructions and give an interpretation of the results in terms of error visibility.

$$\mathbf{X} = \begin{bmatrix} 20 & 17 & 18 \\ 15 & 14 & 15 \\ 19 & 13 & 14 \end{bmatrix}; \ \mathbf{Y_1} = \begin{bmatrix} 19 & 18 & 17 \\ 16 & 15 & 14 \\ 18 & 14 & 13 \end{bmatrix}; \ \mathbf{Y_2} = \begin{bmatrix} 20 & 17 & 18 \\ 15 & 23 & 15 \\ 19 & 13 & 14 \end{bmatrix}$$

Outline solution

$$|\mathbf{X}-\mathbf{Y_1}| = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}; \ |\mathbf{X}-\mathbf{Y_2}| = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 9 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

- In the first case the MSE is given by:

$$\mathrm{MSE} = \frac{9}{9} = 1$$

- and the PSNR for this reconstruction is given by:

$$\mathrm{PSNR} = 10 \cdot \log_{10}\left(\frac{9 \times 255^2}{9}\right) = 48.13\mathrm{dB}$$

- In the second case the MSE is given by:

$$\text{MSE} = \frac{81}{9} = 9$$

- and the PSNR for this reconstruction is given by:

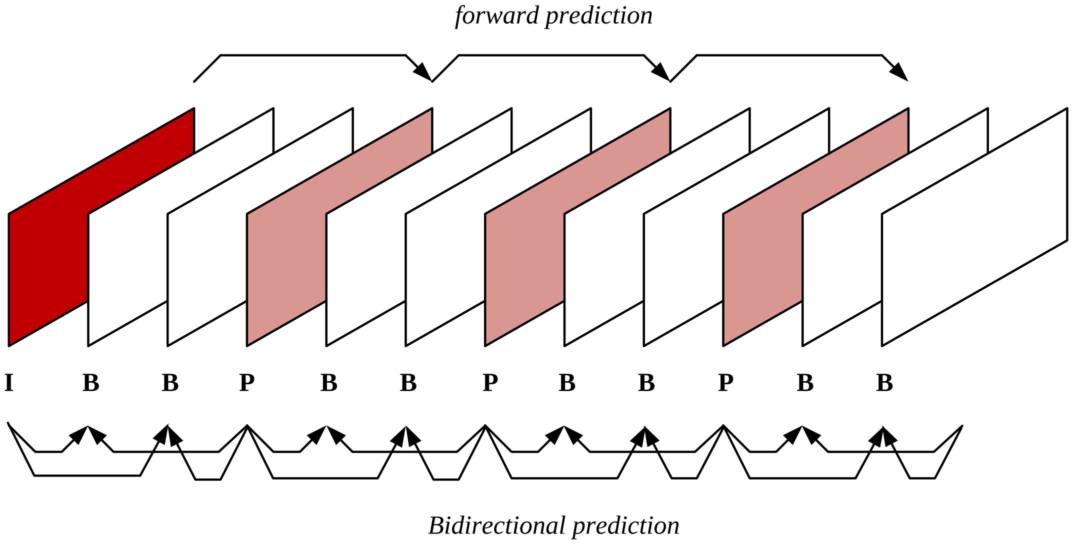
$$\text{PSNR} = 10 \cdot \log_{10}\left(\frac{9 \times 255^2}{81}\right) = 38.59\text{dB}$$

- Comparing these two cases, the Mean absolute differences are identical, but the mean squared error is significantly higher for the second case which has a localised rather than a distributed error. The perceptual effect of these errors will depend on context, but, in general, it is likely that localised errors will be more noticeable that distributed ones.

———oooo———

Q4.9 Describe a typical GOP structure used in MPEG-2 television broadcasting, explaining the different types of frame coding employed, the predictive relationships between all frames in the GOP and the transmission order of the frames. If a transmission error affects the 2nd P-frame in the GOP, how many pictures are likely to be affected due to error propagation?

Outline solution



- A typical GOP structure would comprise 12 or 15 frames in an IBBPBBPBBPBB format.

- See main text for a full description of frame types.
- Assuming a 12 frame GOP, if a transmission error affects the 2nd P-frame in the GOP, then temporal propagation of errors would persist until the next I frame. i.e. a total of 6 P and B frames would be affected.

——oooo——

Q4.10 An HDTV satellite operator allocates 10Mb/s to each programme in the DVB-S multiplex. State what colour sub-sampling format will be used for this and calculate the required compression ratio for a 1080i25 system.

Outline solution

- The colour subsampling format used for such a satellite transmission is invariably 4:2:0.
- Assuming a typical 8 bit format, the raw bit rate for a 1080i25 4:2:0 system is: $1920 \times 1080 \times 25 \times 8 \times 1.5 = 622,080,000 \approx 622$Mbps
- The required compression ratio for a 1080i25 4:2:0 transmission at 10Mbps would therefore be approximately: 62:1.

——oooo——

Q4.11 Compute the gamma corrected version of the following image block for the case where $\gamma = 0.45$. Assume an 8 bit wordlength.

$$\mathbf{X} = \begin{bmatrix} 20 & 17 & 18 \\ 15 & 14 & 15 \\ 19 & 13 & 14 \end{bmatrix}$$

Outline solution

- Using the equation:

$$V = c_1\Phi^\gamma + c_2$$

- Assuming $c_1 = 1$ and $c_2 = 0$, then we have:

$$\mathbf{X}' = \begin{bmatrix} 81 & 76 & 78 \\ 71 & 69 & 71 \\ 79 & 67 & 69 \end{bmatrix}$$

- Note that values of gamma less than one will 'stretch' lower signal values such as those in this example.

——oooo——

Q4.12 Bayer filtering (demosaicing) is commonly used to interpolate colour planes for single sensor cameras.

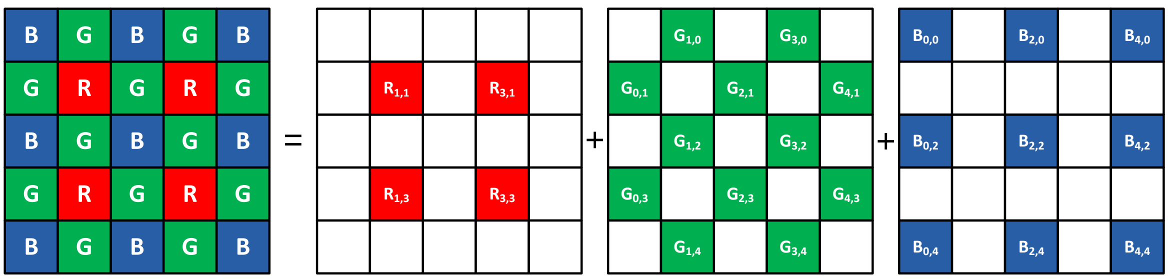
a) Draw a diagram of a 5 by 5 Colour Filter Array.

b) Derive the demosaicing kernels for this array for the R, G and B colour planes. Assume that bilinear interpolation is used for demosaicing.

c) Given the following acquired pixel values for the array in (a) in [R G B] format, calculate missing values, for location [x,y]=[1,1] using the interpolation kernel in (b).

[x x 10]	[x 10 x]	[x x 2]	[x 3 x]	[x x 1]
[x 10 x]	[20 x x]	[x 5 x]	[4 x x]	[x 0 x]
[x x 20]	[x 10 x]	[x x 4]	[x 2 x]	[x x 0]
[x 10 x]	[20 x x]	[x 5 x]	[4 x x]	[x 1 x]
[x x 20]	[x 10 x]	[x x 3]	[x 2 x]	[x x 1]

Outline solution a)



b)

$$K_B = K_R = \frac{1}{4}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix};\ K_G = \frac{1}{4}\begin{bmatrix} 0 & 1 & 0 \\ 1 & 4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

c)

$$R_{1,1} = \frac{1}{4}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} * \begin{bmatrix} 0 & 0 & 0 \\ 0 & 20 & 0 \\ 0 & 0 & 0 \end{bmatrix} = 20$$

$$G_{1,1} = \frac{1}{4}\begin{bmatrix} 0 & 1 & 0 \\ 1 & 4 & 1 \\ 0 & 1 & 0 \end{bmatrix} * \begin{bmatrix} 0 & 10 & 0 \\ 10 & 0 & 5 \\ 0 & 10 & 0 \end{bmatrix} = 9$$

$$B_{1,1} = \frac{1}{4}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} * \begin{bmatrix} 10 & 0 & 2 \\ 0 & 0 & 0 \\ 20 & 0 & 4 \end{bmatrix} = 9$$

——oooo——

Chapter 5: Transforms for image and video compression

Q5.1 Derive the 1-D two-point KLT for a stationary real-valued process, x, that has the following autocorrelation matrix:

$$\mathbf{R}_x = \begin{bmatrix} r_{xx}(0) & r_{xx}(1) \\ r_{xx}(1) & r_{xx}(0) \end{bmatrix}$$

Outline solution

- We obtain the eigenvalues by solving $|\lambda \mathbf{I} - \mathbf{R}_x| = 0$. This yields:

$$\begin{aligned} \lambda_1 &= r_{xx}(0) + r_{xx}(1) \\ \lambda_2 &= r_{xx}(0) - r_{xx}(1) \end{aligned}$$

- The eigenvectors are thus given by (where a and b are arbitrary constants):

$$\begin{aligned} V_1 &= \begin{bmatrix} a \\ a \end{bmatrix} \\ V_2 &= \begin{bmatrix} b \\ -b \end{bmatrix} \end{aligned}$$

- If we constrain these vectors to form an orthonormal pair, then:

$$a = b = \frac{1}{\sqrt{2}}$$

- In the case of the 2 point KLT, the basis functions are not dependent on the autocorrelation values (this is not the case for higher order transforms) and the KLT transform matrix is given by:

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

——oooo——

Q5.2 Prove that the following vectors are an orthonormal pair:

$$\mathbf{a}_0 = \begin{bmatrix} \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{bmatrix}^{\mathrm{T}}; \quad \mathbf{a}_1 = \begin{bmatrix} \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} \end{bmatrix}^{\mathrm{T}}$$

Outline solution

- The conditions for orthonormality are:

$$\mathbf{a}_i^{\mathrm{H}}\mathbf{a}_j = \begin{cases} 1; \ i = j \\ 0; \ i \neq j \end{cases}$$

- These conditions are satisfied in this case as we have:

$$\mathbf{a}_0^{\mathrm{T}}\mathbf{a}_0 = \begin{bmatrix} \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{bmatrix} \begin{bmatrix} \frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} \end{bmatrix} = 1$$

$$\mathbf{a}_1^{\mathrm{T}}\mathbf{a}_1 = \begin{bmatrix} \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} \end{bmatrix} \begin{bmatrix} \frac{\sqrt{2}}{2} \\ -\frac{\sqrt{2}}{2} \end{bmatrix} = 1$$

$$\mathbf{a}_0^{\mathrm{T}}\mathbf{a}_1 = \begin{bmatrix} \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{bmatrix} \begin{bmatrix} \frac{\sqrt{2}}{2} \\ -\frac{\sqrt{2}}{2} \end{bmatrix} = 0$$

——oooo——

Q5.3 Prove that the four point DWHT is a unitary transform.

Outline solution

- The four point DWHT is given by:

$$\mathbf{H}_4 = \frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix}$$

- The norm of each basis function is unity. For example, for the first row we have:

$$\mathbf{h}_0^{\mathrm{T}}\mathbf{h}_0 = \frac{1}{4}\begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = 1$$

- The inner product of different basis functions is always 0. For example, for the first 2 rows we have:

$$\mathbf{h}_0^{\mathrm{T}}\mathbf{h}_1 = \frac{1}{4}\begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}\begin{bmatrix} 1 \\ 1 \\ -1 \\ -1 \end{bmatrix} = 0$$

———oooo———

Q5.4 Compute the basis functions for the eight point 1-D DWHT. Compute the first four basis functions for the four point 2-D DWHT.

Outline solution

- The 8-point 1-D DWHT can be computed as follows:

$$\mathbf{H}_8 = \frac{1}{\sqrt{8}}\begin{bmatrix} \mathbf{H}_4 & \mathbf{H}_4 \\ \mathbf{H}_4 & -\mathbf{H}_4 \end{bmatrix}$$

- Where:

$$\mathbf{H}_4 = \frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix}$$

- The 8-point DWHT is thus given by (reordering the basis functions in sequency order):

$$\mathbf{H}_8 = \frac{1}{2\sqrt{2}}\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \end{bmatrix}$$

- For the case of the four point 2-D DWHT, we can compute the basis functions as the outer product of the 1-D basis functions. Thus we have:

$$\mathbf{h}_0\mathbf{h}_0^{\mathrm{T}} = \frac{1}{4}\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}\begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix} = \frac{1}{4}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

- Similarly:

$$\mathbf{h}_0\mathbf{h}_1^{\mathrm{T}} = \frac{1}{4}\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}\begin{bmatrix} 1 & 1 & -1 & -1 \end{bmatrix} = \frac{1}{4}\begin{bmatrix} 1 & 1 & -1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & 1 & -1 & -1 \end{bmatrix}$$

$$\mathbf{h}_0\mathbf{h}_2^{\mathrm{T}} = \frac{1}{4}\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}\begin{bmatrix} 1 & -1 & -1 & 1 \end{bmatrix} = \frac{1}{4}\begin{bmatrix} 1 & -1 & -1 & 1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

$$\mathbf{h}_0\mathbf{h}_3^{\mathrm{T}} = \frac{1}{4}\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}\begin{bmatrix} 1 & -1 & 1 & -1 \end{bmatrix} = \frac{1}{4}\begin{bmatrix} 1 & -1 & 1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & 1 & -1 \end{bmatrix}$$

- Other 2-D basis functions can be computed in a similar manner.

——oooo——

Q5.5 Use the DWHT to transform the following 2-D image block, **S**. Assuming that all data and coefficients are represented as 8 bit numbers and that compression is achieved in the transform domain by selecting only one most dominant coefficient for transmission, compute the decoded data matrix and its PSNR.

$$\mathbf{S} = \begin{bmatrix} 10 & 10 & 10 & 10 \\ 10 & 10 & 10 & 10 \\ 10 & 10 & 9 & 9 \\ 10 & 10 & 9 & 9 \end{bmatrix}$$

Outline solution

- Computing the forward 2-D transform:

$$\mathbf{C} = \mathbf{H}\mathbf{X}\mathbf{H}^{\mathrm{T}}$$

[1 mark]

$$\mathbf{C} = \mathbf{H}\mathbf{X}\mathbf{H}^{\mathrm{T}} = \tfrac{1}{4}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix}\begin{bmatrix} 10 & 10 & 10 & 10 \\ 10 & 10 & 10 & 10 \\ 10 & 10 & 9 & 9 \\ 10 & 10 & 9 & 9 \end{bmatrix}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} =$$
$$\begin{bmatrix} 39 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

[3 marks]

———oooo———

- Selecting the 4 most dominant coefficients and performing an inverse transform, followed by rounding:

$$\tilde{\mathbf{S}} = \mathbf{H}^{\mathrm{T}}\mathbf{C}\mathbf{H} = \mathrm{rnd}\left(\frac{1}{4}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix}\begin{bmatrix} 39 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix}\right)$$

$$= \begin{bmatrix} 10 & 10 & 10 & 10 \\ 10 & 10 & 10 & 10 \\ 10 & 10 & 10 & 10 \\ 10 & 10 & 10 & 10 \end{bmatrix}$$

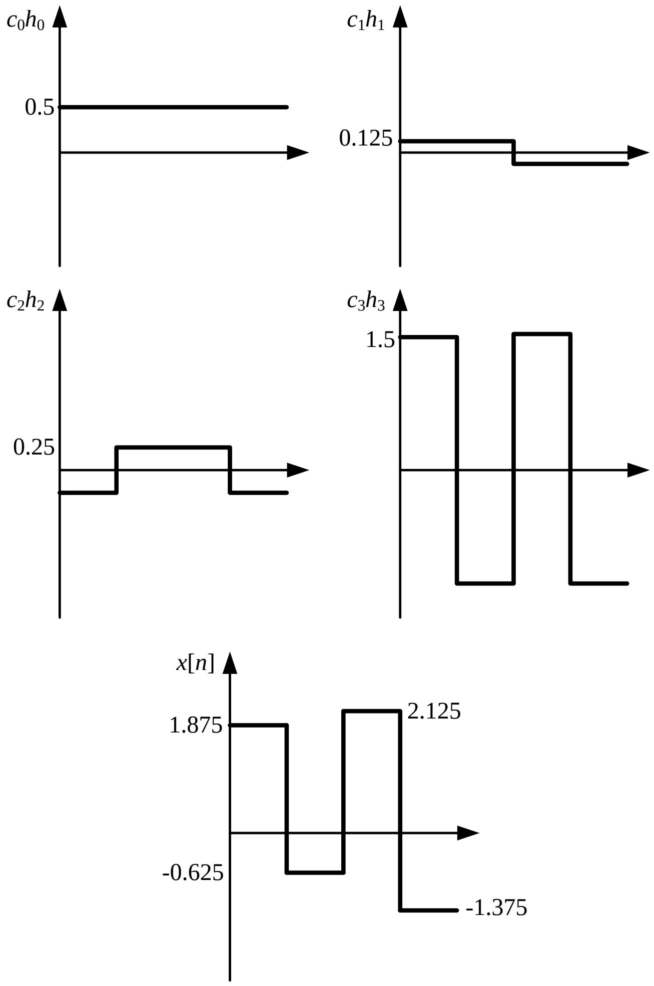
- The PSNR for this reconstruction is then:

$$\mathrm{PSNR} = 10 \cdot \log_{10}\left(\frac{16 \times 255^2}{4}\right) = 54.2\mathrm{dB}$$

———oooo———

Q5.6 Given the four point DWHT basis functions, if the coefficients after transformation are: $c(0) = 1$; $c(1) = \frac{1}{4}$; $c(2) = -\frac{1}{2}$; $c(3) = 3$, plot the weighted basis functions and hence reconstruct the original signal waveform.

Outline solution



———oooo———

Q5.7 The 1-D discrete cosine transform is given by:

$$C(k) = \sqrt{\frac{2}{N}} \varepsilon_k \sum_{n=0}^{N-1} x[n] \cos\left(\frac{\pi (n+0.5) k}{N}\right) \quad 0 \leq n, k \leq N-1$$

where:

$$\varepsilon_k = \begin{cases} 1/\sqrt{2} : & k = 0 \\ 1 : & otherwise \end{cases}$$

Calculate the numerical values of the basis functions for a 1-D 4 point DCT.

Outline solution

- The basis function matrix for the 4 point DCT is given by:

$$\mathbf{A}_4 = \sqrt{\frac{1}{2}} \begin{bmatrix} \sqrt{\frac{1}{2}} & \sqrt{\frac{1}{2}} & \sqrt{\frac{1}{2}} & \sqrt{\frac{1}{2}} \\ \cos(\frac{\pi}{8}) & \cos(\frac{3\pi}{8}) & -\cos(\frac{3\pi}{8}) & -\cos(\frac{\pi}{8}) \\ \cos(\frac{\pi}{4}) & -\cos(\frac{\pi}{4}) & -\cos(\frac{\pi}{4}) & \cos(\frac{\pi}{4}) \\ \cos(\frac{3\pi}{8}) & -\cos(\frac{\pi}{8}) & \cos(\frac{\pi}{8}) & -\cos(\frac{3\pi}{8}) \end{bmatrix}$$

- Evaluating numerical values to 3 decimal points gives:

$$\mathbf{A}_4 = \begin{bmatrix} 0.5 & 0.5 & 0.5 & 0.5 \\ 0.653 & 0.271 & -0.271 & -0.653 \\ 0.5 & -0.5 & -0.5 & 0.5 \\ 0.271 & -0.653 & 0.653 & -0.271 \end{bmatrix}$$

- This indicates potential issues with drift due to numerical mismatch between encoder and decoder. This is one of the reasons why integer transforms have been introduced in standards such as H.264/AVC.

