CARDIFF UNIVERSITY
EXAMINATION PAPER

Academic Year:	2009/2010
Examination Period:	Autumn
Examination Paper Number:	CM0340 Solutions
Examination Paper Title:	Multimedia
Duration:	2 hours

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

Structure of Examination Paper:

There are 13 pages.
There are 4 questions in total.
There are no appendices.
The maximum mark for the examination paper is 80 and 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) *What is the* distinction *between lossy and lossless data compression?*

Lossless Compression — after decompression gives an exact copy of the original data [1]

Lossy Compression — after decompression gives ideally a 'close' approximation of the original data, in many cases perceptually lossless but a byte-by-byte comparison of files shows differences. [1]

2 Marks — Bookwork

Give one example *of a lossy and lossless compression algorithm.*

Lossless Compression Examples: 1 from Entropy Encoding Schemes (Shannon-Fano Huffman coding), arithmetic coding, LZW algorithm [1] used in GIF image file format. [1]

Lossy Compression Examples : 1 from Transform Coding (FFT/DCT based quantisation), differential encoding, vector quantisation [1]

2 Marks — Bookwork

(b) *List* three pattern substitution *based compression algorithms.*

Repetitive Sequence Suppression [1]

Run-length Encoding [1]

Pattern Substitution [1]

3 Marks — Bookwork

For each algorithm, give one application where the method is used with respect to multimedia data.

Repetitive Sequence Suppression Example: 1 from Silence suppression in audio, 'white space' in text, simple uniform backgrounds in images [1]

Run-length Encoding : 1 from Computer graphics generated images, Faxes, part of JPEG (latter stage) pipeline [1]

Pattern Substitution : 1 from Pattern recognition/token substitution, Entropy coding (Huffman), LZW/GIF, vector quantisation [1]

3 Marks — Bookwork

(c) *What is the basic concept used in defining an* Information Theoretic *approach to data compression*?

The **entropy** of an information source S, defined as:

$$H(S) = \eta = \sum_i \, p_i \log_2 \frac{1}{p_i}$$

,

is the basis *Information Theoretic* compression algorithms. [2]

2 Marks — Bookwork

(d) *Why is the* Huffman coding *algorithm* better *at data compression that the* Shannon-Fano *Algorithm?*

- (A bottom-up approach)
- 'Captures' the ideal entropy more closely that Shannon-Fano [2]

2 Marks — Bookwork

(e) *What* advantages *does the* arithmetic coding *algorithm offer over* Huffman coding *algorithm with respect to data compression?*

- Good compression ratio (better than Huffman coding), entropy around the Shannon Ideal value. [1]
  - Huffman coding uses an integer number of bits for each symbol, [1]
    * hence k is never less than 1.
  - Use decimal number of bits [1]

3 Marks — Bookwork

Are there any disadvantages with the arithmetic coding *algorithm?*

- Memory: potentially large symbol tables needed [1]
- Speed due possibly complex computations due to large symbol tables, [1]

2 Marks — Bookwork

(f) *Given the following Differential Pulse Code Modulated (DPCM) Sequence* reconstruct *the* original signal.

$$+4 + 2 + 3 - 2 + 3 - 1 + 1 + 1$$

DPCM decoding: Simply start with accumulator zero for each number add the value to current accumulator, output accumulator value.
So solution is:

$$4\ \ 6\ \ 9\ \ 7\ \ 10\ \ 9\ \ 11\ \ 12$$

[4]

4 Marks Unseen Problem: DPCM encoding covered in lectures

(g) *Given the following Run Length Encoded (RLE) Sequence* reconstruct *the* original 2D 8x8 (binary) data array.

$$
\begin{array}{l}
(0,8),\\
(0,1),(1,1),(0,4),(1,1),(0,1),\\
(0,1),(1,2),(0,2),(1,2),(0,1),\\
(0,1),(1,6),(0,1),\\
(0,2),(1,4),(0,2),\\
(0,3),(1,2),(0,3),\\
(0,2),(1,1),(0,2),(1,1),(0,2),\\
(0,1),(1,1),(0,4),(1,1),(0,1)
\end{array}
$$

