CARDIFF UNIVERSITY
EXAMINATION PAPER

Academic Year:	2006/2007
Examination Period:	Autumn
Examination Paper Number:	CM0340 SOLUTIONS
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 **??** pages.
There are **??** 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) *Why is data compression desirable for multimedia activities?*

Data is high resolution in both temporal and/or spatial domains. [1]
Storgage, bandwidth and processing limitations need compressed data. [1]

2 MARKS — BOOKWORK

(b) *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

(c) *Consider the following sequence of 8-bit integers:*

4 6 9 11 13 12 13 14 12 11

Show how you would code this sequence using:

i. *Differential Pulse Code Modulation (DPCM)*

Sequence: +4 +2 +3 +2 +2 -1 +1 +1 -2 -1 [3]

ii. *Differential Pulse Code Modulation with delta modulation*

Sequence: 1 1 1 1 1 0 1 1 0 0 [3]

1 = increase/equality in data, 0 = decrease in data

iii. *Adaptive Differential Pulse Code Modulation (ADPCM) with window size of 3*

Averaged Sequence: 4 5 6 9 11 12 13 13 13 13 12 11
ADPCM Sequence: 4 +1 +1 +3 +2 +1 +1 0 0 0 -1 -1 [4]

10 Marks — UNSEEN application of knowledge/bookwork

Comment on the relative efficiency of these methods on the above sequence.

Compression relies on coding the differences in less bits In above DPCM needs at least 3bits as does ADPCM, Delat clearly 1 bit [1]

Delta Modulation using 1 bit so cant code any real differences [1]

ADPCM averages out errors/noise in data [1]

3 Marks — UNSEEN application of knowledge/bookwork

Q2. (a) *What is MIDI?*

Definition:

A protocol that enables computers, synthesizers, keyboards, and other musical devices to communicate with each other. [2]

2 Marks — bookwork

(b) *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

(c) *Briefly outline how MIDI might be used to control to following:*

i. *Musical disco lights that pulse different lights in time with some MIDI music.*

Assemble hardware converter that pulses different light banks to ranges of MIDI Notes [2]
Possible extend so that light banks respond to different MIDI Channels [1]

3 Marks Total

ii. *Video DJing where the playing and synchronisation of video and music clips is controlled via MIDI.*

A variety of possible solutions possible. Simplest solution based on some existing VJ software (www.arkaos.net)

Treat problem as live sequencing of Video and music samples/MIDI segments. [1]
Map out on keyboard like drum maps/samplers — one key per video/music clip. [1]
Effects/Transitions/Fades/Wipes — controlled by other midi events [1]
Assemble sequence of actions/transitions in software app a bit like a sequencer to learn new effects [1]
MIDI controllers can be assigned to control parameters of many effects. e.g volume, colour saturation, speed of transitions/effects. [1]

5 Marks Total

iii. *A motion capture MIDI interface where musical information is input via arm and upper body motion or gestures.*

Based on GypsyMIDI: MOCAP MIDI converter
(http://www.animazoo.com/products/gypsyRmidillo.htm)

Motion capture device capture points from body (e.g. wrist, elbow, shoulder. [1]
Specialist software converts this point data to MIDI [1]
Angles between joints, speed of points, gestures may be learnt [1]
Motion events mapped to midi events, configurable (perhaps — as in GypsyMidi). [1]
Midi events mapped to synthesiser.computer sequencer etc. [1]

5 Marks Total

iv. *A singing musical sampler/synthesiser where the synthesiser can take in an English phrases and sing these with a MIDI supplied melody.*

Based on Worbuilder Application (demoed in lecture but not explained) in EAST WEST SYMPHONIC CHOIR Software Sample/Synthesiser (http://www.soundsonline.com/EastWest-Quantum-Leap-Symphonic-Choirs-pr-EW-165.html)

Basic Idea is to map samples of words or parts of words (sung phonemes) to keys on the keyboard, just like drum samples or any triggered samples. (Choir sung phonemes actually a smaller set than spoken or single voice singing.) [1]
Organisation is via groups of phonemes and NOT pitch now across keyboard. So direct mapping of keyboard pitch to sample is not possible [2]
Solution use and intermediary software, Wordbuilder, that converts the sung phrase and note data to mappings to trigger actual sample [1]
Wordbuilder is much like a simple sequencer where timing of triggering phrases can be controlled via user interface. [1]
Phrases maybe be English (or Latin) but ideally in phonetic form. Wordbuilder does English to Phoneme conversion but direct phonetic input is possible [1]

6 Marks Total

19 Marks — Unseen. Most completely unseen. *MOCAP/VJ and Choir Sampler system briefly mentioned as an example of MIDI apps in lecture but not how they work. MOCAP involves so lateral thinking of how devices may work. Most others build on mapping of sounds/video to midi events (key presses on a keyboard) similar to percussion mapping which has been dealt with in lectures.*

Q3. (a) *What are critical bands in relation to the human ear's perception of sound?*

Frequncy masking occurs in human ear where certain frequencies are masked by neighbouring frequencies. Range of closeness for frequency masking depends on the frequencies and relative amplitudes. Each **band** where frequencies are masked is called **the Critical Band** [2]