——oooo——

Q5.8 The DCT is an orthonormal transform. Explain the term orthonormal and describe what it means in practice for the transform. Write down the formula and the basis function matrix for the 1-D 4-point DCT. Using this, show that, for the 1-D DCT:

$$\epsilon_k = \begin{cases} 1/\sqrt{2} & k = 0 \\ 1 & \text{otherwise} \end{cases}$$

Outline solution

- The DCT basis functions are given by:

$$a(k,n) = \sqrt{\frac{2}{N}}\varepsilon_k \cos\left(\frac{\pi k}{N}(n+0.5)\right); \ 0 \le k, n \le N-1$$

- For the case of a 4 point DCT, the basis function matrix is thus:

$$\mathbf{A} = \sqrt{\frac{2}{N}}\varepsilon_k \begin{bmatrix} \cos(0) & \cos(0) & \cos(0) & \cos(0) \\ \cos(\frac{\pi}{8}) & \cos(\frac{3\pi}{8}) & \cos(\frac{5\pi}{8}) & \cos(\frac{7\pi}{8}) \\ \cos(\frac{\pi}{4}) & \cos(\frac{3\pi}{4}) & \cos(\frac{5\pi}{4}) & \cos(\frac{7\pi}{4}) \\ \cos(\frac{3\pi}{8}) & \cos(\frac{9\pi}{8}) & \cos(\frac{15\pi}{8}) & \cos(\frac{21\pi}{8}) \end{bmatrix}$$

$$= \sqrt{\frac{1}{2}}\varepsilon_k \begin{bmatrix} \cos(0) & \cos(0) & \cos(0) & \cos(0) \\ \cos(\frac{\pi}{8}) & \cos(\frac{3\pi}{8}) & -\cos(\frac{3\pi}{8}) & -\cos(\frac{\pi}{8}) \\ \cos(\frac{\pi}{4}) & -\cos(\frac{\pi}{4}) & -\cos(\frac{\pi}{4}) & \cos(\frac{\pi}{4}) \\ \cos(\frac{3\pi}{8}) & -\cos(\frac{\pi}{8}) & \cos(\frac{\pi}{8}) & -\cos(\frac{3\pi}{8}) \end{bmatrix}$$

- Let us convince ourselves that this transform is orthonormal and confirm the value of ε_k. Consider for example, the first basis function. We can rewrite this as the vector $\mathbf{a}_0$, as follows:

$$\mathbf{a}_0 = k_0 \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}$$

- Now we know that for orthonormality the basis functions must have unity norm, thus:

$$\|\mathbf{a}_0\| = k_0\sqrt{1^2 + 1^2 + 1^2 + 1^2} = 2k_0 = 1$$

- Hence:

$$k_0 = \frac{1}{2}$$

- Similarly for the second basis function:

$$\|\mathbf{a}_1\| = k_1\sqrt{\cos^2(\frac{\pi}{8}) + \cos^2(\frac{3\pi}{8}) + \cos^2(\frac{3\pi}{8}) + \cos^2(\frac{\pi}{8})} = \sqrt{2}k_1 = 1$$

- Hence:

$$k_1 = \frac{1}{\sqrt{2}}$$

- And it can similarly be shown that:

$$k_2 = k_3 = \frac{1}{\sqrt{2}}$$

- So in general we have, as expected:

$$k_i = \frac{1}{\sqrt{2}}\varepsilon_i;\ \varepsilon_i = \begin{cases} 1/\sqrt{2}: & i = 0 \\ 1: & i \neq 0 \end{cases}$$

——oooo——

Q5.9 Compute the DCT transform coefficient vector for an input sequence, $\mathbf{x} = [1001]^{\mathrm{T}}$.

Outline solution

- The 1-D DCT for this input is given by:

$$\mathbf{C} = \begin{bmatrix} 0.5 & 0.5 & 0.5 & 0.5 \\ 0.653 & 0.271 & -0.271 & -0.653 \\ 0.5 & -0.5 & -0.5 & 0.5 \\ 0.271 & -0.653 & 0.653 & -0.271 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

———oooo———

Q5.10 Calculate the DCT of the following 2×2 image block:

$$\mathbf{X} = \begin{bmatrix} 21 & 19 \\ 15 & 20 \end{bmatrix}$$

Outline solution

$$\mathbf{C}_2 = \mathbf{A}\mathbf{X}\mathbf{A}^{\mathrm{T}}$$

$$\mathbf{C}_2 = \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & -1/\sqrt{2} \end{bmatrix} \begin{bmatrix} 21 & 19 \\ 15 & 20 \end{bmatrix} \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & -1/\sqrt{2} \end{bmatrix}$$

$$= \frac{1}{2} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 21 & 19 \\ 15 & 20 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

$$= \begin{bmatrix} 37.5 & -1.5 \\ 2.5 & 3.5 \end{bmatrix}$$

———oooo———

Q5.11 Quantise the result from Q5.9 using the following quantisation matrix:

$$\mathbf{Q} = \begin{bmatrix} 4 & 8 \\ 8 & 8 \end{bmatrix}$$

Outline solution

$$\mathbf{C}_{2Q} = \mathrm{rnd}\left(\mathbf{C}_2 /.\mathbf{Q}\right)$$

$$\mathbf{C}_{2Q} = \begin{bmatrix} \mathrm{rnd}({}^{37.5}/_{4}) & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 9 & 0 \\ 0 & 0 \end{bmatrix}$$

——oooo——

Q5.12 Perform inverse quantisation and an inverse DCT on the output from Q5.11.

Outline solution

- Perform inverse quantisation (rescaling) and inverse quantisation on the previous result:

$$\tilde{\mathbf{X}} = \begin{bmatrix} {}^{1}/_{\sqrt{2}} & {}^{1}/_{\sqrt{2}} \\ {}^{1}/_{\sqrt{2}} & -{}^{1}/_{\sqrt{2}} \end{bmatrix} \begin{bmatrix} 36 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} {}^{1}/_{\sqrt{2}} & {}^{1}/_{\sqrt{2}} \\ {}^{1}/_{\sqrt{2}} & -{}^{1}/_{\sqrt{2}} \end{bmatrix}$$

$$= \frac{1}{2} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 36 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

$$= \begin{bmatrix} 18 & 18 \\ 18 & 18 \end{bmatrix}$$

- Note: As expected, this gives a reconstruction error due to quantisation. Inverse transformation without quantisation would give the same values as in the original block - try it!

——oooo——

Q5.13 Compute the 2-D-DCT of the following (4×4) image block:

$$\mathbf{X} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

Outline solution

- Let:

$$\mathbf{C}_2 = \mathbf{C}_1 \mathbf{A}^{\mathrm{T}}$$

- Where:

$$\mathbf{C}_1 = \mathbf{A}\mathbf{X}$$

- If we evaluate to 3dp we have:

$$\mathbf{C}_1 = \begin{bmatrix} 0.5 & 0.5 & 0.5 & 0.5 \\ 0.653 & 0.271 & -0.271 & -0.653 \\ 0.5 & -0.5 & -0.5 & 0.5 \\ 0.271 & -0.653 & 0.653 & -0.271 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

- And hence:

$$\mathbf{C}_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0.5 & 0.653 & 0.5 & 0.271 \\ 0.5 & 0.271 & -0.5 & -0.653 \\ 0.5 & -0.271 & -0.5 & 0.653 \\ 0.5 & -0.653 & 0.5 & -0.271 \end{bmatrix} = \begin{bmatrix} 0.5 & 0.653 & 0.5 & 0.271 \\ 0 & 0 & 0 & 0 \\ 0.5 & 0.653 & 0.5 & 0.271 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

- Note that, due to finite precision arithmetic, numerical errors can be introduced. Hence if these coecients are inverse-transformed, then the result may differ from the original. This problem can be addressed by using higher precision arithmetic or by employing integer transforms, such as that used in H.264/AVC.

———oooo———

Q5.14 Given the following block of DCT coefficients and the associated quantisation matrix, compute the block of quantised coefficients. Perform zig-zag scanning to form a string of {run/value} symbols (where 'run' is the number of zeros preceding a non-zero value) appropriate for entropy coding.

$$\mathbf{C} = \begin{bmatrix} 128 & 50 & -20 & 22 & 12 & 27 & -5 & 7 \\ 40 & -25 & 26 & 20 & -34 & -2 & 13 & -5 \\ -10 & 22 & 12 & 12 & 26 & 12 & 3 & 8 \\ 12 & -2 & 16 & -7 & 9 & 3 & 17 & 17 \\ -32 & 6 & 21 & 9 & 18 & 5 & 4 & 7 \\ -10 & -7 & -14 & 3 & -2 & 13 & 18 & 18 \\ 11 & -9 & -9 & 4 & 8 & 13 & 6 & 9 \\ -7 & 19 & 15 & 8 & 6 & -6 & 18 & 33 \end{bmatrix}$$

$$\mathbf{Q} = \begin{bmatrix} 8 & 16 & 19 & 22 & 26 & 27 & 29 & 34 \\ 16 & 16 & 22 & 24 & 27 & 29 & 34 & 37 \\ 19 & 22 & 26 & 27 & 29 & 34 & 34 & 38 \\ 22 & 22 & 26 & 27 & 29 & 34 & 37 & 40 \\ 22 & 26 & 27 & 29 & 32 & 35 & 40 & 48 \\ 26 & 27 & 29 & 32 & 35 & 40 & 48 & 58 \\ 26 & 27 & 29 & 34 & 38 & 46 & 56 & 69 \\ 27 & 29 & 35 & 38 & 46 & 56 & 69 & 83 \end{bmatrix}$$

Outline solution

- The quantised coefficient matrix is given by (assuming 0.5 values rounded up):

$$\mathbf{C}_{\mathrm{Q}} = \begin{bmatrix} 16 & 3 & -1 & 1 & 0 & 1 & 0 & 0 \\ 3 & -2 & 1 & 1 & -1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

- Performing zig zag scanning:

$$\mathbf{C}_{\mathrm{Q}} = \begin{bmatrix} 16 & 3 & -1 & 1 & 0 & 1 & 0 & 0 \\ 3 & -2 & 1 & 1 & -1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

- The sequence of run value symbols is thus given by:

$$\{0/16,\ 0/3,\ 0/3,\ 0/-1,\ 0/-2,\ 0/-1,\ 0/1,\ 0/1,\ 0/1,\ 0/1,\ 0/-1,\ 2/1,\ 1/1, \cdots$$

$$\cdots\ 0/-1,\ 1/1,\ 4/1,\ 1/1,\ 13/1,\ \mathrm{EOB}\}$$

- Note: In practice, the DC coefficient value may be coded independently of the AC values.

——oooo——

Q5.15 Calculate the number of multiply and accumulate (MAC) operations required to compute a conventional (4×4) point 2-D-DCT. Assume that the separability property of the 2-D-DCT is exploited.

Outline solution

- Assuming separability, the basic 4 point 2-D DCT can be calculated as follows:

$$\mathbf{C} = \mathbf{A}\mathbf{S}\mathbf{A}^{\mathrm{T}}$$

- Considering the dimensions of the matrix calculation:

$$\mathbf{C} = [4 \times 4] \times [4 \times 4] \times [4 \times 4]$$

- Each matrix product requires a total of 16×4 = 64 MACs, thus giving a total for the calculation of 128 MACs.

——oooo——

Q5.16 The complexity of the DCT can be reduced using 'Fast' methods such as McGovern's Algorithm. Derive McGovern's algorithm for a 4-point 1-D-DCT. Compare its complexity (again assuming exploitation of separability) with that of the conventional approach for the case of a (4×4) point 2-D-DCT.

Outline solution

- The basic 1-D 4 point DCT can be written as:

$$\begin{bmatrix} c(0) \\ c(1) \\ c(2) \\ c(3) \end{bmatrix} = \begin{bmatrix} a_2 & a_2 & a_2 & a_2 \\ a_1 & a_3 & -a_3 & -a_1 \\ a_2 & -a_2 & -a_2 & a_2 \\ a_3 & -a_1 & a_1 & -a_3 \end{bmatrix} \begin{bmatrix} x[0] \\ x[1] \\ x[2] \\ x[3] \end{bmatrix}$$

- where

$$a_i = \cos\left(i\frac{\pi}{8}\right)$$

- Splitting into even and odd indexed rows. we can rearrange as follows:

$$\begin{bmatrix} c(0) \\ c(2) \end{bmatrix} = \begin{bmatrix} a_2 & a_2 \\ a_2 & -a_2 \end{bmatrix} \begin{bmatrix} x[0] + x[3] \\ x[1] + x[2] \end{bmatrix} \tag{1}$$

$$\begin{bmatrix} c(1) \\ c(3) \end{bmatrix} = \begin{bmatrix} a_1 & a_3 \\ a_3 & -a_1 \end{bmatrix} \begin{bmatrix} x[0] - x[3] \\ x[1] - x[2] \end{bmatrix} \tag{2}$$

- Equation (1) can be further subdivided as follows:

$$c(0) = a_2 (x[0] + x[1] + x[2] + x[3])$$

$$c(2) = a_2 (x[0] - x[1] - x[2] + x[3])$$

- These two equations can be implemented with 2 multiplies and 4 additions.
- The second equation (2) can be simplified using standard rotator products:

$$\begin{bmatrix} c(1) \\ c(3) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} (a_1 - a_3)(x[0] - x[3]) \\ a_3 (x[0] + x[1] - x[2] - x[3]) \\ (a_3 - a_1)(x[1] - x[2]) \end{bmatrix}$$

- This requires an additional 3 multiplies and 7 additions. The total computation requirement for the 4 point 1-D DCT using this approach is 5 multiplication and 11 addition operations.
- The computational load for a 2-D DCT implemented using this approach, exploiting separability, is thus: 40 multiplications and 88 additions.

——oooo——

Q5.17 Compute the SATD (sum of Absolute Transformed Differences) for the image blocks **X** and **Y** in Q4.7.

Outline solution SATD is defined as the sum of absolute transformed (DWHT is used here) difference between two blocks, so we first calculate the block difference:

$$\mathbf{D} = \mathbf{X} - \mathbf{Y} = \begin{bmatrix} 3 & 8 & 1 & 8 \\ 7 & 0 & 5 & 0 \\ 2 & 6 & 0 & 5 \\ 4 & 1 & 10 & 2 \end{bmatrix} - \begin{bmatrix} 4 & 10 & 1 & 9 \\ 7 & 0 & 6 & 1 \\ 3 & 6 & 0 & 4 \\ 5 & 1 & 12 & 3 \end{bmatrix} = \begin{bmatrix} -1 & -2 & 0 & -1 \\ 0 & 0 & -1 & -1 \\ -1 & 0 & 0 & 1 \\ -1 & 0 & -2 & -1 \end{bmatrix}$$

Then we calculate the 2D DWHT of the difference block:

$$\mathbf{C} = \mathbf{H}\mathbf{D}\mathbf{H}^{\mathrm{T}} = \frac{1}{4} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \begin{bmatrix} -1 & -2 & 0 & -1 \\ 0 & 0 & -1 & -1 \\ -1 & 0 & 0 & 1 \\ -1 & 0 & -2 & -1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix}^{\mathrm{T}}$$

$$\mathbf{C} = \begin{bmatrix} -2.5 & 0 & 0 & -0.5 \\ -0.5 & 0 & 0 & 1.5 \\ -1.5 & 0 & 0 & 0.5 \\ 0.5 & -2 & 0 & 0.5 \end{bmatrix}$$

Finally the sum of absolute transformed difference is calculated for all 16 pixels:

$$\text{SATD} = \sum_{i,j=1}^{16} |c_{i,j}| = 10$$

where $c_{i,j} \in \mathbf{D}$.

Chapter 6: Filterbank methods

Q6.1 For a filterbank downsampler, show that the frequency domain behaviour of the output signal, $x_d[n]$ is related to that of the input $x[n]$ by:

$$X_d(\Omega) = 0.5\left[X\left(e^{j\Omega/2}\right) + X\left(e^{-j\Omega/2}\right)\right]$$

Outline solution

- From chapter 6, the input-output relationship for a downsampler is given by $x_d[n] = x[nM]$. The z transform of the downsampled signal is then given by:

$$X_d(z) = \sum_{n=-\infty}^{\infty} x_d[n]\, z^{-n} = \sum_{n=-\infty}^{\infty} x[nM]\, z^{-n}$$

- Substituting m=nM:

$$X_d(z) = \sum_{m=-\infty}^{\infty} x'[m]\, z^{-m/M} = X'\left(z^{1/M}\right) \tag{3}$$

- Note that $x'[n]$ is not the same as $x[n]$. They can however be related by $x'[n] = c[n]\, x[n]$, where:

$$c[n] = \begin{cases} 1; & n = 0, \pm M, \pm 2M \ldots \\ 0; & \text{otherwise} \end{cases}$$

- where $c[n]$ can be represented, using the DFT, as:

$$c[n] = \frac{1}{M}\sum_{k=0}^{M-1} W_M^{nk}$$

- For the case of M=2, we have:

$$c[n] = \frac{1}{2}\left(1 + (-1)^n\right)$$

- Now the z-transform of $x'[n]$ is given by:

$$X'(z) = \frac{1}{M}\sum_{n=-\infty}^{\infty}\left(\sum_{k=0}^{M-1} W_M^{nk}\right) x[n]\, z^{-n} = \frac{1}{M}\sum_{k=0}^{M-1}\left(\sum_{n=-\infty}^{\infty} x[n]\, W_M^{nk} z^{-n}\right)$$

$$\frac{1}{M}\sum_{k=0}^{M-1} X\left(zW_M^{-k}\right) \tag{4}$$

- Equation (4) can now be substituted into equation (3) to give:

$$X_d(z) = \frac{1}{M}\sum_{k=0}^{M-1} X\left(z^{1/M} W_M^{-k}\right)$$

- Consider the case where M=2:

$$X_d(z) = \frac{1}{2}\left(X\left(z^{1/2}\right) + X\left(-z^{1/2}\right)\right)$$

- Evaluating this on the unit circle in the z-plane gives the frequency response, thus:

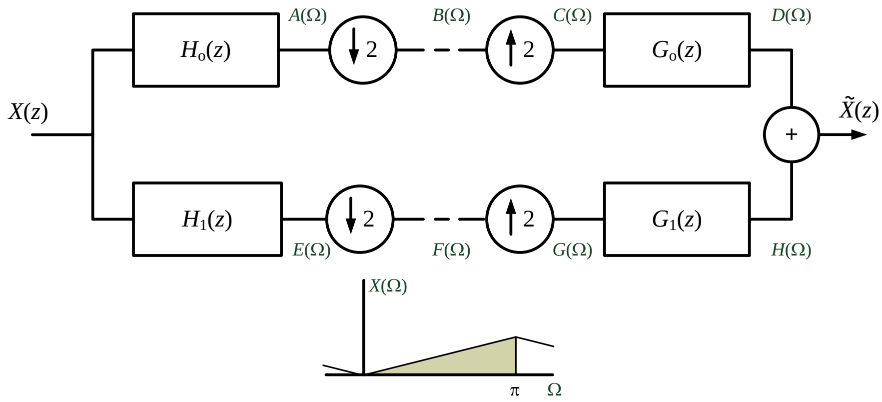
$$X_d\left(e^{j\Omega}\right) = \frac{1}{2}\left(X\left(e^{j\Omega/2}\right) + X\left(-e^{j\Omega/2}\right)\right)$$

- or equivalently:

$$X_d\left(e^{j\Omega}\right) = \frac{1}{2}\left(X\left(e^{j\Omega/2}\right) + X\left(e^{j\Omega/2+\pi}\right)\right)$$

———oooo———

Q6.2 The figure below shows a simple two-band analysis-synthesis filterbank and a representative input spectrum.



It can be shown that the output of this system is given by:

$$\tilde{X}(z) = \frac{1}{2}X(z)\left[G_0(z)H_0(z) + G_1(z)H_1(z)\right] + \frac{1}{2}X(-z)\left[G_0(z)H_0(-z) + G_1(z)H_1(-z)\right]$$

where the upsampler and downsampler relationships are given by:

$$C\left(e^{j\Omega}\right) = B\left(e^{j2\Omega}\right)$$

$$B\left(e^{j\Omega}\right) = 0.5\left[A\left(e^{j\Omega/2}\right) + A\left(-e^{j\Omega/2}\right)\right]$$

Using the up-sampler and down-sampler relationships given above, compute and sketch the spectra at points A to H. Assume that all the filters have ideal brickwall responses. Hence demonstrate graphically that the system is capable of perfect reconstruction.