The format of RLE is for each pair (colour, length) so just 'parse' each row, to expand colour to length number of values to get the **solution**:

0	0	0	0	0	0	0	0
0	1	0	0	0	0	1	0
0	1	1	0	0	1	1	0
0	1	1	1	1	1	1	0
0	0	1	1	1	1	0	0
0	0	0	1	1	0	0	0
0	0	1	0	0	1	0	0
0	1	0	0	0	0	1	0

[4]

4 Marks Unseen Problem: RLE encoding covered in lectures

Question 1 Total Marks 27

Q2. (a) *What is* MIDI?

Definition:

A protocol that enables computers, synthesizers, keyboards, and other musical devices to communicate with each other. [1]

1 Mark — Bookwork

(b) *What features of MIDI make it* suitable *for use in the* MPEG-4 audio compression standard*?*

- MIDI is very low bandwidth when compared to audio. Sounds synthesised at client only control data transmitted [1]
- MIDI can control many performance aspects [1]

2 Marks — Bookwork

(c) *Briefly outline the MPEG-4* structured audio *standard.* [6]

MPEG-4 Structured Audio Tools

MPEG-4 comprises of 6 Structured Audio tools are

SAOL the Structured Audio Orchestra Language: SAOL is the central part of the Structured Audio toolset. It is a software-synthesis language — defines how to make the sounds [1]

SASL the Structured Audio Score Language: MIDI controls how SAOL makes the sounds [1]

SASBF the Structured Audio Sample Bank Format: SASBF is a format for efficiently transmitting banks of sound samples to be used in wavetable, or sampling, synthesis. [1]

Set of MIDI semantics which describes how to control SAOL and SASL scripts (MIDI based) [1] . [1]

Scheduler which describes how to take the above parts and create sound: It is a set of carefully defined and somewhat complicated instructions that specify how SAOL is used to create sound when it is driven by MIDI or SASL. [1]

AudioBIFS part of BIFS, which lets you make audio soundtracks in MPEG-4 using a variety of tools and effects-processing techniques [1]

6 Marks — Bookwork

(d) *What* features *of MIDI make it suitable for* controlling *software or hardware devices*?

- Basic syntax of midi is all about control playing notes, setting up sounds etc. [1]
- A wide number of specialised control messages — some set like sustain, modulation, pitch bend others freely assignable. [1]
- Wide range of controllers available — e.g. built in to keyboards, specialist hardware, software reassignable. [1]
- Midi System Real-time Messages for control of timing/syncing e.g. SMPTE, Midi Time Code. [1]
- System exclusive command set makes MIDI completely ”extensible” to control any device. [1]
- External hardware to convert MIDI to/from other formats e.g. Pitch to Midi converters, Midi to Control Voltage (analogue synths), Motion Capture to MIDI!. [1]

6 Marks — unseen. Assimilation of various aspects of bookwork

(e) *In terms of controlling devices, what limitations does MIDI have in terms of the level of control, the number of devices and the number of independent control items within a device?*

Basic issues surrounding MIDI Control (Students may suggest others, marks will be awarded for any sensible suggestions). Example solution

- **Level of Control**: MIDI controllers only have an 8-bit resolution so ”fine tuned” control or control of a large range may not be possible. [2]
- **Number of Devices**: MIDI allows for 16 different channels on one MIDI connection, so up to 16 devices can only be controlled. [2]
- **Number of independent control items within a device**: Serial device based on ancient RS-232 Interface (although no upgraded to USB/FIREWIRE/ETHERNET). May have limited bandwidth. MIDI is essentially Serial device protocol — even though ”polyphony” in music has to be supported, all MIDI messages sent in a sequence (Fundamental serial resolution is about one message per millisecond). So ”MIDI clog” might occur if a lot of data needs to be sent rapidly. [2]
- (Alternative) **Number of independent control items within a device**: MID has a limited number of assignable controllers. 256 are allowed. [2]
- Possibly other suggestions.

