CARDIFF UNIVERSITY
EXAMINATION PAPER

Academic Year: 2008/2009

Examination Period: Autumn

Examination Paper Number: CM0340 SOLNS

Examination Paper Title: Multimedia
SOLUTIONS

Duration: 2 hours

Do not turn this page over until instructed to do so by the Senior Invigilator.

Structure of Examination Paper:

There are 14 pages.
There are 4 questions in total.
There are no appendices.
The mark obtainable for a question or part of a question is shown in brackets alongside the question.

Students to be provided with:

The following items of stationery are to be provided:
ONE answer book.

Instructions to Students:

Answer 3 questions.
The use of translation dictionaries between English or Welsh and a foreign language bearing an appropriate departmental stamp is permitted in this examination.

Q1. (a) *Briefly outline the* **four** *broad classes of approach that one may exploit to compress multimedia data.* **Do not detail any specific compression algorithms**.

Compression basically employs redundancy in the data:

Temporal — in 1D data, 1D signals, Audio etc. correlation between sequential data points. [2]

Spatial — correlation between neighbouring pixels or data items in 2D. [2]

Spectral — This uses the frequency domain to exploit relationships between frequency of change in data. E.g. in video/imagery, correlation between colour or luminescence components. [2]

Psycho-visual — exploit perceptual properties of the human auditory/visual system. [2]

8 MARKS — BOOKWORK

Give one example of a compression algorithm for each class.

EXAMPLES:

Temporal — Any Audio/Video compression method, Zero length suppression, pattern substitution, Pulse Code Modulation (a few variants), MPEG Audio, MPEG Video, H.264 [1]

Spatial — Any Image/Video compression algorithm, GIF, JPEG, MPEG, H.264. [1]

Spectral — JPEG, MPEG, H.264. [1]

Psycho-visual — MPEG audio, MPEG Video, JPEG (colour conversion). [1]

4 MARKS — BOOKWORK

(b) *What* advantage *does arithmetic coding offer over Huffman coding for data compression?*

Huffman coding assumes an integer number (k) of bits for each symbol hence k is never less than 1 [1]

Arithmetic coding can represent fractional number of bits and can achieve better compression ratios. [1]

2 MARKS — BOOKWORK

(c) *Briefly state an algorithm for arithmetic decoding*.

SEEN In LECTURE Coding:

The idea behind arithmetic coding is

- To have a probability line, 0–1, and
- Assign to every symbol a range in this line based on its probability,

- Order in terms of probability highest first.
- Note: The higher the probability, the higher range which assigns to it.

For each symbol in the sequence assign a code based in symbols probability and then subdivide for all the symbols:

```
range = high - low;
high = low + range * high_range of the symbol being coded;
low = low + range * low_range of the symbol being coded;
```

Decoding is the opposite so need to work out (**unseen in lectures**)

For current code value:

- look up in table and assign symbol [1]
- Eliminate symbol effect by subtracting the `low` value in the range and divide by `range` [2]
- Repeat above two steps until zero reached — see last part of problem. [2]

Total 5 marks — Unseen. (Coding algorithm discussed in lectures, decoding simply mentioned as the reverse process)

(d) *Given the following table of frequency counts, probabilities and probability ranges for the following characters:*

Char	Freq	Prob.	Range
`A`	2	0.5	[0.0, 0.5)
`B`	1	0.25	[0.5, 0.75)
`C`	1	0.25	[0.75, 1.0)

What is the 4 character sequence *for the arithmetic coding: 0.59375?*

Char	Code-Low	Range
B	0. 0.59375 - 0.5 = 0.09375	0.09375/0.25 = 0.375
A	0.375 - 0.0 = 0.375	0.375/0.5 = 0.75
C	0.75 - 0.75 = 0.0	0.0/0.25 = 0.0
A	0.0 - 0.0 = 0.0	0.0/0.5 = 0.0

4 marks for each step in computation of the code

It is possible for the decoder to return a zero value which corresponds to the symbol in a probability range rather than the end of the decoding process. How can this problem be avoided in the arithmetic decoder?

As can be seen in the above decoding it third step decoding to C returns 0 but we need A also 0 as last step.