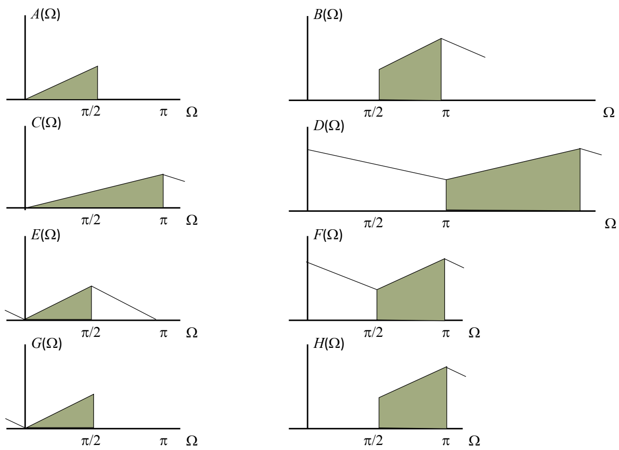
Outline solution

- Let us assume that all filters are ideal and exhibit brickwall responses. The upsampling and downsampling relationships are given by:

$$C\left(e^{j\Omega}\right) = B\left(e^{j2\Omega}\right)$$

$$B\left(e^{j\Omega}\right) = 0.5\left[A\left(e^{j\Omega/2}\right) + A\left(e^{j(\Omega/2+\pi)}\right)\right]$$

- Thus:



——oooo——

Q6.3 Given the following subband filters:

$$H_0(z) = \frac{1}{\sqrt{2}}\left(1 + z^{-1}\right); \quad H_1(z) = \frac{1}{\sqrt{2}}\left(1 - z^{-1}\right)$$
$$G_0(z) = \frac{1}{\sqrt{2}}\left(1 + z^{-1}\right); \quad G_1(z) = \frac{1}{\sqrt{2}}\left(-1 + z^{-1}\right)$$

Show that the corresponding two band filterbank exhibits perfect reconstruction.

Outline solution

- For perfect reconstruction we must satisfy:

$$F_0(z) = \frac{1}{2}\left[H_0\left(z\right)G_0\left(z\right) + H_1\left(z\right)G_1\left(z\right)\right] = cz^{-k}$$

- and for alias-free operation, we must satisfy:

$$F_1(z) = \frac{1}{2}\left[H_0\left(-z\right)G_0\left(z\right) + H_1\left(-z\right)G_1\left(z\right)\right] = 0$$

- We can see from the second equation that this filter bank is alias-free, because:

$$F_1(z) = \frac{1}{4}\left[\left(1-z^{-1}\right)\left(1+z^{-1}\right) + \left(1+z^{-1}\right)\left(-1+z^{-1}\right)\right] = 0$$

- We can further show that the filterbank exhibits perfect reconstruction since:

$$F_0(z) = \frac{1}{4}\left[\left(1+z^{-1}\right)\left(1+z^{-1}\right) - \left(1-z^{-1}\right)\left(1-z^{-1}\right)\right] = z^{-1}$$

——oooo——

Q6.4 Demonstrate that the following filter relationships produce a filterbank that is alias-free and offers perfect reconstruction:

$$H_1\left(z\right) = zG_0\left(-z\right); \qquad G_1\left(z\right) = z^{-1}H_0\left(-z\right);$$
$$P\left(z\right) + P\left(-z\right) = 2z^{-2}; \qquad P\left(z\right) = H_0\left(z\right)G_0\left(z\right)$$

Assuming that $G_0\left(z\right) = H_0\left(z\right) = z^{-1}$, what is the output of this filterbank given an input sequence {1,1,0}?

Outline solution

- For alias-free operation, we must satisfy:

$$F_1(z) = \frac{1}{2}\left[H_0\left(-z\right)G_0\left(z\right) + H_1\left(-z\right)G_1\left(z\right)\right] = 0$$

- We can thus see that this filter bank is alias-free, because:

$$F_1(z) = \frac{1}{2}\left[\left(-z^{-1}\right)\left(z^{-1}\right) + z\left(-z^{-1}\right)\left(z^{-1}\right)\left(-z^{-1}\right)\right] = 0$$

- For perfect reconstruction we must satisfy:

$$F_0(z) = \frac{1}{2}\left[H_0\left(z\right)G_0\left(z\right) + H_1\left(z\right)G_1\left(z\right)\right] = cz^{-k}$$

- Thus:

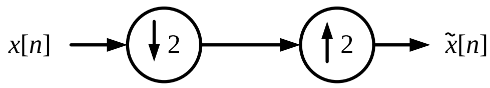
$$F_0(z) = \frac{1}{2}\left[\left(z^{-1}\right)\left(z^{-1}\right) + z\left(-z^{-1}\right)\left(z^{-1}\right)\left(-z^{-1}\right)\right] = z^{-2}$$

- For an input sequence of {1,1,0...}, the output will simply be a delayed version of this, i.e. {0,0,1,1,0...}.

——oooo——

Q6.5 Referring to the figure below, a signal, $x[n]$, is downsampled by 2 and then upsampled by 2. Show that, in the z-domain, the input–output relationship is given by:

$$\tilde{X}\left(z\right) = 0.5\left[X\left(z\right) + X\left(-z\right)\right]$$



Outline solution

- The given upsampler-downsampler combination will produce an output sequence: $\left\{\begin{array}{ccccc} x\left[0\right] & 0 & x\left[2\right] & 0 & \cdots\end{array}\right\}$. Let us see if the given equation produces the same result. The z-transform of the original sampled signal, $x\left[n\right]$, is:

$$X\left(z\right) = \sum_{n=-\infty}^{\infty} x\left[n\right] z^{-n}$$

- Hence:

$$X\left(-z\right) = \sum_{n=-\infty}^{\infty} x\left[n\right]\left(-z\right)^{-n}$$

- So, substituting these expressions into the equation given in the question, we have:

$$\tilde{X}(z) = 0.5\left[X(z) + X(-z)\right] = 0.5\left[\sum_{n=-\infty}^{\infty} x[n]\, z^{-n}\left(1 + (-1)^n\right)\right]$$

- Where:

$$\left(1 + (-1)^n\right) = \begin{cases} 2: & n \text{ even} \\ 0: & n \text{ odd} \end{cases}$$

- This clearly produces an operation identical to the downsampler-upsampler combination.

———oooo———

Q6.6 Demonstrate that the following quadrature mirror filter relationships produce a filter bank that is alias-free and offers perfect reconstruction:

$$\begin{array}{ll} H_0(z) = z^{-2} + z^{-3}; & G_0(z) = H_1(-z); \\ H_1(z) = H_0(-z); & G_1(z) = -H_0(-z) \end{array}$$

What are the limitations of the above filterbank? How, in practice, are more useful QMF filterbanks designed?

Outline solution The filters are given by:

$$\begin{array}{ll} H_0(z) = z^{-2} + z^{-3}; & G_0(z) = z^{-2} + z^{-3}; \\ H_1(z) = z^{-2} - z^{-3}; & G_1(z) = -z^{-2} + z^{-3} \end{array}$$

- For perfect reconstruction we must satisfy:

$$F_0(z) = \frac{1}{2}\left[H_0(z)\, G_0(z) + H_1(z)\, G_1(z)\right] = cz^{-k}$$

- For alias-free operation, we must satisfy:

$$F_1(z) = \frac{1}{2}\left[H_0(-z)\, G_0(z) + H_1(-z)\, G_1(z)\right] = 0$$

- We can see from the second equation that this filter bank is alias-free, because:

$$F_1(z) = \frac{1}{2}\left[\left(z^{-2} - z^{-3}\right)\left(z^{-2} + z^{-3}\right) + \left(z^{-2} + z^{-3}\right)\left(-z^{-2} + z^{-3}\right)\right] = 0$$

- We can further show that the filterbank exhibits perfect reconstruction since:

$$F_0(z) = \frac{1}{2}\left[\left(z^{-2}+z^{-3}\right)\left(z^{-2}+z^{-3}\right)+\left(z^{-2}-z^{-3}\right)\left(-z^{-2}+z^{-3}\right)\right] = 2z^{-5}$$

- In practice, approximate QMF filters can be usefully designed where the characteristic does not exactly provide perfect reconstruction. In such cases, the amplitude distortion can be minimised using an optimisation procedure to iteratively adjust the filter coefficients. We ideally want the magnitude response of the filter characteristic to have a sharp transition so that subbands are spectrally distinct while providing an overall system response that is approximately all-pass. This means that the filters have to be designed so that each branch delivers an attenuation of 6dB (0.5) at the point where the low- and high-pass responses intersect - at one quarter of the sampling frequency. Each of the four filters must therefore have an attenuation of 3dB at the half Nyquist frequency.
- Such filters often minimise the residual amplitude distortion using an objective measure, based upon stop-band attenuation and the requirement to maintain the power symmetry property of the filters. Although these filters do not provide perfect reconstruction, for compression applications where significant quantisation error is introduced prior to transmission of the subband coefficients, this small distortion may not be a major issue. However, wavelet filters can be designed that do offer perfect reconstruction.

——oooo——

Q6.7 A two-channel single stage 1-D QMF filterbank is constructed using the low-pass prototype filter $H_0(z) = (1+z^{-1})$. Derive the other filters needed for this system and compute the output signal for an input: $x[n] = \{2, 3, 6, 4\}$.

Outline solution

- We can base our QMF filter design on the following relationships:

$$H_0(z) = H(z)\,;\ H_1(z) = H(-z)$$

$$G_0(z) = H(z)\,;\ G_1(z) = -H(-z)$$

- Thus the filters are:

$$\begin{array}{ll} H_0(z) = \left(1+z^{-1}\right); & H_1(z) = \left(1-z^{-1}\right); \\ G_0(z) = \left(1+z^{-1}\right); & G_1(z) = \left(-1+z^{-1}\right) \end{array}$$

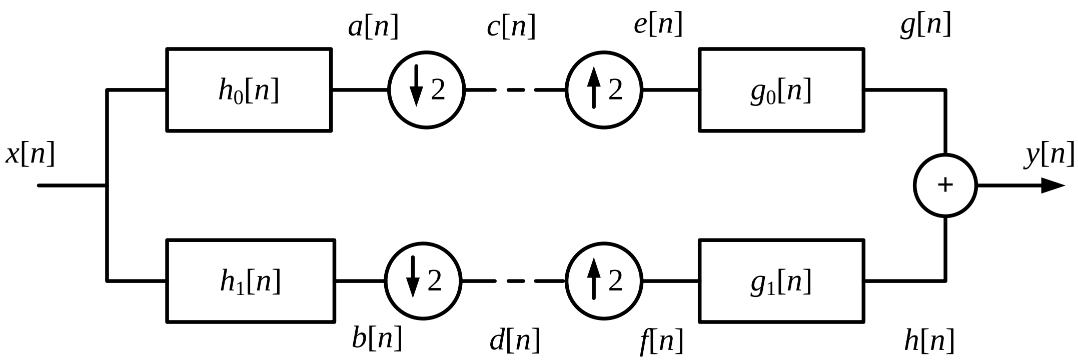
- In this case, the scaling factor of $1/\sqrt{2}$ has been omitted so we expect our response to be scaled by a constant of 2 as well as exhibit a delay of one sample. For an input sequence of $x[n] = \{2, 3, 6, 4\}$, the output will therefore be $y[n] = \{0, 4, 6, 12, 8, 0...\}$.

——oooo——

Q6.8 A 1-D wavelet filterbank comprises two stages of decomposition and uses the same filters as defined in Q6.7 (but factored by $1/\sqrt{2}$ to ensure exact reconstruction). The input sequence is $x[n] = \{1, 1, 1, 1, 1, 1, 1, 1\}$. Using boundary extension, and assuming critical sampling between analysis and synthesis banks, compute the signal values at all internal nodes in this filter bank, and hence demonstrate that the output sequence is identical to the input sequence.

Outline solution

- The input bitstream to the filterbank is $x[n] = \{1, 1, 1, 1, 1, 1, 1, 1\}$. We can compute the convolution of the filters with the input sequence at all intermediate nodes, as defined in the following figure:



- Let us assume that boundary extension is used. The signal values at each intermediate node are given in the following table (ignoring for convenience for for the time being the $1/\sqrt{2}$ factors in the filter responses).

$x[n]$	1 1 1 1 1 1 1 1
$a[n]$	1 2 2 2 2 2 2 2 1
$b[n]$	1 0 0 0 0 0 0 0 -1
$c[n]$	1 2 2 2 1
$d[n]$	1 0 0 0 -1
$e[n]$	1 0 2 0 2 0 2 0 1
$f[n]$	1 0 0 0 0 0 0 0 -1
$g[n]$	1 1 2 2 2 2 2 2 1 1
$h[n]$	-1 1 0 0 0 0 0 0 1 -1
$y[n]$	**0 2 2 2 2 2 2 2 2 0**

- As can be observed, when factored by $1/2$ to account for the filter weights, this provides perfect reconstruction with a delay of 1 sample.

——oooo——

Q6.9 Repeat Q6.8, but this time assume that the bit-allocation strategy employed preserves the low frequency subband but completely discards the high-pass subband. Assuming a wordlength of 4 bits, what is the PSNR of the reconstructed signal after decoding?

Outline solution

- If the high-pass subband is completely discarded during transmission then:

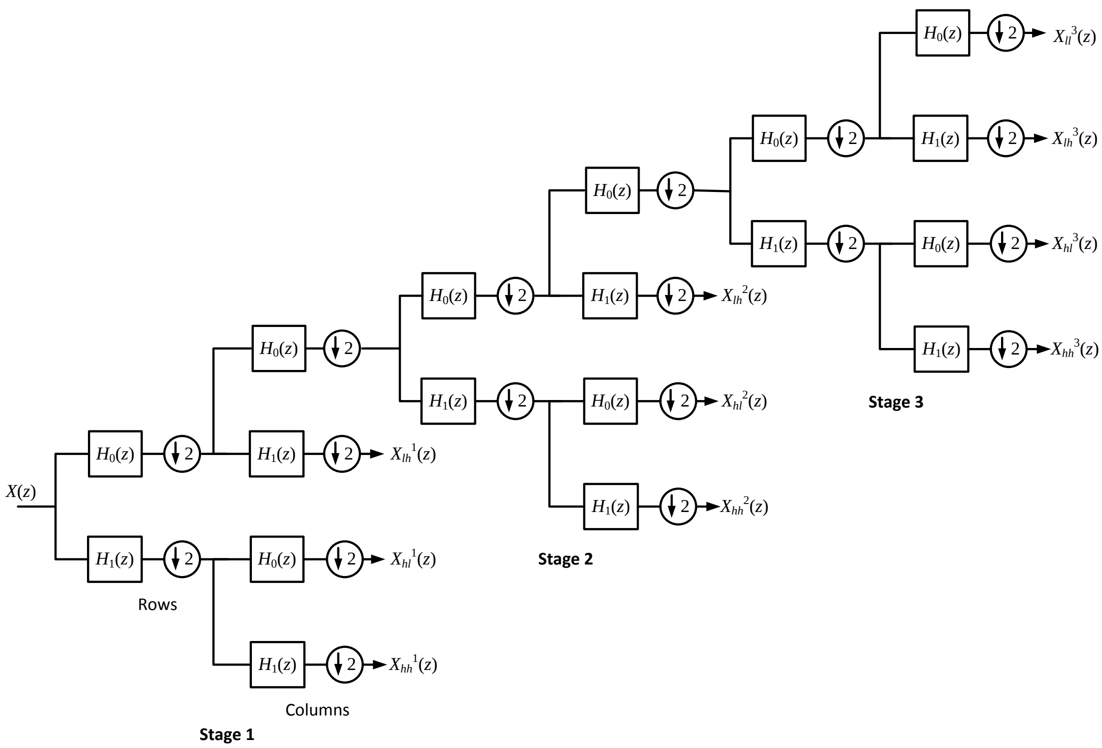
$$y[n] = g\,[n] = \{1, 1, 2, 2, 2, 2, 2, 2, 1, 1\}$$

- Ignoring the first and last samples, the output can be compared to the input using PSNR (assuming a wordlength of 4 bits) to give PSNR=60dB.

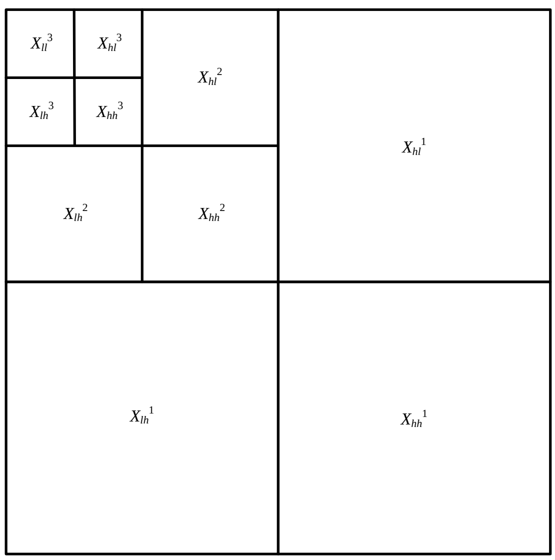
——oooo——

Q6.10 Draw the diagram for a 2-D, three-stage wavelet filter bank. Show how this decomposition tiles the 2-D spatial frequency plane and compute the frequency range of each subband. Assuming that the input is of dimensions 256×256 pixels, how many samples are contained in each subband?

Outline solution



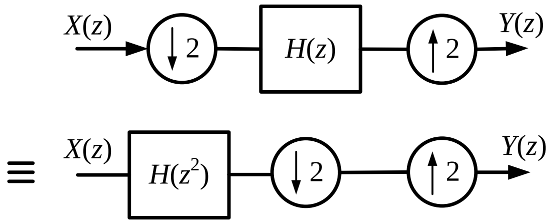
- Note: Although not explicily shown the data path must be transposed between the row and column processing and between each stage in the above diagram.
- Frequency domain tiling:



- With an input image size of 256×256 pixels, the stage 1 subbands are of dimension 128×128 coefficients, the stage 2 subbands are 64×64, and the stage 3 subbands are 32×32.

——oooo——

Q6.11 Prove that the following two diagrams are equivalent.



Outline solution

- Consider an intermediate point, which we will refer to as $Y'(z)$, prior to the final upsampler. In the first case, of the downsampler followed by a filter $H(z)$, we have:

$$Y'(z) = \frac{1}{2}H(z)\left(X\left(z^{\frac{1}{2}}\right) + X\left(-z^{\frac{1}{2}}\right)\right)$$

- And, in the second case, of the modified filter followed by the downsampler, we have:

$$Y'(z) = \frac{1}{2}\left(X\left(z^{\frac{1}{2}}\right)H(z) + X\left(-z^{\frac{1}{2}}\right)H(z)\right)$$

- These two equations are clearly identical. The final output $Y(z)$ in both cases is then:

$$Y(z) = Y'\left(z^2\right) = \frac{1}{2}\left(X(z)\,H(z^2) + X(-z)\,H(z^2)\right)$$

——oooo——

Chapter 7: Lossless compression methods

Q7.1 Consider the following codewords for the given set of symbols:

Symbol	C1	C2	C3
a_1	0	0	01
a_2	10	01	10
a_3	110	110	11
a_4	1110	1000	001
a_5	1111	1111	010

Identify which are prefix codes.

Outline solution

- The only prefix code in this table is C1, as all other codes are ambiguous. For example in C2, a_1 is the prefix of a_2 and in C3 a_1 is the prefix of a_5.