6 Marks, 2 per solution for each limitation (example solution above) — Unseen question, Extended reasoning from basic MIDI understanding

Suggest a solution that can employed to remedy each of these problems using standard MIDI devices.

Indicative Solutions (Related to above):

- **Level of Control**: MIDI controllers only have an 8-bit resolution. **Solution**: Can utilise two controllers in tandem one for LSB one for MSB to increase resolution to 16 bit for example. [2]
- **Number of Devices**: MIDI allows for 16 different channels on one MIDI connection. **Solution**: MIDI allows for multiple devices (although this have bandwidth implication) USB/FIREWIRE/ETHERNET/LAN has adequate Bandwidth and may support multiple devices as standard. [2]
- **Number of independent control items within a device**: Serial device based on ancient RS-232 Interface (although no upgraded to USB/FIREWIRE/ETHERNET). **Solution**: As above USB/FIREWIRE/ETHERNET/LAN has adequate Bandwidth. Could also use independent devices using independent connections if practical [2]
- (Alternative) **Number of independent control items within a device**: MID has a limited number of assignable controllers. 256 are allowed. **Solution**: Could also use multiple independent devices possibly using independent connections if practical [2]
- Possibly other suggestions.

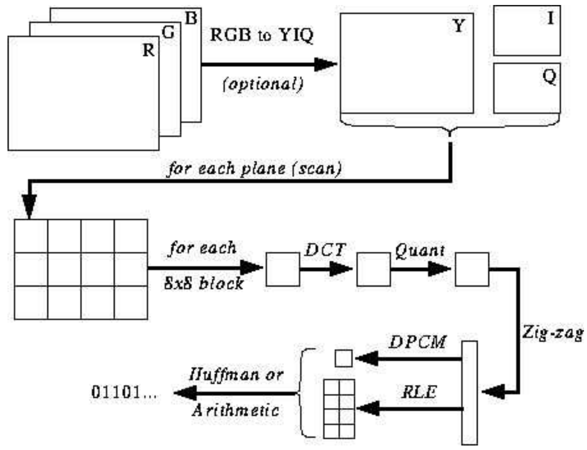
6 Marks, 2 per solution for each limitation (example solution above) — Unseen question, Extended reasoning from basic MIDI understanding

Question 2 Total Marks 27

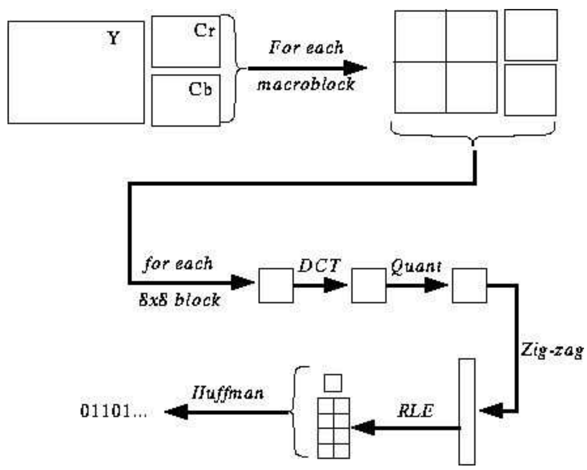
Q3. (a) *Briefly outline, with the aid of suitable diagrams, the* JPEG/MPEG I-Frame compression pipeline *and list* the constituent compression algorithms *employed at each stage in the pipeline.*

The Major Steps in JPEG/MPEG Coding involve:

JPEG:



MPEG:



[2]

- Colour Space Transform and subsampling
- DCT (Discrete Cosine Transformation)
- Quantization
- Zigzag Scan
- Discrete Pulse Code Modulation (DPCM) on DC component (in JPEG),
- Run length encoding (RLE) on AC Components (JPEG), all of zig zag (MPEG).
- Entropy Coding — Huffman or Arithmetic