2 Marks Bookwork

(b) *How does MPEG audio compression achieve critical band approximation?*

Create **Critical Band Filter Banks** [1]
Fourier Transform Audio Data [1]
Bandpass filter data according to know Critical Bandwidths [1]

3 Marks Bookwork

(c) *List* **three** *coding methods in MPEG audio coding that exploit different perceptual characteristics of the human ear when hearing sound. Briefly explain how these arise in the human ear and how these methods are implemented in MPEG audio compression.*

Three Methods

Frequency Masking — Stereocilia in inner ear get excited as fluid pressure waves flow over them. 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). Stereocilia take already excited by a frequency cannot be further excited by a lower amplitude near frequency wave. [2]

MPEG Audio Exploits this by quantising each filter bank with adaptive values from neighbouring bands energy.

[2]

Temporal Masking — Stereocilia need time to come to rest (well so do standing waves set up in closed inner ear). Until at rest stereocilia can't be excited.

[2]

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

Stereo Redundancy — at low frequencies, the human auditory system can't detect where the sound is coming from, So don' need stereo. [2]

Encode to mono and save bits. code low frequency critical bands in mono.

[2]

12 Marks Applied Bookwork: Some lateral thinking as topics cover different aspect of MPEG audio compression and Stereo Redundancy not related to Frequency/Temporal Masking

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

Left Channel:	*12*	*12*	*13*	*14*	*15*	*27*	*36*	*44*
Right Channel:	*12*	*14*	*16*	*4*	*44*	*20*	*2*	*3*

i. *Apply Middle/Side (MS) stereo coding to the sequence.*

Basic Idea:

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

Middle:	24	26	29	18	59	47	38	47
Side:	0	-2	-3	10	-29	7	34	41

[6]

6 Marks Applied bookwork (small part in notes)

ii. *How may this result be employed to achieve compression?*

Encode side in less bits as its 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 Bookwork applied to given data

Q4. (a) *What are the target media for JPEG compression?*

Colour and greyscale images [1]

Photographic Quality Images (Up 24 bit colour images) Many applications e.g., satellite, medical, general photography... [1]

2 Marks Bookwork

(b) What are the main *differences* between the target media for JPEG and GIF compression?

GIF can only deal with 8 colour bit images, JPEG colour 24bit [1]

GIF Target Graphical Images JPEG Photographic Quality Images [1]

2 Marks Bookwork

(c) *Compare the basic compression processes for JPEG and MPEG Intraframe coding. Your solution should outline the common basic processes for both, and particularly emphasise on the differences. Which steps in the process cause both JPEG and MPEG to be lossy*

The Major Steps in JPEG/MPEG Coding involve:

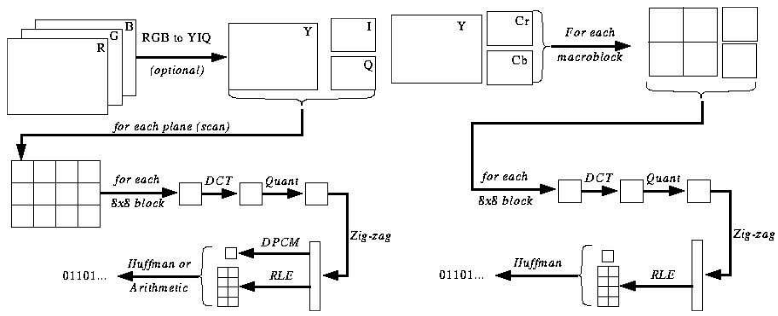


Figure 1: JPEG/MPEG Encoding

- 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]

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 ziz zag scan. AC (JEPG) and complete zig zag scan get RLE. [1]

Lossy steps:

- Colour space subsampling in IQ or UV components. [1]
- Quantisation reduces bits needed for DCT components. [1]

13 Marks Applied Bookwork: Some lateral thinking to compare JPEG and MPEG not in notes at least

(d) *Given the following portion from an 8x8 block from an image after the Discrete Cosine Transform has been applied:*

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

i. *What is the result of the quantisation step of the JPEG/MPEG compression method assuming that a constant quantisation value of 32 was used?*

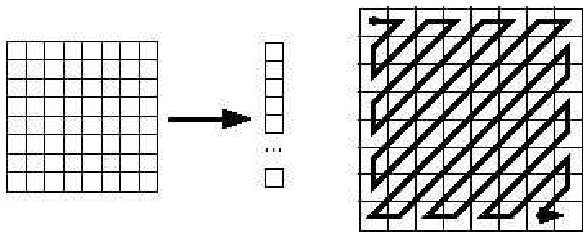
Trick needed to be remembered from notes is that we divide the matrix by the quantisation table or in this case a constant.
So in this case divide all values by 32 and round down.

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

iii. *What is the result of the following run length encoding (RLE) step being applied to the zig-zagstep's output?*

RLE: Replace repeating values with their value plus number of repetitions in the sequence.

Se we get a vector:

(4, 1), (2,1),(4, 1), (1,3), (4,1), (2,1), (0,4), (5,1), (1,3) [4]

10 Marks UNSEEN application of bookwork/knowledge