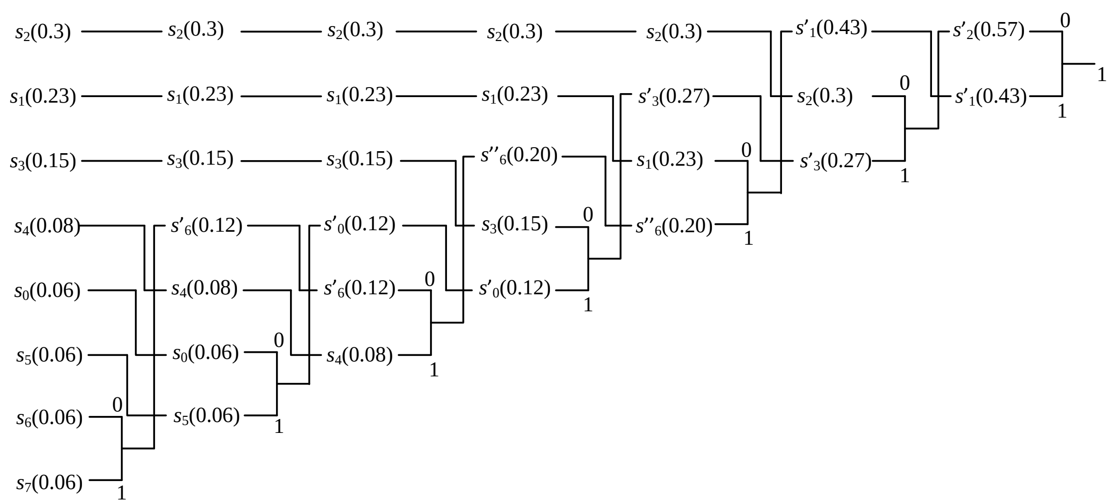
——oooo——

Q7.2 Derive the set of Huffman code words for the symbol set with the following probabilities:

$$P(s_0) = 0.06;\ P(s_1) = 0.23;\ P(s_2) = 0.30;\ P(s_3) = 0.15;$$
$$P(s_4) = 0.08;\ P(s_5) = 0.06;\ P(s_6) = 0.06;\ P(s_7) = 0.06;$$

What is the transmitted binary sequence corresponding to the symbol pattern s_0, s_4, s_6? What symbol sequence corresponds to the code sequence: 000101110? Calculate the first order entropy of this symbol set, the average codeword length and the coding redundancy.

Outline solution



- The transmitted binary sequence corresponding to the symbol pattern s_0, s_4, s_6 is 0110 111 1100.

- The symbol sequence corresponding to the code sequence: 000101110 is s_2, s_3, s_4
- The 1st order entropy for this alphabet is given by:

$$H = -\sum P \log_2 P$$

$$= -(0.06 \times \log_2 0.06 + 0.023 \times \log_2 0.23 + 0.3 \times \log_2 0.3 + \ldots + 0.06 \times \log_2 0.06)$$

$$= 2.6849\text{bits/symbol}$$

- The average codeword length can be computed from the following table. The average length $\bar{l}$ in this case is 2.71 bits/symbol.

Symbol	**Prob.**	**Code**	**Av. length**
s_0	0.06	0110	0.24
s_1	0.23	10	0.46
s_2	0.3	00	0.6
s_3	0.15	010	0.45
s_4	0.08	111	0.24
s_5	0.06	0111	0.24
s_6	0.06	1100	0.24
s_7	0.06	1101	0.24
Overall average length:			**2.71**

- The redundancy of this encoding is:

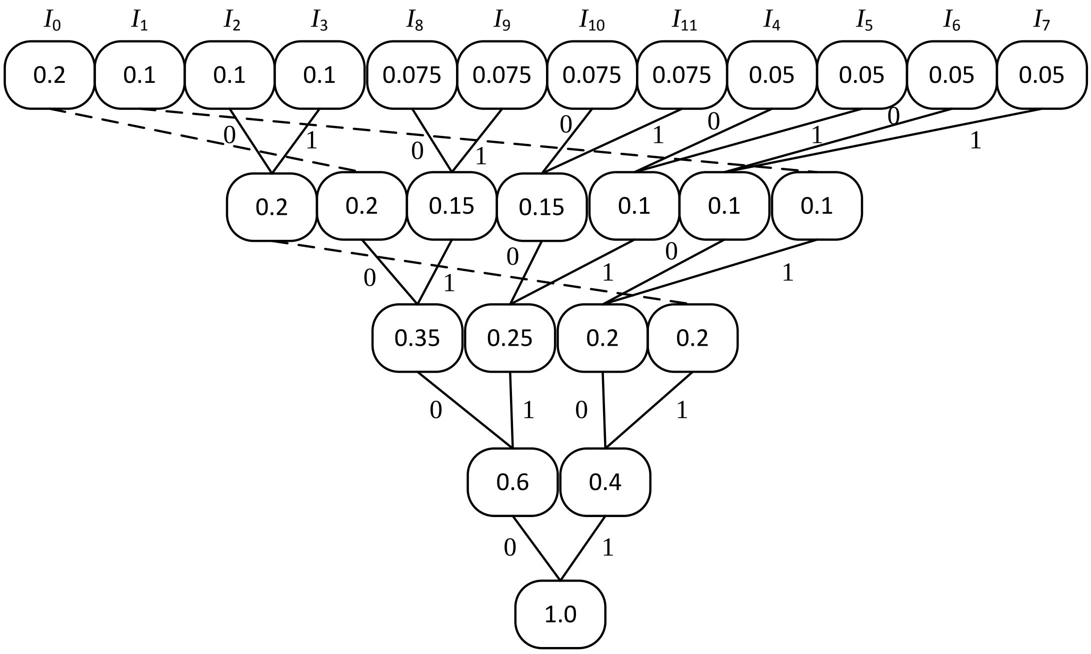
$$R = \frac{\bar{l} - H}{H} \times 100 = \frac{2.71 - 2.6849}{2.6849} \approx 1\%$$

——oooo——

Q7.3 A quantised image is to be encoded using the symbols $I \in \{I_0 \cdots I_{11}\}$. From simulation studies it has been estimated that the relative frequencies for these symbols are as follows: I_0: 0.2; $I_1..I_3$: 0.1; $I_4..I_7$: 0.05; $I_8..I_{11}$: 0.075. Construct the Huffman tree for these symbols and list the resultant code words in each case.

Outline solution

- Using the minimum variance approach, but employing a more compact diagram for convenience here. we have:



- Giving the following codewords:

Symbol	**Prob.**	**Code**
I_0	0.2	000
I_1	0.1	101
I_2	0.1	110
I_3	0.1	111
I_4	0.05	0110
I_5	0.05	0111
I_6	0.05	1000
I_7	0.05	1001
I_8	0.075	0010
I_9	0.075	0011
I_{10}	0.075	0100
I_{11}	0.075	0101

———oooo———

Q7.4 If the minimum variance set of Huffman codes for an alphabet, A, is as shown in the table below, determine the efficiency of the corresponding Huffman encoder.

Symbol	Probability	Huffman Code
a_1	0.5	0
a_2	0.25	10
a_3	0.125	110
a_4	0.0625	1110
a_5	0.0625	1111

Outline solution

- The first order entropy for this set of symbol probabilities is:

$$H = -\sum P \log_2 P = 1.875\text{bits/symbol}$$

- The average codeword length is:

$$\bar{l} = \sum l_i P_i = 1.875\text{bits/symbol}$$

- The efficiency is thus:

$$E = \frac{H}{\bar{l}} = 100\%$$

——oooo——

Q7.5 Consider the following 4×4 matrix, **C**, of DCT coefficients, produced from a block-based image coder. Using zig zag scanning and run length coding (assume a {*run*, *value*} model where *value* is the integer value of a non-zero coefficient and *run* is the number of zeros preceding it), determine the transmitted bitstream after entropy coding.

$$\mathbf{C} = \begin{bmatrix} 2 & 0 & 3 & 0 \\ 0 & 3 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Use the following symbol to codeword mappings:

Symbol	**Code**	**Symbol**	**Code**
{0,1}	00	{1,3}	0111
{0,2}	10	{2,1}	0110
{0,3}	110	{2,2}	0101
{1,1}	1111	{2,3}	01001
{1,2}	1110	EOB	01000

Outline solution

- The {RUN,VALUE} symbols for **C** are (assuming the DC value is treated in the same way as the AC values here):

$$(0,2) \quad (2,1) \quad (0,3) \quad (0,3) \quad (1,1) \quad (\text{EOB})$$

- The corresponding codewords are:

$$(10) \quad (0110) \quad (110) \quad (110) \quad (1111) \quad (01000)$$

——oooo——

Q7.6 Assuming that the DCT coefficient matrix and quantisation matrix in Q5.14 form part of a JPEG baseline codec, derive the Huffman coded sequence that would be produced by the codec for the AC coefficients.

Outline solution

- Considering the AC coefficients only, the corresponding code words are shown in the following table:

Coeff	**run/size**	**Huffman code**	**Amplitude**	**#bits**
3	0/2	01	11	4
-3	0/2	01	11	4
-1	0/1	00	0	3
-2	0/2	01	01	4
-1	0/1	00	0	3
1	0/1	00	1	3
1	0/1	00	1	3
1	0/1	00	1	3
1	0/1	00	1	3
-1	0/1	00	0	3
1	2/1	11100	1	6
1	1/1	1100	1	5
-1	0/1	00	0	3
1	1/1	1100	1	5
1	4/1	111011	1	7
1	1/1	1100	1	5
1	10/1	111111010	1	10
1	2/1	11100	1	6
EOB	0/0	1010	-	4
Total AC bits				84

- The coded bitstream is formed by the concatenation of the sequence of Huffman codes and amplitude values from the above table, followed by the EOB code.

——oooo——

Q7.7 What is the Exp-Golomb code for the symbol index 132_{10}?

Outline solution

- The codeword group index is given by:

$$\zeta_i = \lfloor \log_2 (i+1) \rfloor = \lfloor \log_2 (133) \rfloor = 7$$

- And the residual is given by:

$$\nu_i = 1 + i - 2^{\zeta_i} = 133 - 2^7 = 5$$

- The Exp-Golomb code for the symbol index $132_{10}$is thus: 000000010000101

——oooo——

Q7.8 Show how the Exp-Golomb codeword 000010101 would be decoded and compute the value of the corresponding symbol index. What is the corresponding Golomb-Rice code for this index (assume m=4)?

Outline solution The Exp-Golomb codeword:

- To decode an Exp-Golomb codeword we count the number of leading zeros (ζ_i), ignore the following 1 and then decode the index i from the next ζ_i bits. In this case there are 4 leading zeros and the encoded residual $\nu_i = 0101_2 = 5_{10}$.
- The index can then be calculated as:

$$i = \nu_i - 1 + 2^{\zeta_i} = 5 - 1 + 16 = 20$$

The corresponding Golomb-Rice code for 20:

- The group size for 20 is m=4.
- The remainder is given by:

$$\nu_i = i - \left\lfloor \frac{i}{m} \right\rfloor m = 20 - \left\lfloor \frac{20}{4} \right\rfloor 4 = 0$$

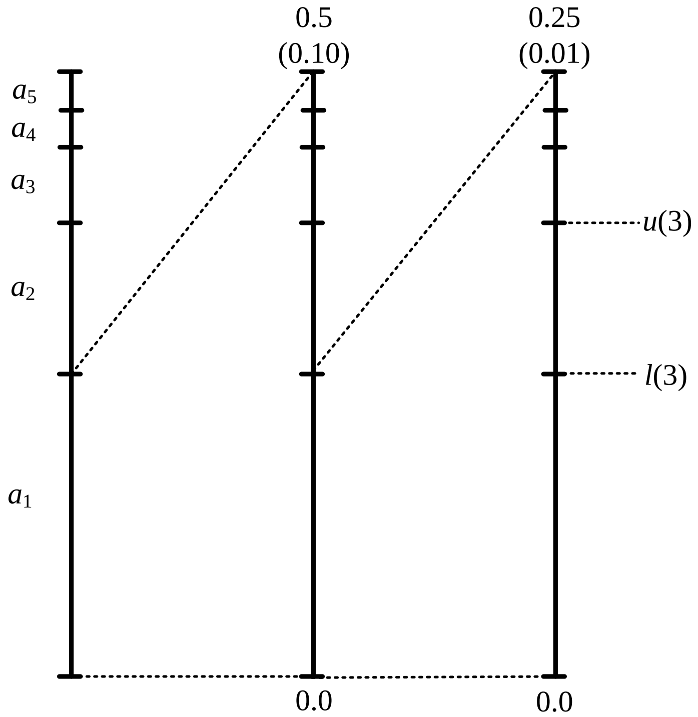
- The Golomb-Rice code for 20 is thus formed from a unary code specifying the group number (in this case 5) concatenated with the 2 bit residual for the group, i.e.: 00000100.

———oooo———

Q7.9 Given the symbols from an alphabet, $A = \{a_1, a_2, a_3, a_4, a_5\}$ and their associated probabilities of occurrence in the table below, determine the shortest arithmetic code which represents the sequence: $\{a_1, a_1, a_2\}$.

Symbol	**Probability**
a_1	0.5
a_2	0.25
a_3	0.125
a_4	0.0625
a_5	0.0625

Outline solution



- The lower and upper limits on the arithmetic coding interval are given by:

$$l\,(i) = l(i-1) + (u(i-1) - l(i-1))P_x\,(x_i - 1)$$
$$u\,(i) = l(i-1) + (u(i-1) - l(i-1))P_x(x_i)$$

- In this case we have:

$$l\,(3) = l(2) + (u(2) - l(2))P_x\,(a_1) = 0.125_{10} = 0.001_2$$
$$u\,(3) = l(2) + (u(2) - l(2))P_x(a_2) = 0.1875_{10} = 0.0011_2$$

- The shortest codeword that will represent this interval uniquely is thus 0.001_2.

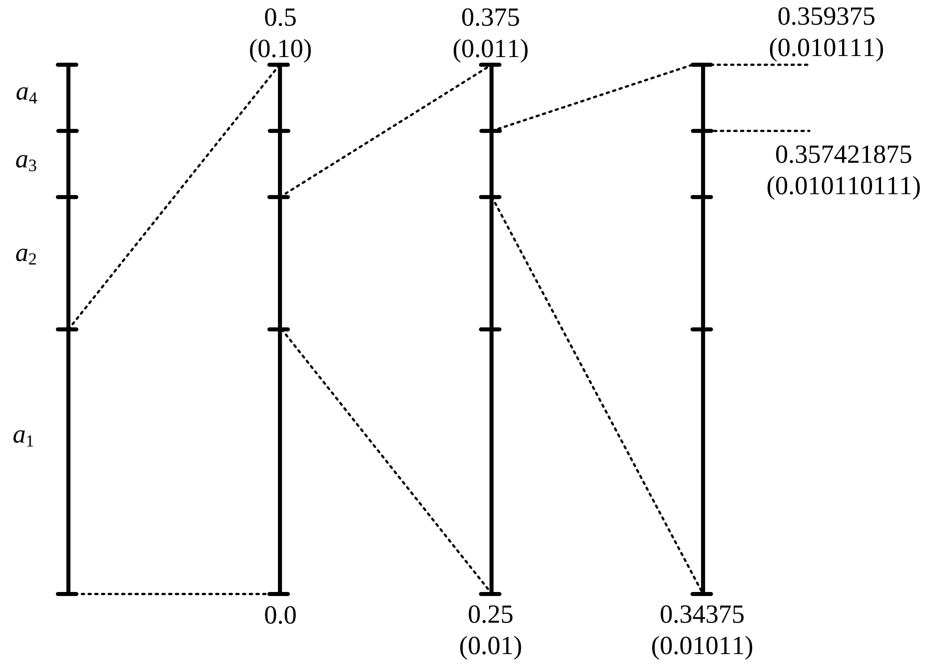
——oooo——

Q7.10 Given the following symbols and their associated probabilities of occurrence, determine the binary arithmetic code which corresponds to the sequence: $\{a_1,\ a_2,\ a_3,\ a_4\}$. Demonstrate how an arithmetic decoder, matched to the encoder, would decode the bitstream: 010110111.

Symbol	**Probability**
a_1	0.5
a_2	0.25
a_3	0.125
a_4(EOB)	0.125

Outline solution

- The arithmetic encoder graph is given below:



- An appropriate codeword for the sequence $\{a_1, a_2, a_3, a_4\}$ would be: 010110111.

- Decoding the bitstream: 010110111: The decoding process is summarised in the following table. As each bit is loaded, it can be processed to further differentiate symbols. When the desired number of symbols or an EOB symbol is received, the decoding terminates.

Rx bit	**Interval**	**Symbol**
0	$[0, 0.5)_{10} = [0, 0.1)_2$	a_1
1	$[0.25, 0.5)_{10} = [0.01, 0.10)_2$	-
0	$[0.25, 0.375)_{10} = [0.010, 0.011)_2$	a_2
1	$[0.3125, 0.375)_{10} = [0.0101, 0.0110)_2$	-
1	$[0.34375, 0.375)_{10} = [0.01011, 0.011)_2$	a_3
0	$[0.59375, 0.609375)_{10} = [0.100110, 0.100111)_2$	-
1	$\cdots$	-
1	$\cdots$	-
1	$[0.357421875, 0.359375)_{10} = [0.010110111, 0.10111)_2$	a_4

———oooo———

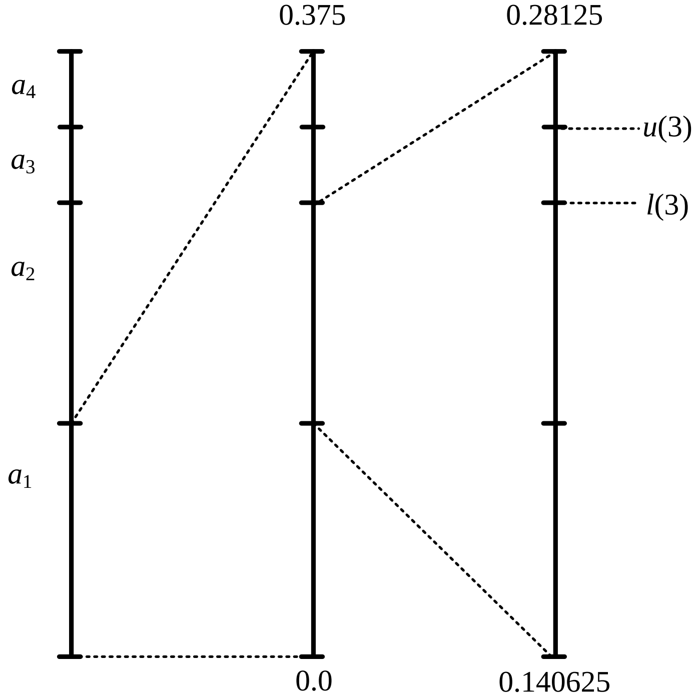
Q7.11 Given the following symbols and their associated probabilities of occurrence, determine the arithmetic code which corresponds to the sequence: a_1, a_2, a_3 (where a_3 represents the EOB symbol).

Symbol	Probability
a_1	0.375
a_2	0.375
a_3	0.125
a_4	0.125

Show how the bitstream produced above would be decoded to produce the original input symbols.

Repeat this question using Huffman encoding rather than arithmetic coding. Compare your results in terms of coding efficiency.

Outline solution



$$l\,(3) = l(2) + (u(2) - l(2))P_x\,(a_2) = 0.24609375_{10} = 0.00111111_2$$
$$u\,(3) = l(2) + (u(2) - l(2))P_x(a_3) = 0.263671875_{10} = 0.010000111_2$$

- The lower limit on the interval (00111111) could be used as the codeword. However a shorter unique codeword exists: 0100000_2. This gives an average of 2.33 bits per symbol for this simple example. However, in the specific case where the decoder can deal with implicit trailing zeros (i.e. no codeword concatenation), and/or the number of symbols transmitted is known (which would normally incur an additional overhead), then the shortest unique codeword for this sequence could be 01_2.

- For the case of Huffman coding, a possible set of codewords (non-unique) are as follows:

Symbol	Prob.	Code
a_1	0.375	1
a_2	0.375	01
a_3	0.125	001
a_4	0.125	000

- For the sequence $a_1,\ a_2,\ a_3$ the transmitted sequence would therefore be: 101001, giving an efficiency of 2 bits per symbol.