[7]

9 Marks Bookwork

What are the key differences *between the JPEG and MPEG I-Frame compression pipeline?*

Four main differences for

- JPEG uses YIQ whilst MPEG use YUV (YCrCb) colour space [1]
- MPEG used larger block size DCT windows 16 even 32 as opposed to JPEG's 8 [1]
- Different quantisation — MPEG usually uses a constant quantisation value. [1]
- Only Discrete Pulse Code Modulation (DPCM) on DC component in JPEG on zig zag scan. AC (JEPG) and complete zig zag scan get RLE. [1]

4 Marks Applied Bookwork: Some lateral thinking to compare JPEG and MPEG not directly compared in course notes at least

(b) Motion JPEG (or M-JPEG*) is a video format that uses JPEG picture compression for* each frame *of the video. Why is M-JPEG* not widely used *as a video compression standard?*

Compressing in just each frame does not yield a high enough compression ratio that is required for general video needs. Can exploit temporal aspect of video to get better compression. [2]

2 Marks Bookwork

Briefly state what additional approaches *are used by MPEG video compression algorithms to improve on M-JPEG.*

Adopt some form of temporal compression. Use P-frames and B-frames to to differencing between frames and also motion estimation. [2]

2 Marks Bookwork

(c) *What processes above give rise to the* lossy *nature of JPEG/MPEG video compression?*

Lossy steps:

- Colour space subsampling in IQ or UV components. [2]
- Quantisation reduces bits needed for DCT components. [2]

4 Marks Bookwork

(d) *Given the following portion from a* block *(assumed to be 4x4 pixels to simplify the problem) from an image after the Discrete Cosine Transform stage of the compression pipeline has been applied::*

128	*32*	*64*	*160*
32	*16*	*12*	*32*
128	*64*	*46*	*128*
4	*31*	*40*	*32*

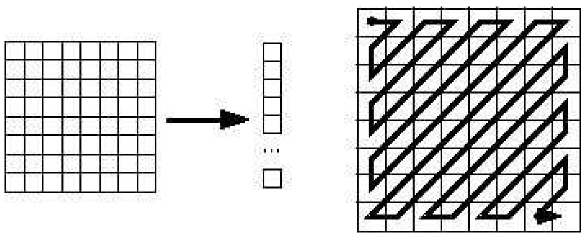
i. *What is the result of the* quantisation step *of the MPEG video compression method assuming that a constant quantisation value of 32 is used?*
Trick needed to be remembered from notes is that we divide the matrix by the quantisation table or in this case a constant.
So in this case divide all values by 32 and round down (Integer division).

4	1	2	5
1	0	0	1
4	2	1	4
0	0	1	1

[3]

ii. *What is the result of the following* zig-zag step *being applied to the quantised block?*

Trick needed to be remembered from notes is that Zig-zag reads of values from DCT in an increasing low frequency order (better that row by row). Create a vector rather than a matrix.



So we get a vector from matrix above:

4 1 1 4 0 2 5 0 2 0 0 1 1 4 1 4 [3]

6 Marks: Unseen Problem

Question 3 Total Marks 27

Q4. (a) *In MPEG audio compression, what is* frequency masking*?*

When an audio signal consists of multiple frequencies the sensitivity of the ear changes with the relative amplitude of the signals. If the frequencies are close and the amplitude of one is less than the other close frequency then the second frequency may not be heard. [2]

2 Marks: Bookwork

(b) *Briefly describe the* cause *of frequency masking in the human auditory system?*

Frequency Masking:

- Stereocilia in inner ear get excited as fluid pressure waves flow over them. [1]
- Stereocilia of different length and tightness on Basilar membrane so resonate in sympathy to different frequencies of fluid waves (banks of stereocilia at each frequency band). . [1]
- Stereocilia already excited by a frequency cannot be further excited by a lower amplitude near frequency wave. [1]