Solution: Need some end of input (end-of-file) additional symbol.

4 Marks

TOTAL 8 MARKS — UNSEEN. Coding algorithm discussed in lectures, decoding simply mentioned as the reverse process.

Q2. (a) *In a digital signal processing system, what are meant by* block *and* sample-by-sample *processing.*
Give **one** *example of an application of* **each type**.

Block processing : data is transferred into a **memory buffer** and then processed each time the buffer is filled with new data. [1]
E.g. fast Fourier transforms (FFT), Discrete Cosine Transform (DCT), convolution, convolution reverb — **more soon** [1]

Sample-by-sample processing : input is processed on individual sample data. [1]
E.g. volume control, envelope shaping, modulation (ring/amplitude), IIR/FIR filtering.... [1]

4 Marks Total — Bookwork

(b) *Give definitions of the* transfer function *and* frequency response *of a digital system, in terms of its impulse response.*

Given an impulse response h(n) simply apply the Z-Transform:

$$H(z) = \sum_{n=-\infty}^{\infty} h(n).z^{-n}$$

to get the **transfer function** $H(z)$. [1]

Similarly apply the Fourier Transform:

$$H(f) = \sum_{n=-\infty}^{\infty} h(n).e^{-i2\pi fn/f_s}$$

to get the **Frequency Response** $H(f)$. [1]

2 Marks Total —- Bookwork

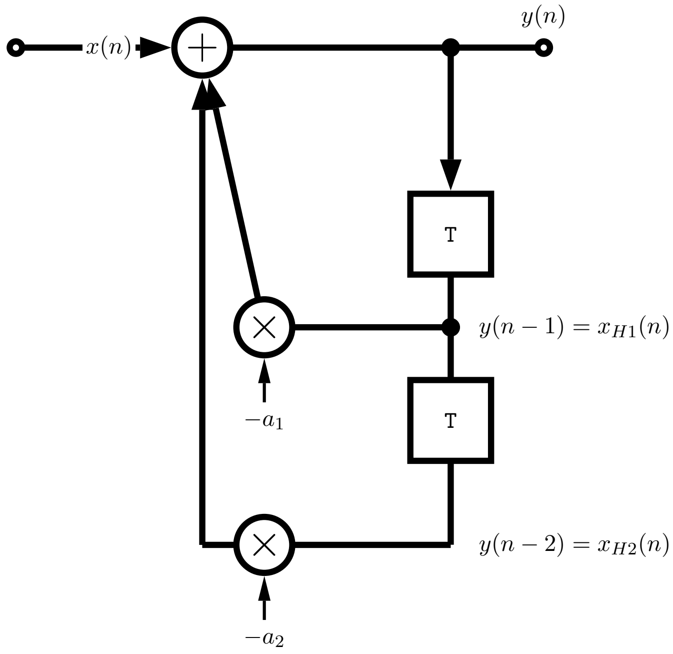
(c) *Briefly discuss* three *algorithmic approaches to implementing* filtering *in a digital system*

Infinite Impulse Response Filter (IIR) :

A simple IIR system can be described as follows:

$$y(n) = x(n) - a_1 y(n-1) - a_2 y(n-2)$$

- The output signal $y(n)$ is *fed back* through a series of delays
- Each delay is weighted
- Fed back weighted delay summed and passed to new output.
- Such a feedback system is called a **recursive system**