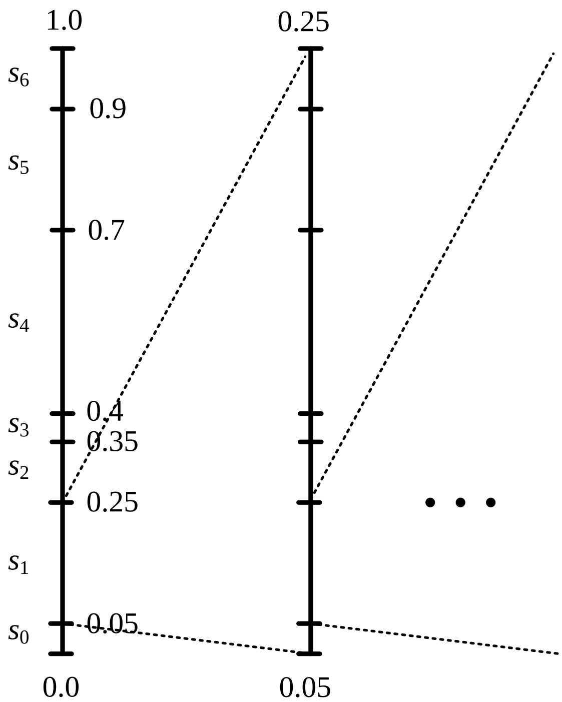
——oooo——

Q7.12 Derive the arithmetic code word for the sequence $s_1,\ s_1,\ s_2,\ s_2,\ s_5,\ s_4,\ s_6$, given a symbol set with the following probabilities:

$$P(s_0) = 0.065;\ P(s_1) = 0.20;\ P(s_2) = 0.10;\ P(s_3) = 0.05;$$
$$P(s_4) = 0.30;\ P(s_5) = 0.20;\ P(s_6) = 0.10 = \text{EOB}$$

Outline solution

- Follow a similar approach to previous solutions in this section. The first stage, corresponding to the first transmitted symbol, is shown below.



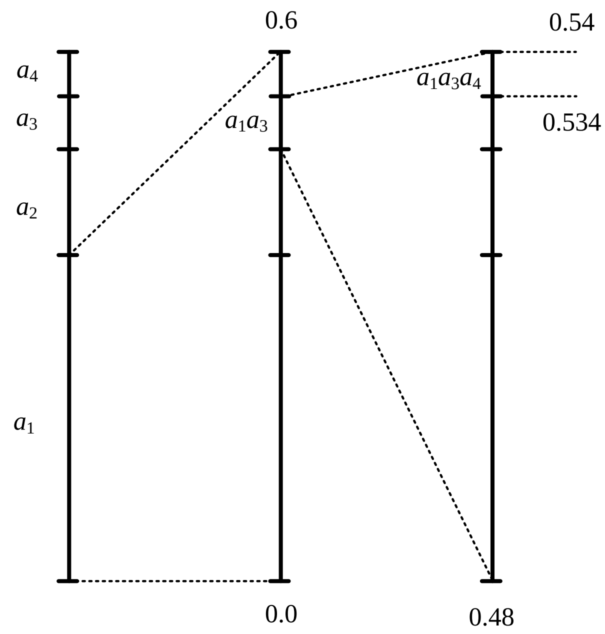
- Following this approach for the complete sequence yields a final arithmetic code interval of $[0.0713336, 0.0713360)$.

——oooo——

Q7.13 Consider an alphabet, $A = \{a_1, a_2, a_3, a_4\}$, where $P(a_1) = 0.6$; $P(a_2) = 0.2$; $P(a_3) = 0.1$; $P(a_4) = 0.1$. Compute the sequence, S, of 3 symbols which corresponds to the arithmetic code 0.58310.

Outline solution

- The arithmetic coding diagram for this problem is given below, where it can easily be verified that the symbol sequence corresponding to the codeword 0.58310 is $\{a_1, a_4, a_2\}$.

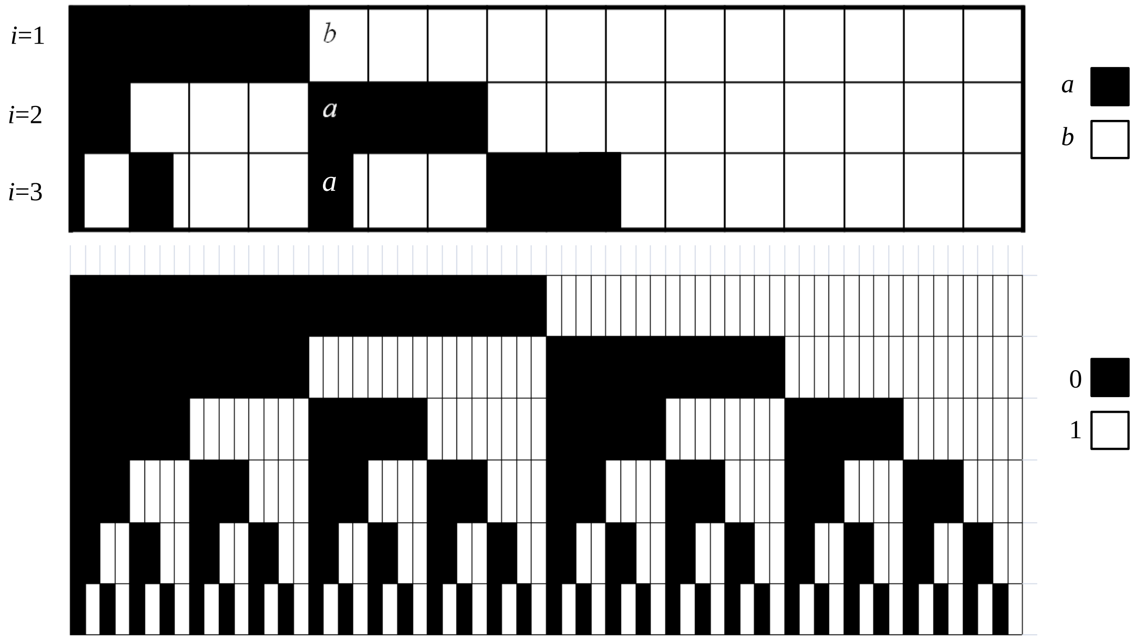


——oooo——

Q7.14 Given an alphabet of 2 symbols A={a,b}, where $P(a) = 0.25$; $P(b) = 0.75$, draw a diagram showing coding and probability intervals for the associated arithmetic coder. Derive a binary arithmetic code for the sequence S=baa.

Outline solution

- The coding and probability intervals for this problem are illustrated below:



- The shortest codeword that uniquely represents the sequence *baa* can be seen to be: 01000.

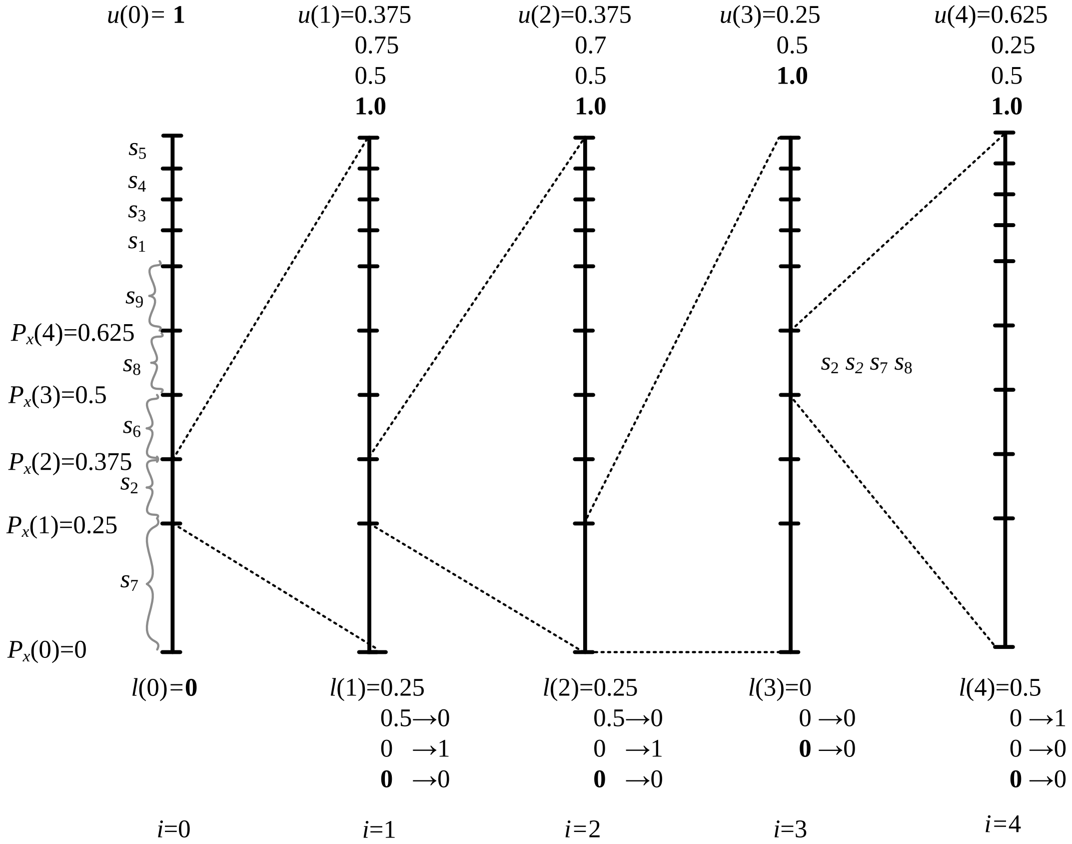
———oooo———

Q7.15 Assuming the symbol probabilities in the table below, compute the arithmetic codeword for the sequence $s2 \quad s2 \quad s7 \quad s8$. Use interval scaling in your solution.

Symbol	**Prob**	**Symbol**	**Prob**
s1{0,1}	0.0625	s5{2,1}	0.0625
s2{0,2}	0.125	s6{2,2}	0.125
s3{1,1}	0.0625	s7{3,1}	0.25
s4{1,2}	0.0625	s8{EOB}	0.125
		s9{Other}	0.125

Outline solution

- An outline of the aithmetic encoder design is shown below:



- Therefore, using the lower limit as the codeword, the sequence has 11 bits: 01001000100

- Note: to guarantee uniqueness of code in general we require:

  Length $= \left\lceil \log_2 \left(\frac{1}{\prod p_i} \right) \right\rceil + 1 = \left\lceil \log_2 \left(\frac{1}{(0.125 \times 0.125 \times 0.125 \times 0.25)} \right) \right\rceil + 1 = 12$bits

  bits

——oooo——

Chapter 8: Coding moving pictures: motion prediction

Q8.1 Implement the full search BBME algorithm on the 6×6 search window, **S**, using the current-frame template, **M**, given below.

$$\mathbf{S} = \begin{bmatrix} 1 & 5 & 4 & 9 & 6 & 1 \\ 6 & 1 & 3 & 8 & 5 & 1 \\ 5 & 7 & 1 & 3 & 4 & 1 \\ 2 & 4 & 1 & 7 & 6 & 1 \\ 2 & 4 & 1 & 7 & 8 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}; \mathbf{M} = \begin{bmatrix} 7 & 7 \\ 7 & 7 \end{bmatrix}$$

Outline solution

- Consider for example, the candidate motion vector **d**=[0,0]. In this case the displaced frame difference signal is given by:

$$\mathbf{E} = \begin{bmatrix} 7 & 7 \\ 7 & 7 \end{bmatrix} - \begin{bmatrix} 1 & 3 \\ 1 & 7 \end{bmatrix} = \begin{bmatrix} 6 & 4 \\ 6 & 0 \end{bmatrix}$$

- Giving an SAD = 16. Similarly evaluating for all other $[d_x, d_y]$ offsets, gives a minimum SAD=2 for the motion vector **d**=[1,1].

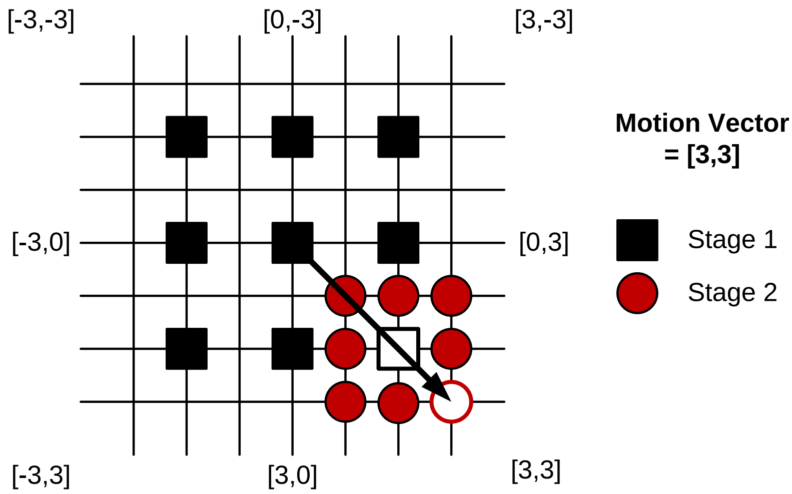
——oooo——

Q8.2 Given the following reference window and current block, show how an N-Step search algorithm would locate the best match (assume a SAD optimisation criterion). Determine the motion vector for this block. Does this produce the same result as an exhaustive search?

$$\begin{bmatrix} 1 & 4 & 3 & 2 & 1 & 2 & 3 & 2 & 2 \\ 0 & 1 & 2 & 0 & 2 & 3 & 0 & 2 & 1 \\ 0 & 2 & 0 & 0 & 1 & 2 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 2 & 1 & 1 & 4 & 4 & 0 & 0 & 1 \\ 0 & 3 & 2 & 3 & 0 & 1 & 2 & 2 & 3 \\ 0 & 3 & 3 & 1 & 0 & 1 & 1 & 2 & 2 \\ 0 & 4 & 2 & 3 & 0 & 2 & 1 & 2 & 1 \\ 0 & 1 & 2 & 1 & 2 & 3 & 2 & 2 & 0 \end{bmatrix} \quad \begin{bmatrix} 1 & 2 & 3 \\ 1 & 2 & 0 \\ 2 & 2 & 0 \end{bmatrix}$$

Outline solution

- The 2 step search algorithm would execute as shown in the following diagram:



- The optimum 1st stage vector is **d**=[2,2] with an SAD=7.The motion vector found after the 2nd stage of the N step search is **d**=[3,3] with and SAD=2.
- In this case the solution is identical to that produced by an exhaustive search, but using 17 search points rather than 49 search points.

——oooo——

Q8.3 Use bidirectional exhaustive search motion estimation, such as that employed in MPEG-2, to produce the best match for the following current block and two reference frames:

$$\begin{bmatrix} 1 & 2 & 3 & 4 \\ 1 & 2 & 2 & 3 \\ 1 & 1 & 2 & 2 \\ 2 & 2 & 2 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 3 & 4 \\ 1 & 1 & 0 & 3 \\ 1 & 1 & 2 & 0 \\ 2 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 1 & 2 \end{bmatrix}$$

Outline solution

- Assuming the conventional motion vector directions, the best forward prediction match is given by **d**=[0, -1, -1] which gives an SAD=0. Note the first index indicates the reference frame selected.
- The best match backwards is SAD=2, given by the following motion vectors: **d**=[1, -1, -1]; **d**=[1, -1, 0]; **d**=[1, 0, 0]; **d**=[1, 0, 1].
- The best motion vector overall is given by the forward prediction, **d**=[0, -1, -1].

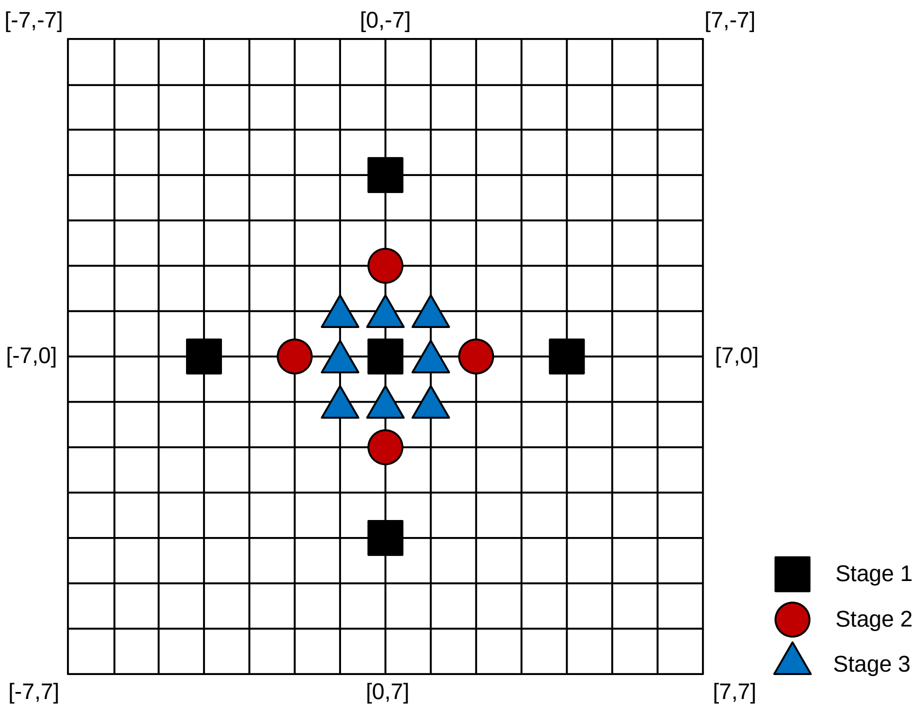
——oooo——

Q8.4 Explain how the Two Dimensional Logarithmic (TDL) search method improves the search speed of block-based motion estimation. Illustrate this method for the case of a $\pm 6 \times \pm 6$ search window where the resultant motion vector is [2,5]. Quantify the savings for this particular example over a full search. What is the main advantage and disadvantage of this method?

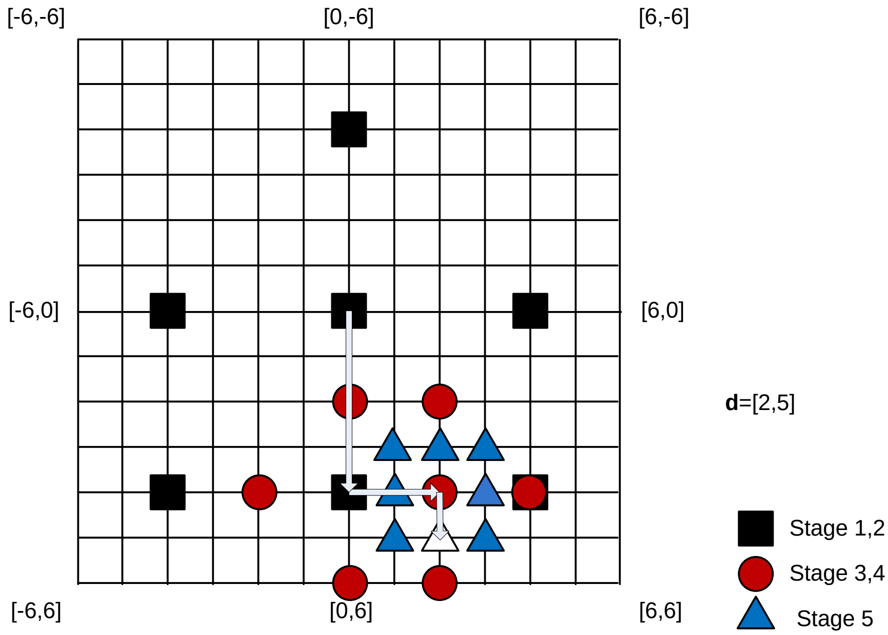
Outline solution

- The TDL search reduces the number of motion vector candidates evaluated. The search patterns used in this method are shown below and form the shape of a $+$ with 5 points, except for the final stage where a 9-point square pattern is checked.

- The largest pattern is initially located at the centre of the search grid, all 5 points are evaluated and the one with the lowest BDM is selected as the centre of the next search. The process is repeated until the minimum point remains at the centre, in which case the search pattern radius is halved. This process continues until the radius is equal to 1, when the 9 point square pattern is invoked.



- The following illustrates the case of a $\pm 6 \times \pm 6$ search window where the resultant motion vector is [2,5].