3 Marks: Bookwork

(c) *In MPEG audio compression, what is* temporal masking*?*

After the ear hears a loud sound, consisting of multiple frequencies, it takes a further short while before it can hear a quieter sound close in frequency. [2]

2 Marks: Bookwork

(d) *Briefly describe the* cause *of temporal masking in the human auditory system?*

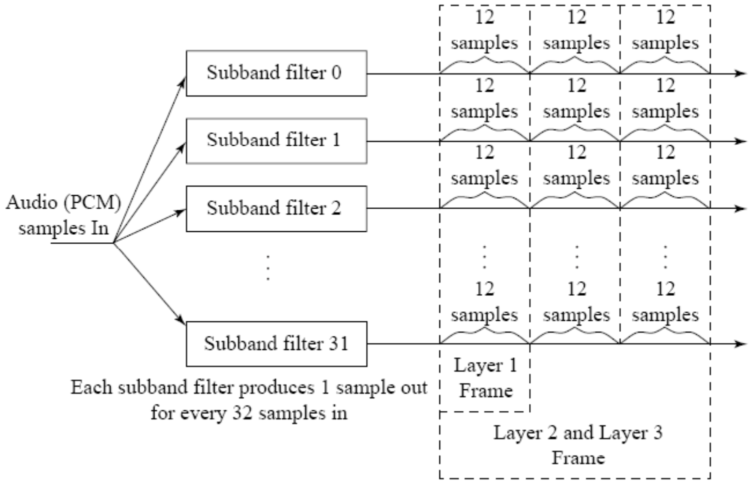
- (Like frequency masking) Stereocilia in inner ear get excited as fluid pressure waves flow over them and respond to different frequencies. [1]
- Stereocilia already excited by a certain frequency will take a while to return to rest state, as inner ear is a closed fluid chamber and pressure waves will eventually dampen down. [1]
- Similar to frequency masking Stereocilia in a 'dampening state' may not respond to a a lower amplitude near frequency wave. [1]

3 Marks: Bookwork

(e) *Briefly describe, using a suitable diagram if necessary, the MPEG-1 audio compression algorithm, outlining how frequency masking and temporal masking are encoded.*

MPEG audio compression basically works by:

- Dividing the audio signal up into a set of frequency subbands (Filtering) [1]



[2]

- Use filter banks to achieve this. [1]
- Subbands approximate **critical bands**. [1]
- Each band quantised according to the **audibility of quantisation noise**. [1]

Frequency masking and temporal masking are encoded by:

Frequency Masking MPEG Audio encodes this by quantising each filter bank with adaptive values from neighbouring bands energy, defined by a look up table.

[2]

Temporal Masking —
Not so easy to model as frequency masking. MP3 achieves this with a 50% overlap between successive transform windows gives window sizes of 36 or 12 and applies basic frequency masking as above. [2]

10 Marks: Bookwork

(f) *Given two stereo channels of audio:*

Left Channel:	14	11	10	16	17	20
Right Channel:	11	14	16	5	44	20

i. *Apply Middle/Side (MS) stereo redundancy coding to the sequence.* [3]
(Recap): **Stereo Redundancy—** at low frequencies, the human auditory system can't detect where the sound is coming from, So don' need stereo.
Middle/Side (MS) stereo redundancy coding Basic Idea:

- **Middle —** sum of left and right channels
- **Side —** difference of left and right channels.

So solution is :

Middle:	25	25	26	21	61	40
Side:	3	-3	-6	9	-27	0

3 Marks: Unseen Problem

ii. *How may this result be employed to achieve compression? Illustrate your answer with respect to the above data.*
Encode **side** in less bits as it is essentially Differential Pulse Code Modulation.
[1]
Use specially tuned threshold values to compress the side channel signal further. [1]
Code **Middle** in normal (for audio) 16 bits (8 Bits would be OK for this answer) [1]
Code **Side** in reduced number of bits. Needs to be ***signed*** so in the above 7 bits needs [1]

4 Marks: Unseen Problem

Question 4 Total Marks 27

 END OF EXAMINATION