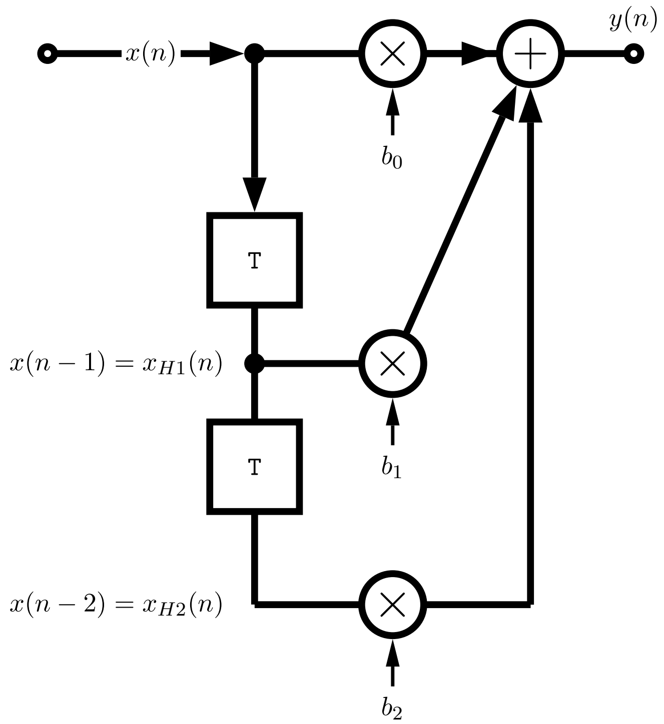


Finite Impulse Response Filter (FIR) :

A simple FIR system can be described as follows:

$$y(n) = b_0 x(n) + b_1 x(n-1) + b_2 x(n-2)$$

- The input is fed through delay elements
- Weighted sum of delays give $y(n)$



Fourier Space Filtering :

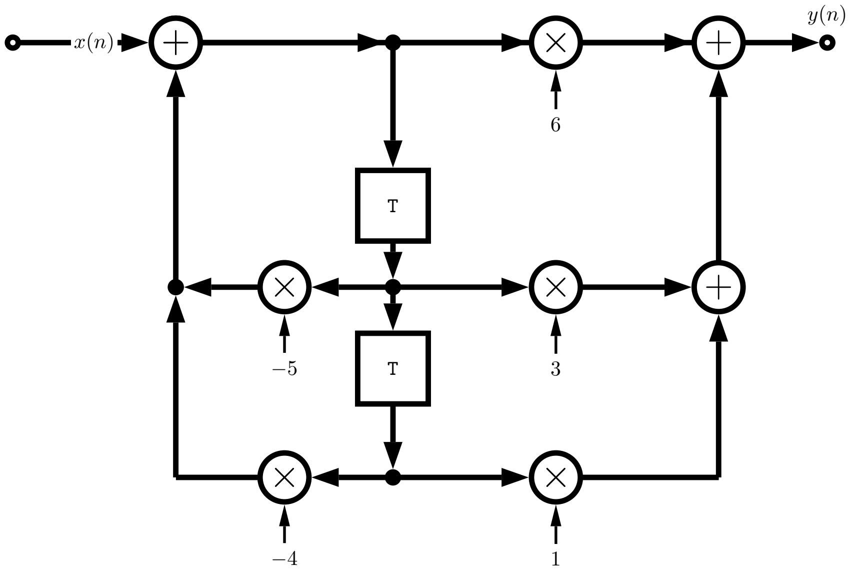
- $F(u, v)$ is the Fourier transform of the original image,
- $H(u, v)$ is a filter function (in Fourier Space) could be inverse of Fourier Transform of a real space filter function,
- $G(u, v = H(u, v)F(u, v))$ is the **Fourier transform of the improved image**.
- Inverse Fourier transform $G(u, v)$ to get $g(x, y)$ our **improved image**

TOTAL 9 Marks — 3 marks per method

(d) *Given the following difference equation construct its* signal flow diagram*:*

$$
\begin{aligned}
y(n) &= 6x(n) + 3x(n-1) + 1x(n-2) \\
&\quad -5y(n-1) - 4y(n-2)
\end{aligned}
$$

Solution:

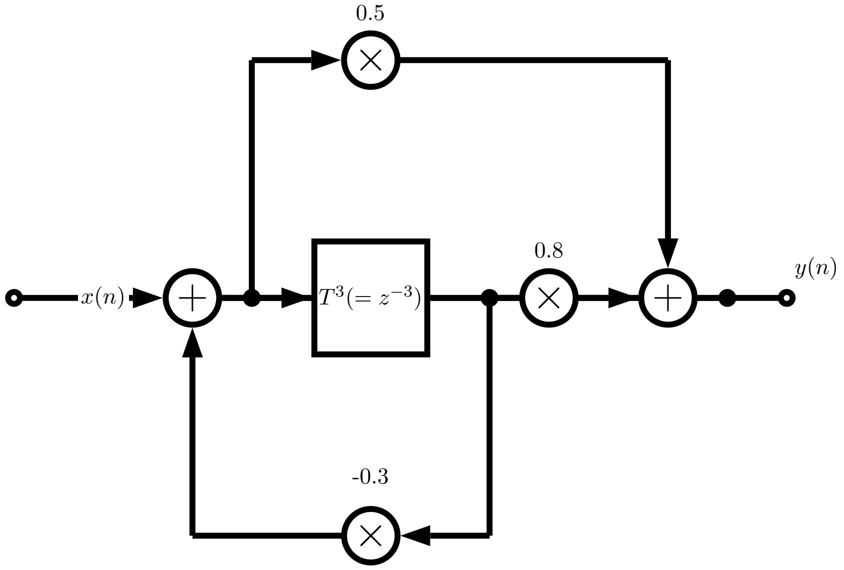


[6]

6 Marks — Unseen.

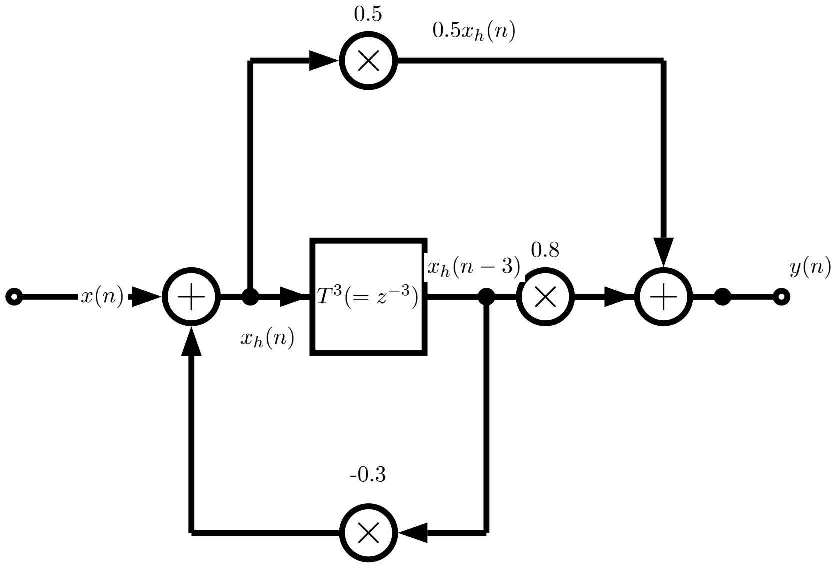
Possible other less efficient solutions with more than two delay units (not shared for $y(n-1)/x(n-1)\ldots$ **tapping award less marks if so**

(e) *Given the following signal flow diagram construct its* difference equation, $y(n)$*:*



$$
\begin{aligned}
x_h(n) &= x(n) - 0.3x(n-3) \\
y(n) &= 0.5x_h(n) + 0.8x_h(n-3)
\end{aligned}
$$

Trick is to break up feedback loop into sub equation $x_h(n)$.



[6]

6 Marks — Unseen.

Q3. (a) *What is the* difference *between* reverb *and* echo*?*

Echo — implies a distinct, delayed version of a sound, [1]

Reverb — each delayed sound wave arrives in such a short period of time such that we do not perceive each reflection as a copy of the original sound. [1]

TOTAL 2 Marks — Bookwork

(b) *Give the names of two filter based approaches to simulating the reverb effect in digitial audio.* [2]

Schroeder's Reverberator [1]

Moorer's Reverberator [1]

Comment on how one approach builds on the other and how filters are used to achieve the desired effect.par

Schroeder's Reverberator :

- Early digital reverberation algorithms tried to mimic the a rooms reverberation by primarily using **two types** of infinite impulse response (IIR) filters.

  Comb filter — usually in parallel banks

  Allpass filter — usually sequentially after comb filter banks

- A delay is (set via the feedback loops allpass filter) aims to make the output would gradually decay.