- Comparing this with a full search, the FS would require 13x13=169 evaluations, whereas the 2DLS requires only 21 evaluations.
- The main advantage of this approach is reduced complexity and the main disadvantage of this method is that it can be susceptible to the influence of local minima in the search space.

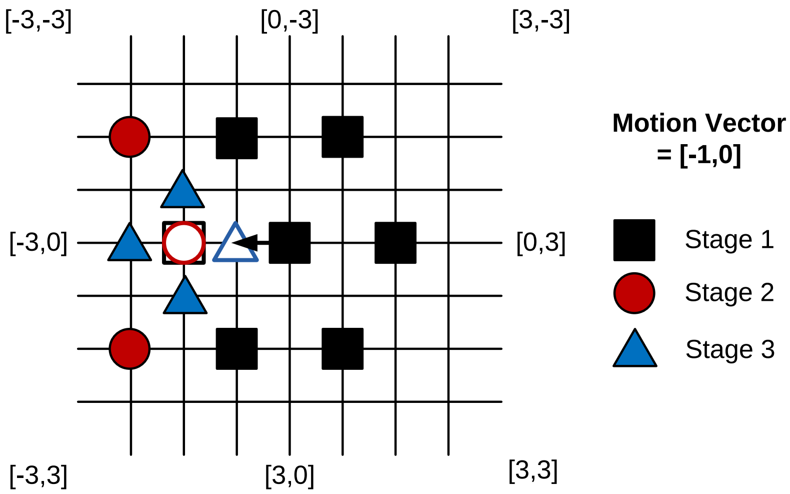
——oooo——

Q8.5 Implement a hexagonal search block matching algorithm on the search window, **S**, using the current-frame template, **M**, as given below. Determine the motion vector for this search. Assume that any search points where the hexagon goes outside of the reference frame are invalid

$$\mathbf{S} = \begin{bmatrix} 0 & 1 & 2 & 3 & 22 & 19 & 18 & 23 \\ 7 & 3 & 6 & 5 & 33 & 31 & 13 & 22 \\ 4 & 6 & 3 & 7 & 23 & 23 & 15 & 26 \\ 8 & 4 & 1 & 3 & 11 & 22 & 29 & 19 \\ 2 & 8 & 9 & 7 & 8 & 14 & 16 & 18 \\ 5 & 0 & 7 & 3 & 7 & 15 & 12 & 13 \\ 7 & 4 & 6 & 6 & 9 & 8 & 8 & 12 \\ 1 & 2 & 3 & 9 & 10 & 9 & 8 & 12 \end{bmatrix} ; \quad \mathbf{M} = \begin{bmatrix} 1 & 4 \\ 10 & 7 \end{bmatrix}$$

Outline solution

- The three steps in the hexagonal search for this problem are illustrated below, giving a motion vector of **d**=[-1, 0].



———oooo———

Q8.6 Given the following current block P and its set of three adjacent neighbours: A, B, C and D with motion vectors as indicated, use motion vector prediction to initialise the search for the best motion vector for this block.

$\mathbf{d}_B$=[1,1]	$\mathbf{d}_C$=[1,0]	$\mathbf{d}_D$=[1,0]	
$\mathbf{d}_A$=[1,2]	P		

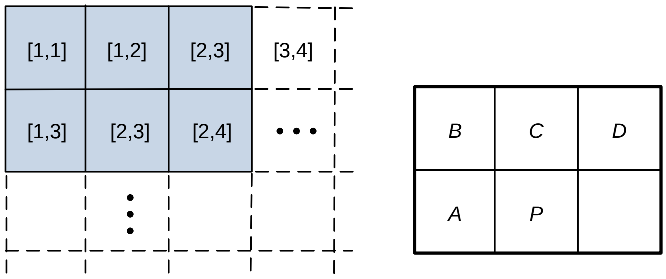
Outline solution

- In this case a typical predictor for the motion vector would be:

$$\hat{\mathbf{d}}_P = \text{med}\left(\mathbf{d}_A, \mathbf{d}_C, \mathbf{d}_D\right) = [1, 0]$$

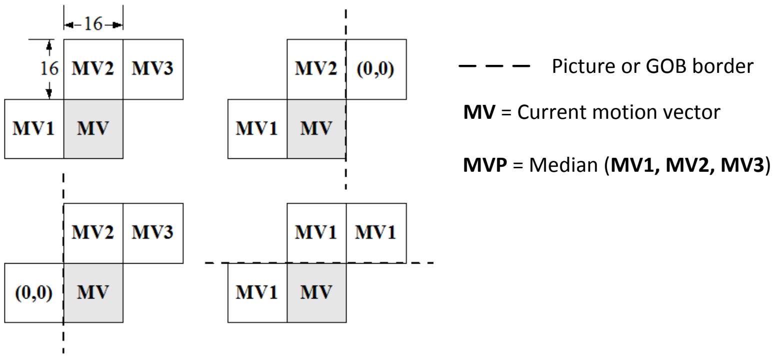
———oooo———

Q8.7 Motion vectors for six adjacent blocks in two partial rows of a frame are shown below. Using the prediction scheme: $\hat{\mathbf{d}}_P = \text{med}(\mathbf{d}_A, \mathbf{d}_C, \mathbf{d}_D)$, compute the predicted motion vectors for each of these blocks together with their coding residuals. State any assumptions that you make regarding the prediction of motion vectors for blocks located at picture boundaries.



Outline solution

- Using the prediction scheme adopted for H.263 and H.264, which is based on the following boundary rules:



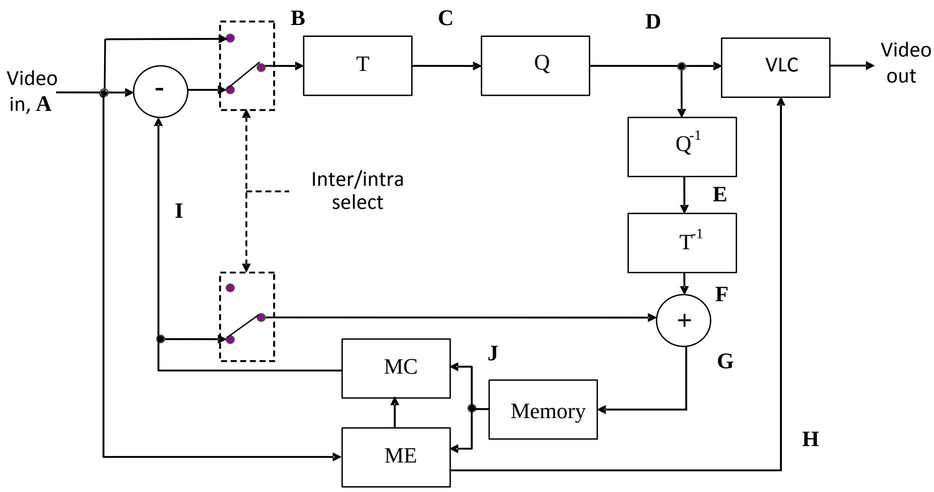
- The predicted motion vectors and residuals for each of the 6 blocks outlined (labelled top left to bottom right) are given in the table below:

Block, P	1	2	3	4	5	6
$\hat{\mathbf{d}}_P$	[0,0]	[1,1]	[1,2]	[1,1]	[1,3]	[2,3]
$\mathbf{e}_P$	[1,1]	[0,1]	[1,1]	[0,2]	[1,0]	[0,1]

——oooo——

Chapter 9: The block-based hybrid video codec

Q9.1 The figure below shows a block diagram of a basic hybrid (block transform, motion compensated) video coding system.



Using the following assumptions and data at time *n:*

Image frame size:	12×12 pixels
Macroblock size:	4×4 pixels
Operating mode:	inter frame
Motion estimation:	Linear translational model (4×4 blocks)

Transform matrix: $\mathbf{T} = \frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix}$

Quantisation matrix: $\mathbf{Q} = \begin{bmatrix} 1 & 1 & 4 & 4 \\ 1 & 2 & 4 & 4 \\ 4 & 4 & 4 & 4 \\ 4 & 4 & 4 & 4 \end{bmatrix}$

Current input block: $\mathbf{A} = \begin{bmatrix} 2 & 2 & 0 & 0 \\ 2 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$; assume exact centre of frame.

Reference memory:

$$\mathbf{J} = \begin{bmatrix} 0 & 0 & 2 & 2 & 2 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 4 & 10 & 4 & 7 & 4 & 7 & 4 & 4 & 4 & 3 & 2 & 1 \\ 8 & 4 & 7 & 6 & 7 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 6 & 4 & 6 & 5 & 4 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 3 & 2 & 10 & 6 & 8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 4 & 14 & 4 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 5 & 6 & 6 & 6 & 6 & 0 & 0 & 7 & 0 & 0 \\ 0 & 0 & 0 & 12 & 0 & 0 & 0 & 3 & 4 & 2 & 3 & 4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & 4 & 5 & 8 & 0 \\ 7 & 7 & 8 & 9 & 0 & 0 & 0 & 0 & 6 & 7 & 0 & 0 \\ 0 & 0 & 0 & 10 & 3 & 0 & 0 & 0 & 0 & 7 & 0 & 0 \\ 0 & 0 & 2 & 2 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Compute:

1. The current motion vector, $\mathbf{H}$, at time n.
2. The motion compensated output, $\mathbf{I}$, at time n.
3. The DFD for the current input frame, $\mathbf{B}$.
4. The transformed and quantised DFD output, $\mathbf{D}$.

Outline solution 1. Current motion vector:

- By inspection:

$$\mathbf{H} = [1, -2]$$

2. Motion compensated output:

$$\mathbf{I} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

3. DFD for current input frame:

$$\mathbf{B} = \mathbf{A} - \mathbf{I} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

4. Transformed and quantised output:

- The transformed output is:

$$\mathbf{C} = \mathbf{TBT}^{\mathrm{T}} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

- and the quantised output (assuming 0.5 values rounded down) is:

$$\mathbf{D} = \mathbf{C}_{(\mathrm{Q})} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

——oooo——

Q9.2 If the encoded bitstream from Q9.1 is decoded by a compliant decoder, compute the decoder output at time n that corresponds to the input block, $\mathbf{A}$.

Outline solution

- The output of the decoder is given by the matrix $\mathbf{G}$ in the figure. Thus:

$$\mathbf{E} = \mathrm{Q}^{-1}(\mathbf{D}) = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{F} = \mathbf{T}^{\mathrm{T}}\mathbf{ET} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{G} = \mathbf{F}{+}\mathbf{I} = \begin{bmatrix} 2 & 2 & 0 & 0 \\ 2 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

- So in this case, a perfect reconstruction of the input block occurs.

——oooo——

Q9.3 Compute the transformed and quantised output for the current input block in Q9.1, but this time for the case of intra frame coding.

Outline solution

- Assuming no intra prediction is used:

$$\mathbf{C} = \mathbf{TAT}^{\mathrm{T}} = \frac{1}{4}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix}\begin{bmatrix} 2 & 2 & 0 & 0 \\ 2 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} = \begin{bmatrix} 2 & 2 & 0 & 0 \\ 2 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

- Assuming the same quantisation matrix is used as for the inter frame case (Note: in practice they are likely to be different):

$$\mathbf{D} = \mathbf{C}_{(\mathrm{Q})} = \begin{bmatrix} 2 & 2 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

——oooo——

Q9.4 Using horizontal, vertical and DC modes only, produce the H.264/AVC intra prediction for the highlighted 4×4 luminance block below:

1	2	3	4	2	3	5	4	7	8	9	5
2	4	4	6	7	8	4	6	4	4	3	2
4	3	5	6	3	3	3	4	4	4	3	3
4	4	5	6	3	3	3	4	4	4	3	3
1	1	4	6	**2**	**3**	**6**	**3**	x	x	x	x
4	3	3	5	**3**	**2**	**5**	**3**	x	x	x	x
2	2	3	5	**3**	**3**	**4**	**4**	x	x	x	x
1	3	5	3	**1**	**2**	**4**	**4**	x	x	x	x

Outline solution

6	3	3	3	4	4	4	3	3
6	**2**	**3**	**6**	**3**				
5	**3**	**2**	**5**	**3**				
5	**3**	**3**	**4**	**4**				
3	**1**	**2**	**4**	**4**				

- DC mode: Mean (3,3,3,4,6,5,5,3)=32/8=4; SAD=18.

4	4	4	4
4	4	4	4
4	4	4	4
4	4	4	4

- Vertical Mode: SAD=14.

3	3	3	4
3	3	3	4
3	3	3	4
3	3	3	4

- Horizontal Mode: SAD=28.

6	6	6	6
5	5	5	5
5	5	5	5
3	3	3	3

- The lowest SAD is given by the vertical mode (SAD=14), so this would be selected as the prediction for this block.

——oooo——

Q9.5 Following the full search result from Q8.1, refine the motion vector for this search to $\frac{1}{2}$ pixel accuracy using the 2- tap interpolation filter: $(s_i + s_j)/2$ (where s_i and s_j are the horizontal or vertical whole-pixel locations adjacent to the selected $\frac{1}{2}$ pixel location).

Outline solution

- From Q8.1, the search window and current frame block are repeated below:

$$\mathbf{S} = \begin{bmatrix} 1 & 5 & 4 & 9 & 6 & 1 \\ 6 & 1 & 3 & 8 & 5 & 1 \\ 5 & 7 & 1 & 3 & 4 & 1 \\ 2 & 4 & 1 & 7 & 6 & 1 \\ 2 & 4 & 1 & 7 & 8 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}; \mathbf{M} = \begin{bmatrix} 7 & 7 \\ 7 & 7 \end{bmatrix}$$

- Evaluating all integer $[d_x, d_y]$ offsets, gave a minimum SAD=2 for the motion vector $\mathbf{d}$=[1,1].

- If we now interpolate the search window values to half pixel scale around the full pixel optimum, applying the interpolation filter $(s_i + s_j)/2$ to the available full pixel values, we obtain the red values in the following matrix:

$$\mathbf{S}_{interp} = \text{rnd} \begin{bmatrix} 1 & 2 & 3 & 3.5 & 4 & 2.5 & 1 \\ 1 & * & 5 & * & 5 & * & 1 \\ 1 & 4 & 7 & 6.5 & 6 & 3.5 & 1 \\ 1 & * & 7 & * & 7 & * & 1 \\ 1 & 4 & 7 & 7.5 & 8 & 4.5 & 1 \\ 1 & * & 4 & * & 4.5 & * & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

- The sub-pixel values marked as * must be interpolated based on the previously interpolated values. It should be noted that this can be a source of drift between encoder and decoder if a consistent approach to interpolation and rounding is not adopted. Here we avoid drift by preserving full precision sub-pixel values (i.e. not rounding) until all values have been computed. Rounding is then applied as the final step. Alternatively rounding could be applied at intermediate steps so long as the processes are identical at both encoder and decoder.

- Filling in the gaps in the above matrix yields the new values in blue below:

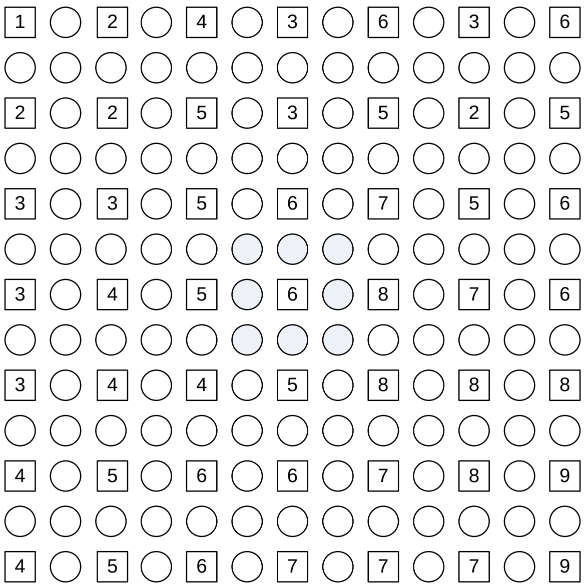
$$\mathbf{S}_{interp} = \text{rnd}\begin{bmatrix} 1 & 2 & 3 & 3.5 & 4 & 2.5 & 1 \\ 1 & 3 & 5 & 5 & 5 & 3 & 1 \\ 1 & 4 & 7 & 6.5 & 6 & 3.5 & 1 \\ 1 & 4 & 7 & 7 & 7 & 4 & 1 \\ 1 & 4 & 7 & 7.5 & 8 & 4.5 & 1 \\ 1 & 2.5 & 4 & 4.25 & 4.5 & 2.75 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 3 & 4 & 4 & 3 & 1 \\ 1 & 3 & 5 & 5 & 5 & 3 & 1 \\ 1 & 4 & 7 & 7 & 6 & 4 & 1 \\ 1 & 4 & 7 & 7 & 7 & 4 & 1 \\ 1 & 4 & 7 & 8 & 8 & 5 & 1 \\ 1 & 3 & 4 & 4 & 5 & 3 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

- Now applying a local refinement search at subpixel level, we see that none of the subpixel offsets provides a lower SAD than the integer solution.

- Hence the resultant motion vector remains as $\mathbf{d}$=[1,1].

——oooo——

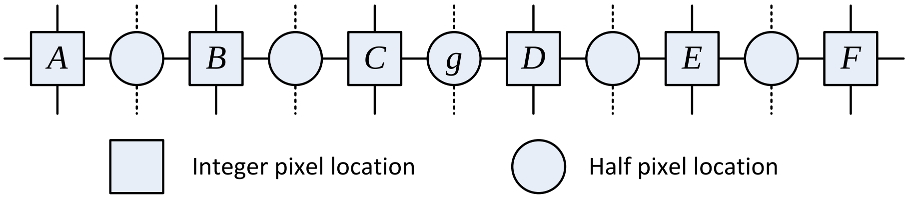
Q9.6 Use the H.264/AVC sub-pixel interpolation filter to generate the half-pixel values for the shaded locations in the following search window:



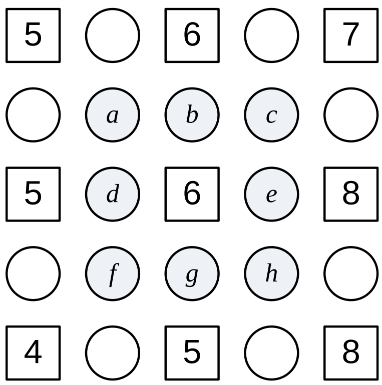
Outline solution

- The H.264/AVC sub-pixel interpolation filter is given by:

$$g = \frac{(A - 5B + 20C + 20D - 5E + F)}{32}$$



- Defining the local 1/2 pixel locations as follows:



- We can compute these sub-pixel values using the above interpolation filter above. For example:

$$b = \text{rnd}\left(\frac{(3 - 5 \times 3 + 20 \times 6 + 20 \times 6 - 5 \times 5 + 6)}{32}\right) = 7$$

$$d = \text{rnd}\left(\frac{3 - 5 \times 4 + 20 \times 5 + 20 \times 6 - 5 \times 8 + 7)}{32}\right) = 5$$

- Computing all intermediate values similarly gives:

$$a = 6;\ b = 7;\ c = 7;\ d = 5;\ e = 7;\ f = 4;\ g = 5;\ h = 7$$

——oooo——

Q9.7 H.264 employs a 4×4 integer transform instead of the 8×8 DCT used in previous coding standards. Prove that the 4 point integer approximation to the 1-D DCT transform matrix, $\mathbf{A}$, is given by:

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix} \otimes \mathbf{E}_f$$

where $\mathbf{E}_f$ is a 4×4 scaling matrix. State any assumptions made during your derivation.

Outline solution

- The 1-D 4×4 DCT is given by:

$$\mathbf{A} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ \sqrt{\frac{1}{2}}\cos(\frac{\pi}{8}) & \sqrt{\frac{1}{2}}\cos(\frac{3\pi}{8}) & -\sqrt{\frac{1}{2}}\cos(\frac{3\pi}{8}) & -\sqrt{\frac{1}{2}}\cos(\frac{\pi}{8}) \\ \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & \frac{1}{2} \\ \sqrt{\frac{1}{2}}\cos(\frac{3\pi}{8}) & -\sqrt{\frac{1}{2}}\cos(\frac{\pi}{8}) & \sqrt{\frac{1}{2}}\cos(\frac{\pi}{8}) & -\sqrt{\frac{1}{2}}\cos(\frac{3\pi}{8}) \end{bmatrix}$$

$$= \begin{bmatrix} a & a & a & a \\ b & c & -c & -b \\ a & -a & -a & a \\ c & -b & b & -c \end{bmatrix} \quad \text{where :} \quad \begin{aligned} a &= \tfrac{1}{2} \\ b &= \sqrt{\tfrac{1}{2}}\cos(\tfrac{\pi}{8}) \\ c &= \sqrt{\tfrac{1}{2}}\cos(\tfrac{3\pi}{8}) \end{aligned}$$

- Letting $d = \frac{c}{b}$, the integer version of the transform can be written as follows:

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & d & -d & -1 \\ 1 & -1 & -1 & 1 \\ d & -1 & 1 & -d \end{bmatrix} \otimes \begin{bmatrix} a & a & a & a \\ b & b & b & b \\ a & a & a & a \\ b & b & b & b \end{bmatrix}$$

- where $\otimes$ represents a scalar multiplication. Since $d \approx 0.414$, the next step is to approximate d by a value of 0.5, to give:

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix} \otimes \begin{bmatrix} a & a & a & a \\ \frac{b}{2} & \frac{b}{2} & \frac{b}{2} & \frac{b}{2} \\ a & a & a & a \\ \frac{b}{2} & \frac{b}{2} & \frac{b}{2} & \frac{b}{2} \end{bmatrix}$$

- Since for all row vectors, $\mathbf{a}_i\mathbf{a}_j^{\mathrm{T}} = 0$, the basis vectors remain orthogonal. To make the matrix orthonormal, we must ensure that $\|\mathbf{a}_i\| = 1$. Hence:

$$a = \frac{1}{2}; \quad b = \sqrt{\frac{2}{5}}; \quad d = \frac{1}{2}$$

- Extending this separable transform to process 2-D signals gives:

$$\begin{aligned} \mathbf{C} &= \mathbf{A}\mathbf{S}\mathbf{A}^{\mathrm{T}} \\ &= \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix} [\mathbf{S}] \begin{bmatrix} 1 & 2 & 1 & 1 \\ 1 & 1 & -1 & -2 \\ 1 & -1 & -1 & 2 \\ 1 & -2 & 1 & -1 \end{bmatrix} \otimes \mathbf{E} \end{aligned} \tag{5}$$

- where:

$$\mathbf{E} = \begin{bmatrix} a^2 & \frac{ab}{2} & a^2 & \frac{ab}{2} \\ \frac{ab}{2} & \frac{b^2}{4} & \frac{ab}{2} & \frac{b^2}{4} \\ a^2 & \frac{ab}{2} & a^2 & \frac{ab}{2} \\ \frac{ab}{2} & \frac{b^2}{4} & \frac{ab}{2} & \frac{b^2}{4} \end{bmatrix}$$

- Thus we can perform the forward and inverse transform operations as: $\mathbf{S} = \mathbf{A}\mathbf{C}\mathbf{A}^{\mathrm{T}}$with the scalar multiplication by $\mathbf{E}$ absorbed into the quantisation process.

———oooo———

Chapter 10: Measuring and managing picture quality

Q10.1 List the primary factors that should be controlled and recorded during subjective video assessment trials.

Outline solution

- Important factors are listed in most standardised testing methodologies. These include:
  - **Display parameters:** size, brightness, dynamic range, resolution.
  - **Viewing environment:** room illumination and ambient lighting, viewing angle, viewing distance:
  - **Coding schemes**: A recording of the codecs employed and their parameters/profiles/levels/configurations.
  - **Content description:** the number of sequences used, sequence duration, sequence content, spatial resolution, temporal resolution, the bit-depth, and the amount of spatial and temporal activity present in the sequences.
  - **Impairments**: the number of impairment types and their descriptions. These typically include compression artefacts obtained from the codecs under test and also possible transmission losses. These should be selected to be consistent with the range of channel conditions prevailing in the application areas of interest.
  - **Subjects**: It is normal to record the number of observers used and their profile, including gender, age, whether they are experts or non-experts and whether they have been screened to ensure that they possess normal visual acuity (with or without corrective lenses).
  - **Testing methodology and statistical analysis methods employed**: For example the duration of each test and a description of the testing methodology used.

——oooo——

Q10.2 List the primary attributes that a good subjective database should possess or be based on.

Outline solution

- A database of subjective opinions on video quality should possess a number of attributes, including:
  - A large number of test sequences in formats appropriate to modern transmission and storage requirements and with a range of content and spatio-temporal activity types.
  - A large number of subjects and hence opinions in order to provide robust statistics.

- Impairments that are typical of modern transmission scenarios including compression artefacts from contemporary codecs at a range of bit rates and losses typical of relevant communication channels.
- They should employ viewing conditions that are appropriate to modern displays and environments.
- Metadata that provides a thorough description of the above conditions and characteristics and of the testing methods adopted.

——oooo——

Q10.3 Given the following three consecutive frames, compute the temporal activity (TI) for this sequence.

$$\mathbf{S}_1 = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 5 \\ 3 & 4 & 5 & 6 \\ 4 & 5 & 6 & 7 \end{bmatrix}; \quad \mathbf{S}_2 = \begin{bmatrix} 1 & 2 & 3 & 5 \\ 2 & 3 & 4 & 6 \\ 3 & 4 & 5 & 7 \\ 4 & 5 & 6 & 8 \end{bmatrix}; \quad \mathbf{S}_3 = \begin{bmatrix} 1 & 2 & 4 & 6 \\ 2 & 3 & 5 & 7 \\ 3 & 4 & 6 & 8 \\ 4 & 5 & 7 & 9 \end{bmatrix}$$

Outline solution

$$\mathrm{TI} = \max_{\forall z} \left\{ \sigma_{\forall(x,y)} \left(\mathbf{S}_z(x,y) - \mathbf{S}_{z-1}(x,y) \right) \right\}$$

- Let:

$$\mathbf{S}'_1 = \mathbf{S}_2 - \mathbf{S}_1 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}; \quad \mathbf{S}'_2 = \mathbf{S}_3 - \mathbf{S}_2 = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

$$\bar{\mathbf{S}}'_1 = 0.25; \quad \bar{\mathbf{S}}'_2 = 0.5$$

$$\sigma_j = \sqrt{\frac{1}{N} \sum_{i=1}^{N} \left(s'_j(i) - \bar{\mathbf{S}}'_j \right)}$$

- Hence:

$$\sigma_{\mathbf{S}'_1} = \sqrt{\frac{1}{16} \left((0.75)^2 \times 4 + (0.25)^2 \times 12 \right)} = \frac{\sqrt{3}}{4}$$

$$\sigma_{\mathbf{S}'_2} = \sqrt{\frac{1}{16} (0.5)^2 \times 16} = \frac{1}{2}$$

- Therefore:

$$\mathrm{TI} = \max_{\forall z} \left\{ \underset{\forall(x,y)}{\sigma} \left(\mathbf{S}_z(x,y) - \mathbf{S}_{z-1}(x,y) \right) \right\} = \frac{1}{2}$$

——oooo——

Q10.4 Consider the operation of a constant bit rate video transmission system at 25fps with the following parameters:

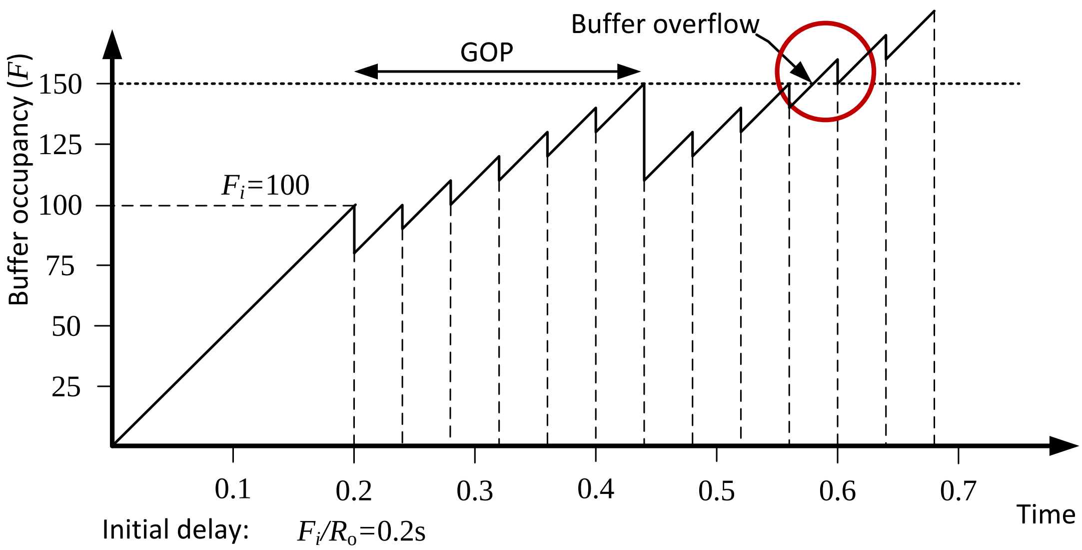
$$R_\mathrm{o} = 500\mathrm{kbps};\ B = 150\mathrm{kb};\ F_i = 100\mathrm{kb};\ \mathrm{GOP} = 6\,\mathrm{frames}$$

If the pictures transmitted have the following sizes, compute the occupancy of the decoder buffer over time and determine whether underflow or overflow occurs.

Frame no.	Picture size (kbits)
1	20
2-6	10
7	40
8-12	10

Outline solution

- The ramp up time for the buffer is $F_i/R_\mathrm{o} = 0.2s$ and one frame period is 0.04s. Plotting the buffer occupancy over the period of these initial pictures, we can observe that the decoder buffer overflows when pictures 11 and 12 are transmitted.



——oooo——

Q10.5 The following Rate (bits) and Distortion (SSD) results are for three mode candidates when encoding a 64×64 Coding Tree Unit (CTU) of an inter predicted frame of the BasketballDrill (843×480, 10 bit) sequence using an HEVC codec. Following the Lagrangian multiplier approach for Rate-Distortion Optimisation, evaluate which of the three modes offers the best RD performance. Assume that the value of the Lagrange parameter λ for mode selection is 232.

Mode	Rate (bits)	SSD
Skip	1	127368
Intra	358	136823
Inter	34	95984

Outline solution

According to the Lagrangian multiplier approach employed for mode selection in RDO, the cost function is given as follows:

$$J_{\text{MODE}} = D + \lambda R,$$

The cost for three candidates modes can thus be calculated as:

$$J_{\text{SKIP}} = D_{\text{SKIP}} + \lambda R_{\text{SKIP}} = 127368 + 232 \times 1 = 127600$$

$$J_{\text{INTRA}} = D_{\text{INTRA}} + \lambda R_{\text{INTRA}} = 136823 + 232 \times 358 = 219879$$

$$J_{\text{INTER}} = D_{\text{INTER}} + \lambda R_{\text{INTER}} = 95984 + 232 \times 34 = 103872$$

Therefore the mode with the lowest cost value among these three is the Inter mode, which offers the best RD performance.

Chapter 11: Communicating pictures: delivery across networks

Q11.1 Using the Huffman codes in the following table, decode the following encoded bitstream: 0 0 1 0 1 0 0 1 1 1 1 1 1 1 0 1 0 0.

Symbol	Probability	Huffman Code
a_1	0.5	0
a_2	0.25	10
a_3	0.125	110
a_4	0.0625	1110
a_5	0.0625	1111

Assuming codeword 110 represents an EOB signal, what would be the effect of a single error in bit position 2 of the above sequence?

Outline solution

- For correct reception and decoding we have:

Rx.	0	0	1	0	1	0	0	1	1	1	1	1	1	1	0	1	0	0
Output	a_1	a_1		a_2		a_2	a_1				a_5				a_4		a_2	a_1

- For the case of the corrupted sequence we have:

Rx.	0	**1**	1	0	1	0	0	1	1	1	1	1	1	1	0	1	0	0
Output	a_1			a_3		a_2	a_1				a_5				a_4		a_2	a_1

- So, 8 symbols are decoded instead of 9. An additional EOB signal is detected after bit position 4 which means that an additional block is decoded. Codeword synchronisation is regained, but the second block will be corrupted and, even if the following EOB (not shown) is decoded correctly, block synchronisation will be lost.

——oooo——

Q11.2 An image encoder uses VLC and Huffman coding of transform coefficients, based on a set of 4 symbols with the following mappings:

$$A \leftrightarrow 0; \qquad B \leftrightarrow 10; \qquad C \leftrightarrow 110; \qquad D \leftrightarrow 111$$

Assuming that the sequence transmitted is $ABCDAC$, determine the received symbol sequences for the following scenarios:

1. An error occurring in the 3rd bit position.
2. An error occurring in the 1st bit position.
3. An error occurring in the 4th bit position.
4. An error occurring in the 8th bit position.

For each scenario, assuming that B represents the EOB symbol, comment on the impact that the error has on the final state of the decoder and on the reconstructed transform coefficients.

Outline solution 1. Correct transmission is ABCDAC... i.e. 0-10-110-111-0-110...
An error occurring in the 3rd bit position:

- Received bits 0-1**1**1-10-111-0-110 would be interpreted as ADBDAC. The correct EOB symbol would be missed so the first block would be corrupted. The second block is also therefore corrupted but the decoder then receives a false EOB and then resynchronises. so block level synchronisation is likely to be preserved.

2. An error occurring in the 1st bit position:

- Received bits **1**10-110-111-0-110 would be interpreted as CCDAC. The correct EOB symbol would be again be missed because AB has been decoded as C. Hence the first block would be corrupted. The second block is also therefore corrupted. The decoder regains symbol level synchronisation but does not receive an EOB symbol so block level synchronisation may be lost.

3. An error occurring in the 4th bit position:

- Received bits 0-10**-0**-10-111-0-110 would be interpreted as ABABDAC. The error means that the first C is decoded as AB so an incorrect EOB signal is decoded. Hence although the decoder regains symbol level synchronisation, an extra block is decoded and block level synchronisation is lost.

4. An error occurring in the 8th bit position:

- Received bits 0-10-110-1**0**-10-110 would be interpreted as ABCBBC. The first block is decoded correctly but DA is decoded as BB, introducing two additional false EOB symbols. Symbol synchronisation is regained but block level synchronisation is lost.

——oooo——

Q11.3 VLC codes for symbols $\{a,\ b,\ c\}$ are: $a = 0$; $b = 11$; $c = 101$, where: $P(a) = 0.5$; $P(b) = 0.25$; $P(c) = 0.25$. Comment on any specific property these codewords exhibit and on its benefits in a lossy transmission environment. Compare the efficiency of these codewords with that of a conventional set of Huffman codes for the same alphabet.

Outline solution

- These codewords are symmetrical and therefore are capable of being decoded reversibly, offering increased error resilience. The average codeword length is 1.75 bits per symbol.
- A set of conventional Huffman codes would be:

$$a = 1;\ b = 01;\ c = 00$$

- These have an average codeword length of 1.5 bits per symbol.

——oooo——

Q11.4 The sequence of VLC codewords produced in Q7.5 is sent over a channel which introduces a single error in the 15th bit transmitted. Comment on the impact that this error has on the reconstructed transform coefficients and any subsequent blocks of data.

Outline solution

- Referring to Q7.5, the symbols produced are:

$$(0,2) \quad (2,1) \quad (0,3) \quad (0,3) \quad (1,1) \quad (\text{EOB})$$

- The corresponding codewords are:

$$(10) \quad (0110) \quad (110) \quad (110) \quad (1111) \quad (01000)$$

- An error in the 15th bit position modifies the sequence as follows:

$$(10) \quad (0110) \quad (110) \quad (110) \quad (1101) \quad (01000)$$

- which would be decoded as:

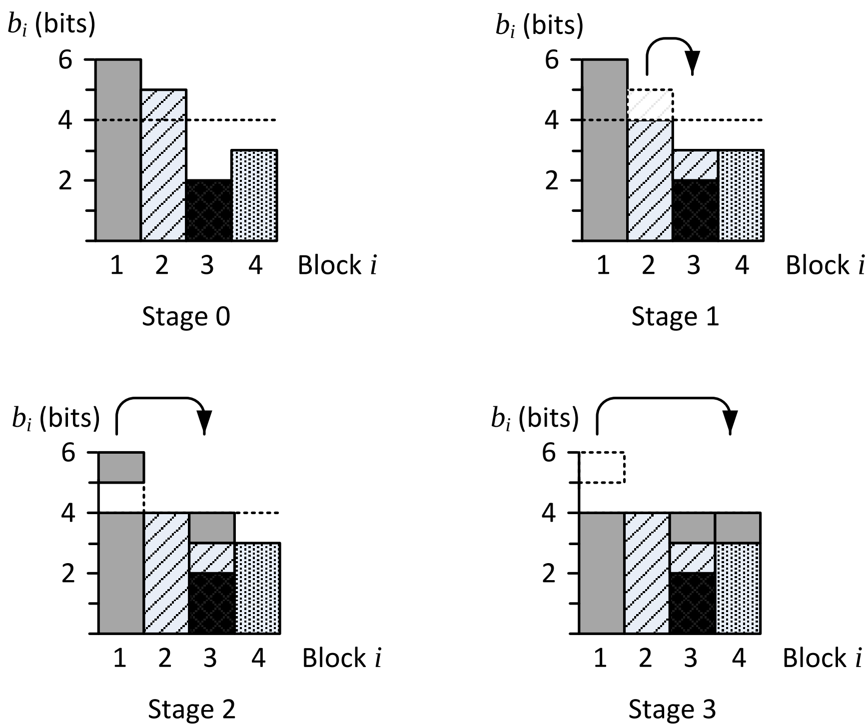
$$(0,2) \quad (2,1) \quad (0,3) \quad (0,3) \quad (0,3) \quad (0,2) \quad (0,2) \quad (0,1)$$

- An increased number of (incorrect) symbols are decoded, the EOB symbol is missed and the errors will propagate to the subsequent block or blocks of data.

——oooo——

Q11.5 Use EREC to code four blocks of data with lengths: $b_1 = 6$, $b_2 = 5$; $b_3 = 2$; $b_4 = 3$. Assuming that an error occurs in the middle of block 3 and that this causes the EOB code to be missed for this block, state which blocks in the frame are corrupted and to what extent.

Outline solution



- An error in block 3 will propagate through that block due to loss of symbol synchronisation. Block 3 will therefore be corrupted. Depending on the resynchronisation characteristics prevailing, the EOB symbol for block 3 may be missed. If so, then other data in slot 3 will also be corrupted, i.e. the higher frequency components of block 2 and possibly block 1.
- Error propagation will terminate at the top of slot 3 which creates an implicit synchronisation point, so no other blocks will be affected.

——oooo——

Q11.6 A video coder uses VLC and Huffman coding based on a set of 4 symbols with probabilities 0.4, 0.3, 0.15 and 0.15. Calculate the average bit length of the resulting Huffman codewords. If this video coder is to employ reversible codes (RVLC) to improve error resilience, suggest appropriate code words and calculate the bit rate overhead compared to conventional Huffman coding.

Outline solution

- It is straightforward to obtain a set of Huffman codewords for these symbols. For example:

$$a_0 = 11;\ a_1 = 10;\ a_2 = 01;\ a_3 = 00$$

- The average codeword length is then 2 bits.
- If RVLC codewords were employed then these could,for example, be:

$$a_0 = 00;\ a_1 = 11;\ a_2 = 101;\ a_3 = 010$$

- The average length then becomes 2.3bits per symbol.