Moorer's Reverberator : Moorer's reverberator build's on Schroeder:

- Parallel comb filters with different delay lengths are used to simulate modes of a room, and sound reflecting between parallel walls [1]
- Allpass filters to increase the reflection density (diffusion). [1]
- Lowpass filters inserted in the feedback loops to alter the reverberation time as a function of frequency [1]
  - Shorter reverberation time at higher frequencies is caused by air absorption and reflectivity characteristics of wall). [1]
  - Implement a dc-attenuation, and a frequency dependent attenuation. [1]
  - Different in each comb filter because their coefficients depend on the delay line length [1]

6 Marks — Bookwork

(c) *State one alternative approach to reverb simulation that does not employ filters.*
Convolution Reverb [1]

1 Mark — Bookwork

Briefly, giving no mathematical detail, describe how this approach is implemented.

- record impulse response of room $g(x)$, input audio is $f(x)$ [1]
- compute Fourier transform of impulse response $G(u$ and audio signal, $F(u)$ [1]
- compute convolution of two signals, multiply both Fourier transforms $H(u) = F(u).G(u)$ [1]
- compute inverse Fourier transform of $H(u)$, $h(x)$ the reverberated signal.[1]

Total 4 marks — Bookwork

(d) *For each of the* three reverb *methods you have described above discuss how, in the following two scenarios, the sounds recorded by the microphone could be* modelled*:*

i. *A long hallway where the long walls are lined with a high frequency absorbing acoustic panels. The sound source is placed at one end of the hallway and a microphone is placed at the other end.*

Schroeder's Reverberator :

- estimate time of sound bouncing down corridor to set delay
- estimate some filtering of high frequencies

[2]

Moorer's Reverberator :

- estimate time of sound bouncing down corridor to set comb filter delay
- estimate some filtering of high frequencies for allpass filters and low-pass filters

[2]

Convolution Reverb :

- Record impulse response of hallway
- Perform convolution reverb computation.

[2]

Total 6 marks — unseen

ii. *A* cardoid microphone *is a microphone that accepts sound from the front and sides but not the back of the microphone.*
In a square recording studio, with uniform surfaces, a cardoid microphone is placed directly facing a sound source a few feet away.

Schroeder's Reverberator :

- Not much one can model except as before, estimate time of set short delay to account for no reflections recorded and estimate some filtering of high frequencies
- Can't easily model Cardoid response

[2]

Moorer's Reverberator :

- There will be little recording of back reflections so allow little feedback to comb filters
- Tapped delay lines which simulate early reflections could have delay and frequency filters set.

[2]

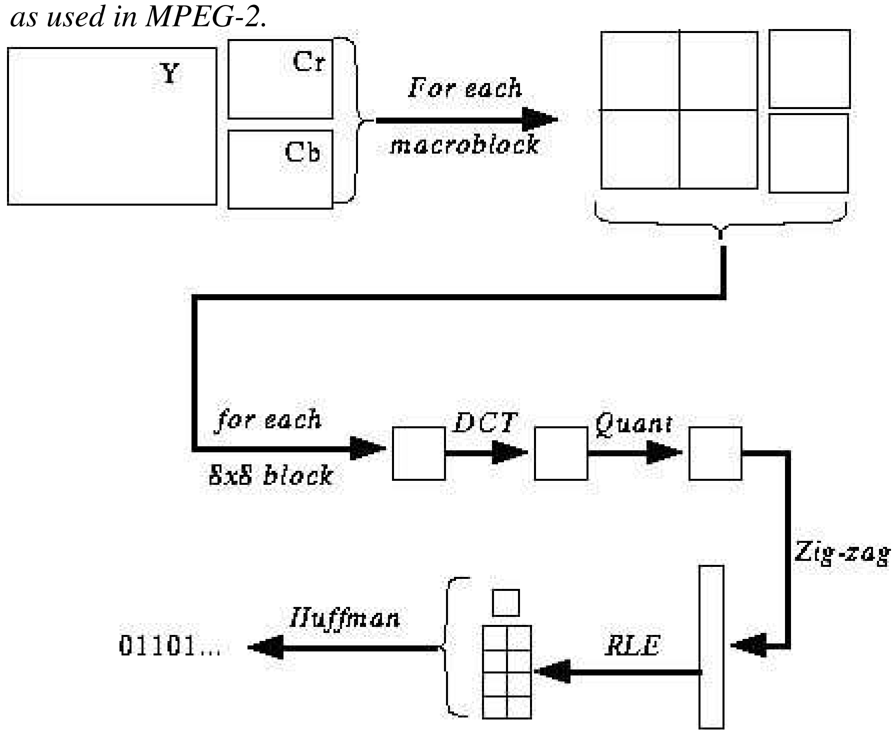
Convolution Reverb :