——oooo——

Q11.7 Assume that $\mathbf{S_1}$ and $\mathbf{S_2}$ below represent corresponding regions in 2 temporally adjacent video frames. Due to transmission errors, the central 4×4 block (pixels marked as '×') in the received current frame, $\mathbf{S_2'}$ is lost. Assuming that the codec operates on the basis of 2×2 macroblocks:

a) Perform temporal error concealment using frame copying to provide an estimate of the lost block.
b) Perform motion compensated temporal error concealment, based on the BME measure. Assume that the candidate motion vectors obtained from the four adjacent blocks are either [0,1], [1,0] or [1,1].

$$\mathbf{S_1} = \begin{bmatrix} 1 & 1 & 1 & 2 & 2 & 2 & 3 & 3 \\ 2 & 2 & 2 & 3 & 3 & 3 & 4 & 4 \\ 3 & 3 & 3 & 4 & 4 & 4 & 4 & 4 \\ 3 & 3 & 3 & 4 & 4 & 4 & 5 & 5 \\ 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 \\ 5 & 5 & 5 & 6 & 6 & 7 & 6 & 6 \\ 6 & 6 & 6 & 7 & 6 & 6 & 6 & 6 \\ 7 & 7 & 7 & 7 & 8 & 8 & 8 & 8 \end{bmatrix}; \mathbf{S_2'} = \begin{bmatrix} 2 & 2 & 3 & 3 & 3 & 4 & 4 & 3 \\ 3 & 3 & 4 & 4 & 4 & 4 & 5 & 5 \\ 3 & 3 & 4 & 4 & 4 & 4 & 4 & 4 \\ 4 & 4 & 5 & \times & \times & 4 & 4 & 6 \\ 5 & 5 & 6 & \times & \times & 6 & 6 & 6 \\ 7 & 6 & 7 & 6 & 6 & 6 & 6 & 7 \\ 6 & 7 & 7 & 6 & 6 & 8 & 8 & 8 \\ 9 & 9 & 9 & 7 & 8 & 7 & 8 & 8 \end{bmatrix}$$

Outline solution (a) Frame copying error concealment:

- In this case the co-located block in the previous fame is used for concealment; i.e. $\begin{bmatrix} 4 & 4 \\ 4 & 4 \end{bmatrix}$, giving a BME of 9.

(b) Motion compensated temporal error concealment:

- For motion vector candidate [0,1], BME=1.
- For motion vector candidate [1,0], BME=9.
- For motion vector candidate [1,1], BME=2.
- On this basis, the motion vector [0,1] would be used and the block selected for concealment is: $\begin{bmatrix} 4 & 4 \\ 6 & 6 \end{bmatrix}$

——oooo——

Q11.8 Assuming again that the central 2×2 block (pixels marked as 'x') in the received current frame, $\mathbf{S}_2'$ (above) is lost, calculate the missing elements using spatial error concealment.

Outline solution

- The interpolation is performed using the following equation:

$$s(x,y) = \frac{d_L s_R(x,y) + d_R s_L(x,y) + d_T s_B(x,y) + d_B s_T(x,y)}{d_L + d_R + d_T + d_B}$$

- In this case this gives for the top left missing value (assume 0.5 is rounded up):

$$s(0,0) = \mathrm{rnd}\left(\frac{1\times 4 + 2\times 5 + 1\times 6 + 2\times 4}{6}\right) = 5$$

- Similarly, calculating for the other missing values we have:

$$\mathbf{S} = \begin{bmatrix} 5 & 5 \\ 6 & 6 \end{bmatrix}$$

——oooo——

Q11.9 A block-based image coder generates a slice comprising 5 DCT blocks of data as follows:

Block Number	Block Data
1	01011011101111
2	101110010111101111
3	1100101111
4	101101101111
5	0001111

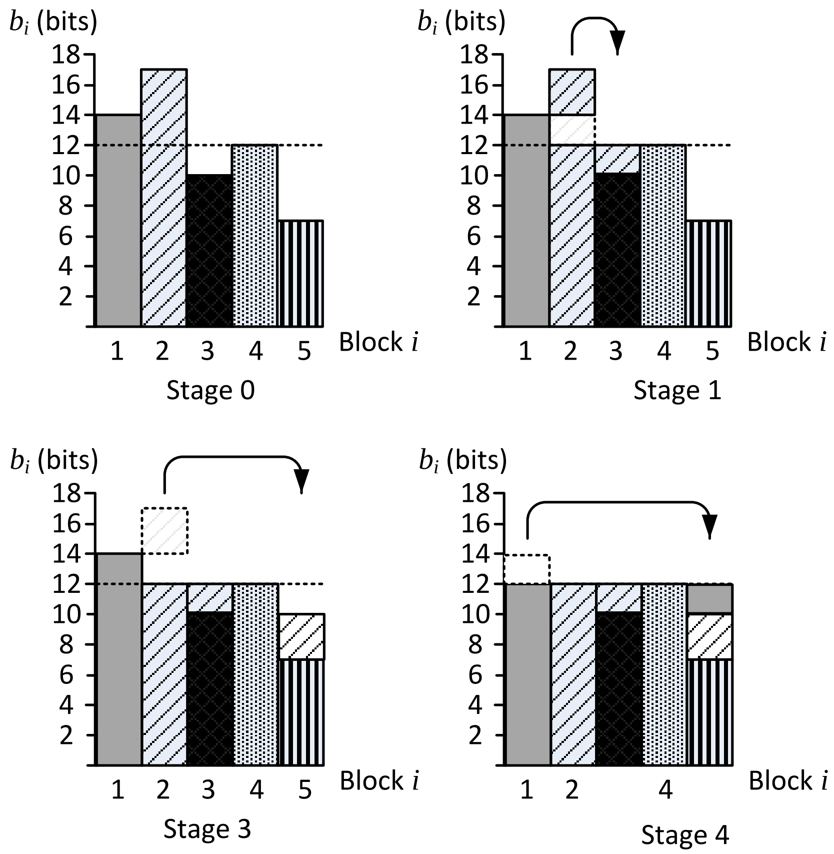
where the symbols and entropy codes used to generate this data are as follows (assume E is the End of Block Symbol):

Symbol	Huffman Code
A	0
B	10
C	110
D	1110
E	1111

Code this data using Error Resilient Entropy Coding (EREC). Choose an appropriate EREC frame size and offset sequence for this slice and show all the encoding stages of the algorithm.

Outline solution

- The block lengths are: $l(b_1) = 14$; $l(b_2) = 17$; $l(b_3) = 10$; $l(b_4) = 12\, l(b_5) = 7$
- The slot length should therefore be: $\left\lceil \frac{1}{N} \sum l(b_i) \right\rceil = 12$
- The slot packing solution thus proceeds as follows:



———oooo———

Q11.10 Perform EREC decoding on the bitstream generated from Q11.9.
If the slice data given in the above question is corrupted during transmission such that the values of the last 2 bits in each of block 2 and block 3 are complemented, how many blocks are corrupted after EREC decoding. Compute how many blocks would contain errors if the slice were transmitted as a conventional stream of entropy codewords without EREC.

Outline solution

- EREC decoding proceeds according to the table below (X=no action, P_i=partial decode from slot i, C_i=complete decode from slot i (EOB detected):

Stage	**Offset**	**bk1**	**bk2**	**bk3**	**bk4**	**bk5**
0	0	**X**	**X**	$\mathbf{C_3}$	$\mathbf{C_4}$	$\mathbf{C_5}$
1	1	**X**	$\mathbf{P_3}$			
2	2	**X**	**X**			
3	3	**X**	$\mathbf{C_5}$			
4	4	$\mathbf{C_5}$				

- The stages are described as follows:

  - At stage 0, we start decoding from the base of the 1st block and proceed until the end of the slot is reached or an EOB symbol is decoded. In the case of block 1, we reach the end of the slot with no EOB detected. We do the same for all other slots. Hence blocks 3, 4 and 5 are all fully decoded at stage 0.
  - At stage 1 we search (for all incomplete blocks) for additional data at an offset of 1. For example, in the case of block 2 we search slot 3: we detect additional data beyond the EOB for block 3, but still no EOB is detected for block 2 so it remains incomplete.
  - At stage 3 block 2 is fully decoded from slot 5 at an offset of 3.
  - Finally at stage 4, block 1 is fully decoded from slot 5 at an offset of 4.

- If the last 2 bits in each of block 2 and block 3 are complemented, then the EOB symbol E=1111 for blocks 2 and 3 will be interpreted as C=110 followed by A=0. Hence the correct EOB signals for these blocks will be missed. Hence, during EREC decoding:

  - At stage 0 blocks 4 and 5 will be correctly decoded.
  - At stage 1 no further decoding can take place
  - At stage 2, the incomplete block 3 incorrectly decodes the remainder of slot 5, introducing errors in the high frequency components of block 3.
  - There is then no other data to decode, so blocks 1 and 2 remain incomplete. Their low frequency components are however intact, so if all the remaining coefficients are replaced with zeros, then errors will only occur in their higher frequency components.
  - In summary, no loss of block synchronisation occurs but 3 blocks exhibit some errors in their high frequency coefficients.

Stage	**Offset**	**bk1**	**bk2**	**bk3**	**bk4**	**bk5**
0	0	**X**	**X**	**X**	$\mathbf{C_4}$	$\mathbf{C_5}$
1	1	**X**	**X**	**X**		
2	2	**X**	**X**	C_5		
3	3	**X**	**X**			
4	4	**X**	**X**			

- If the slice were transmitted as a conventional stream of entropy codewords without EREC, the following symbols would be decoded.

bk1	**bk2**	**bk3**	**bk4**	**bk5**
ABCDE	BDABDCA	CABCA	BCCE	AAAE
Correct	EOB missed	EOB missed	EOB detected but blocks 2,3 and 4 decoded as a single block	Correct but spatially shifted by two block positions

——oooo——

Chapter 12: Video coding standards

Q12.1 Assuming a CIF format luminance-only H.261 sequence with synchronisation codewords at the end of each GOB, calculate the (likely) percentage of corrupted blocks in a frame if bit errors occur in block 1 of macroblock 7 in GOB 2 and block 1 of macroblock 30 of GOB 10.

Outline solution

Arrangement of GOBs in a CIF frame:

1	2
3	4
5	6
7	8
9	10
11	12

GOB structure:

1	2	3	4	5	6	7	8	9	10	11
12	4	14	15	16	17	18	19	20	21	22
23	24	25	26	27	28	29	30	31	32	33

Macroblock format:
(Y only)

1	2
3	4

- Assuming that synchronisation codewords are employed at the end of each GOB.
  - The error in block 1 of MB7 of GOB2: Assuming that the error propagates to the end of GOB2 (worst case), this will affect 27 macroblocks, or 108 blocks.
  - The error in block 1 of MB30 of GOB10: Assuming that the error propagates to the end of GOB10, this will affect 4 macroblocks or 16 blocks.
  - The total likely percentage of corrupted blocks is therefore:

$$\frac{124}{12 \times 33 \times 4} \times 100 = 8\%$$

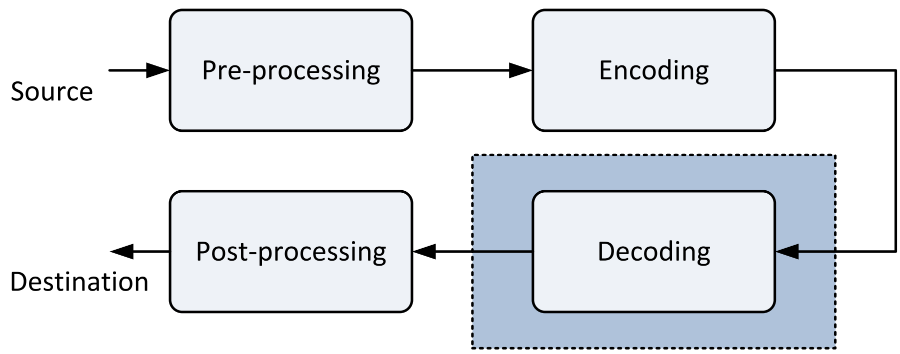
———oooo———

Q12.2 What parts of a video codec are normally subject to standardisation in term of compliance testing? Why is this?

Outline solution

- Video coding standards conventionally define the bitstream format and syntax and the decoding process, not (for the most part) the encoding process. This is illustrated in the figure below, where the dashed box indicates the normative aspects of the standard. A standard-compliant encoder is thus one that produces a compliant bitstream and a standard-compliant decoder is one that can decode a compliant bitstream. Standard compliance of an encoder thus provides no guarantee of quality.

- This approach supports straightforward compliance testing while enabling manufacturers to differentiate their products through innovative low complexity, robust and efficient coding solutions.



———oooo———

Q12.3 Discuss the concepts of profiles and levels in MPEG-2 indicating how these have enabled a family of compatible algorithms covering a range of applications and bitrates to be defined.

Outline solution

- To match performance against decoder capability or capacity, MPEG-2 introduced an extensive range of *profiles* and *levels*. A profile is a defined subset of the entire bitstream syntax and profiles are further partitioned into levels. Each level specifies a range of allowable values for the parameters in the bitstream.

- MPEG-2 supports six profiles – Simple, Main, SNR, Spatial, High 4:2:2 and Multiview. Provisions for scalability are included in the SNR, Spatial and High profiles whereas the Simple Profile and Main Profile allow only single-layer coding. It also offers four possible levels – Low, Main, High1440 and High - in each profile. The parameters for the Main Level approximately correspond to normal TV resolution, the Low Level corresponds to CIF resolution, and the values for High1440 and High correspond to HDTV resolution.

———oooo———

Q12.4 Compare the primary features of H.264/AVC and MPEG-2 and highlight those features that have enabled H.264 to approximately halve the bitrate of MPEG-2 for equivalent picture quality.

Outline solution

- Some of the key differentiating features between these two codecs are given in the following table.

Codec feature	MPEG-4/AVC	MPEG-2
Formats supported	QCIF to UHDTV (4k).	SDTV to HDTV
Frame types	I, P, B, SP, SI, Hierarchical B frames. (MB skip mode). B frames can be used as a reference.	I, P, B (MB skip mode).
DC coding	Hadamard transform.	Differential
Inter-coding transform	Block 8×8 or 4×4 integer transform.	Block 8×8 DCT.
Intra-prediction	Multi-direction, multi-pattern 9 directions.	DC only.
Motion compensation	Quarter pixel, unrestricted vectors, multiple reference frames.	Half pixel, maximum of 1 past reference for P frames and 2 references for B frames.
ME block size	16×16, 16×8, 8×16, 8×8, 8×4, 4×8, 4×4	16×16
Entropy coding	CAVLC or CABAC	Conventional VLC.
Loop filter	In loop non-linear deblocking filter.	None.
Error resilience	Slices, FMO, SI, SP frames, multiple reference frames. Network independence.	Basic slices. External error resilience features and channel coding specified in DVB standards.

——oooo——

Q12.5 Compare the features of H.265/HEVC and H.264/AVC and highlight those features that have enabled HEVC to approximately halve the bitrate of H.264 for equivalent picture quality.

Outline solution

- Some of the key differentiating features between these two codecs are given in the following table.

Codec feature	**MPEG-4/AVC**	**HEVC**
Formats supported	QCIF to UHDTV (4k) typically to 60fps.	Formats from QVGA to UHDTV (8k) up to 300fps.
Transform	Block 8×8 or 4×4 integer transform	4×4, 8×8, 16×16 and 32×32. Transform basis functions are derived from the DCT. Smaller transform sizes are obtained by sub-sampling the larger 32×32 matrix. In the case of intra coded 4×4 residuals, an alternative integer transform is used - derived from the DST.
Intra-prediction	Multi-direction- 9 modes, multi-pattern	Multi-direction- 35 modes, multi-pattern
Motion compensation	Quarter pixel, unrestricted vectors, multiple reference frames. 6 tap interpolation filter for half pixel and bilinear for quarter pixel.	Quarter pixel, unrestricted vectors, multiple reference frames. Half-pixel interpolation uses an 8 tap filter and a 7 tap filter is used for quarter-sample interpolation. These are applied separably and (unlike H.264) without intermediate rounding.
ME block size	16×16, 16×8, 8×16, 8×8, 8×4, 4×8, 4×4	Flexible block partitioning based on CTU structure with maximum size 64×64 samples
MV prediction	MV prediction based on local MV rank order statistics.	An Advanced Motion Vector Prediction mode (AMVP) is used to derive likely motion vector candidates. Improved skip modes and MV merging capabilities between adjacent blocks.
Entropy coding	CAVLC or CABAC	CABAC only with improved context modelling.
Loop filter	In loop non-linear deblocking filter.	Loop filter but with added sample offset adjustment to reduce banding and ringing artefacts.
Architecture	—	Increased exploitation of parallelisation for efficient hardware implementation.

——oooo——

Q12.6 Why has H.265/HEVC increased its maximum block size to 64×64 pixels and VVC increased this further to 128×128?

Outline solution

- HEVC has increased its maximum block size to 64 by 64 pixels in order to cope better with the spatio-temporal characteristics of new formats such as UHDTV with 4K and 8K spatial sampling at frame rates up to 120Hz.

- VVC has further increased this to support further RD performance gains but to save complexity - has limited ghe number of non zero coefficients in its transform matrix.

- The block partitioning approach, using quad-trees, also provides more flexibility in representing complex spatio-temporal patterns.

——oooo——

Q12.7 Compare the features of H.265/HEVC and H.266/VVC and highlight those features that have enabled HEVC to approximately halve the bitrate of H.264 for equivalent picture quality.

Outline solution

Codec feature	HEVC	**VVC**
Formats supported	Formats from SQCIF to UHDTV (8k) up to 300fps.	Format from SQCIF to UHDTV (8k) up to 300fps
Transform	4×4, 8×8, 16×16 and 32×32. Transform basis functions are derived from the DCT. Smaller transform sizes are obtained by sub-sampling the larger 32×32 matrix. In the case of intra coded 4×4 residuals, an alternative integer transform is used - derived from the DST.	Multiple primary transform selections are adopted with new types of DCT and DST.
Intra-prediction	Multi-direction- 35 modes, multi-pattern	65 directional intra modes, plus surface fitting and DC prediction. Mode dependent intra smoothing. Multi-reference line intra prediction
Motion compensation	Quarter pixel, unrestricted vectors, multiple reference frames. Half-pixel interpolation uses an 8 tap filter and a 7 tap filter is used for quarter-sample interpolation. These are applied separably and (unlike H.264) without intermediate rounding.	$\frac{1}{16}$ pixel motion vector precision
ME block size	Flexible block partitioning based on CTU structure with maximum size 64×64 samples	Maximum CTU size for luma block is 128×128. A quadtree structure with nested multi-type partitions using binary and ternary segmentation
MV prediction	An Advanced Motion Vector Prediction mode (AMVP) is used to derive likely motion vector candidates. Improved skip modes and MV merging capabilities between adjacent blocks.	Affine motion model based inter prediction. Combined intra and inter prediction. Adaptive motion vector resolution
Entropy coding	CABAC only with improved context modelling.	Dependent quantization with max QP increased to 63. CABAC engine with an adaptive double probability update model.
Loop filter	Loop filter but with added sample offset adjustment to reduce banding and ringing artefacts.	New adaptive loop filter. Luma mapping with chroma scaling.
Architecture	Increased exploitation of parallelisation for efficient hardware implementation.	

Chapter 13: Communicating pictures: the future

Q13.1 What are the primary challenges and demands for video compression in the future?

Outline solution

- The primary demands relate to the need to communicate increasing amounts of video content while satisfying increased quality demand from users.
- This is especially the case with emerging, more immersive formats with higher spatial resolution, higher temporal resolution, higher dynamic range and multiple views.

——oooo——

Q13.2 Consider the case of an 8K resolution video signal in 4:2:2 format with 14 bits dynamic range and a frame rate of 300fps. Calculate the total bit rate needed for this video.

Outline solution

- An 8k resolution is conventionally interpreted as 7680×4320 pixels. Assuming a frame rate of 300fps and a dynamic range of 14 bits, in 4:2:2 format (2 samples per pixel), we can calculate the (uncompressed) throughput requirement as follows:

$$7680 \times 4320 \times 2 \times 14 \times 300 = 2.786918 \times 10^{11} \approx 279\text{Gbps}$$

——oooo——

Q13.3 How might video compression algorithms develop in the future in order to cope with the demands of formats such as that described in Q13.2?

Outline solution

- Perception-based compression methods such as Parametric Video Coding (Sec 13.4) use an analysis/synthesis framework rather than the conventional energy minimisation approach. These have demonstrated the potential for significant savings compared to block-based methods.
- Increased exploitation of context can provide a basis for effectively exploiting a priori knowledge about the spatio-temporal characteristics of a scene.

——oooo——