- Record impulse response of hallway with a **cardoid** microphone.
- Perform convolution reverb computation.

[2]

Total 6 marks — unseen

Q4. (a) *Briefly outline the basic principles of* Intra-Frame Coding *in Video Compression, as used in MPEG-2.*



This is a basic Intra Frame Coding Scheme is as follows:

- Convert tp **more effective** color space: YUV (YCbCr). [1]
- A **macroblock** usually consists of 4 Y blocks, 1 Cr block, and 1 Cb block. (4:2:0 chroma subsampling)
  - Since eye most sensitive luminance, less sensitive chrominance. [1]
- Break frame up into macroblocks which are typically 16x16 pixels. [1]
- Perform DCT on each Macroblock [1]
- Quantization is by constant value for all DCT coefficients. **I.e., no quantization table as in JPEG**. [1]
- Zig-zag vectorisation of quantised DCT coefficients [1]
- Run length encoding (RLE) on zig-zaf vector [1]
- Huffman coding on RLE values [1]

Total 8 Marks — bookwork

(b) What is the key *difference* between *I-Frames*, *P-Frames* and *B-Frames*?

I-Frame — Basic Reference frame for each Group of pictures — essentially a JPEG Compressed image. [1]

P-Frame — Coded forward Difference frame w.r.t last I or P frame [1]

B-Frame — Coded backward Difference frame w.r.t last I or P frame [1]

Total 3 Marks — Bookwork

(c) *Why are I-frames inserted into the compressed output stream relatively frequently?*

Differences between frames get too large — large errors – hard to track fast blocks etc. So need to restart card with a new I-frame. [2]

Total 2 Marks — Bookwork

(d) *Given the following coding order of a group of frames in MPEG-2:*

I	*P*	*B*	*B*	*B*	*P*	*B*	*B*	*B*	*I*	*B*	*B*	*B*	*I*	*P*	*B*	*P*
1	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*	*16*	*17*

What is display order of the frames?

Display Order is:

I	B	B	B	P	B	B	B	P	I	B	B	B	I	B	P	P
1	3	4	5	2	7	8	9	6	10	11	12	13	14	16	15	17

[7]

2 Marks for decoding the first IBBBP 1 Mark for decoing next BBBP (essentially a repeat of first block) 2 Marks for next IBBB — I frames cant change order so no change in order. 2 Marks for IBPP — only one (first in this sequence) P frame changes order.

7 Marks Unseen

(e) *The following macroblock window has a best sum of absolute difference (SAD) match of 1 to a given MPEG Interframe search:*

4	*2*
3	*5*

Should inter or intraframe coding be employed to code this macroblock, and why?

Method from lecture notes:

Based upon the motion estimation a decision is made on whether INTRA or INTER coding is made.

To determine INTRA/INTER MODE we do the following calculation:

$$MB_{mean} = \frac{\Sigma_{i=0,j=0}^{N-1}|C(i,j)|}{N}$$

$$A = \Sigma_{i=0,j=0}^{n,m} \mid C(i,j) - MB_{mean} \mid$$

If $A < (SAD - 2N)$ INTRA Mode is chosen.

So for this problem:

$MB_{mean} = 7$ [2]

$$\begin{aligned} A &= \mid 4-7 \mid + \mid 2-7 \mid + \mid 3-7 \mid + \mid 5-7 \mid \\ &= 3+5+4+2 \\ &= 14 \end{aligned}$$

[2] $SAD - 2N = -3$ [2] So A is not less than $(SAD - 2N)$ we choose **INTER FRAME Coding** [1]

7 marks — unseen problem application of bookwork formula.