CARDIFF UNIVERSITY
EXAMINATION PAPER

SOLUTIONS

Academic Year:	2005-2006
Examination Period:	Spring 2006
Examination Paper Number:	CM0340
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 **THREE** pages.
There are **FOUR** questions in total.

There are no appendices.

The maximum mark for the examination paper is 81 marks, and the mark obtainable for each part of a question is shown in brackets alongside the question. Full marks can be obtained by correctly answering 3 questions.

Students to be provided with:

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

Instructions to Students:

Answer **THREE** questions.

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

1. *(a) Give a definition of a Multimedia System.*

A ***Multimedia System*** is a system capable of processing multimedia data and applications.

2 Marks Bookwork

(b) Briefly describe five Multimedia Authoring Paradigms.

Any 5 from:

- **Scripting Language** -- the Scripting paradigm is the authoring method closest in form to traditional programming. A powerful, object-oriented scripting language is usually the centerpiece of such a system e.g. Lingo.

- **Iconic/Flow Control --** This tends to be the speediest (in development time) authoring style; it is best suited for rapid prototyping and short-development time projects. Style is much like a schematics where icons represent key media objects or collections of objects and also control over the media. Linkage between media indicated by connections that control to flow of the data.

- **Frame** -- The Frame paradigm is similar to the Iconic/Flow Control paradigm but distinct groupings of media are via a Frame layout.

- **Card/Scripting** --- The Card/Scripting paradigm provides a great deal of power (via the incorporated scripting language) and an index-card style flow structure, similar to a frame

- **Cast/Score/Scripting** --- The Cast/Score/Scripting paradigm uses a music score as its primary authoring metaphor; the synchronous elements are shown in various horizontal tracks with simultaneity shown via the vertical columns.

- **Hierarchical Object** -- The Hierarchical Object paradigm uses a object metaphor (like OOP) which is visually represented by embedded objects and iconic properties. Although the learning curve is non-trivial, the visual representation of objects can make very complicated constructions possible.

- **Hypermedia Linkage** -- The Hypermedia Linkage paradigm is similar to the Frame paradigm in that it shows conceptual links between elements; however, it lacks the Frame paradigm's visual linkage metaphor.

- **Tagging** -- The Tagging paradigm uses tags in text files (for instance, SGML/HTML, SMIL (Synchronised Media Integration Language), VRML, 3DML and WinHelp) to link pages, provide interactivity and integrate multimedia elements.

5 Marks Bookwork (1 per paradigm)

(c) *Briefly describe five ways in which content can be formatted and delivered in a Multimedia Authoring System.*

1. Scripting (writing)
   Standard Text --- say what you want with word
2. Graphics (illustrating)
   "A picture is worth a thousand words" say what you want with a graphic illustrations
3. Animation (wiggling)
   Now we approach multimedia --- say what you want with a graphic animation or video
4. Audio (hearing)
   Sounds can convey alerts, ambience and contents say what you want with a narration
5. Interactivity (interacting)
   True mulitedia immerse yourself in am interactive presentation, possibly more instructive. Interactive actions can start animations, audio, move to new parts of presentation, control simulations etc.

10 Marks --- Bookwork (2 Marks per point)

(d) What extra information is multimedia good at conveying with respect to conventional media? Specifically:

(i) What can spoken text convey that written text cannot?

Spoken Text can convey:

Emotion more readily

Or Feelings more readily

Unwritten Sounds to express feelings or emotions example "tut tutting", sharp intakes of breath

Accents/dialects readily apparent

Important when more than speaker is present

Perhaps in getting certain messages across

E.g. assimilating a radio play is easier when we can distinguish speakers more easily

If in stereo position in 3D sound space is discernable

Possibly useful in creating a feeling of space or locating people in immersive 3D environments

Easier to author/synchronise with Video

Since Audio is already time dependent media, aligning media type on a timeline in a multimedia authoring package is usually simple

Writen text will need to be "animated" e.g. rollover credits, subtitles --- not difficult but some additional multimedia editing of raw text is required

(ii) When might written text be better than spoken text?

Hard to assimilate/remember a lot of spoken word

Written text easier to reread (as opposed to replay)

Additional information, such as

what a person looks like, what he is wearing, general appearance can be possibly more easily conveyed

Directions as to what else is happening, e.g a person moving around, waving arms more easily conveyed

e.g. Comparison a (radio) screenplay has plenty of stage directions

Written text has negligible bandwidth, high quality audio has significant bandwidth requirements

7 Marks Unseen Total (4 Marks for (i) 3 Marks for (ii))

2. *(a) Briefly explain how the human visual system senses colour. How is colour exploited in the compression of multimedia graphics, images and video?*

- The eye is basically just a “biological camera”
- Eye through lens etc focused light onto the Retina (back of eye)
- Retina consists of neurons that fire nerve signals proportional to incident light
- Each neuron is either a rod or a cone. Rods are not sensitive to colour.
- Cones organized in banks that sense red green and blue

Multimedia Context:

Since Rods do not sense colour only sense luminosity intensity Eye is more sensitve to luminence than colour. Also Eye is more sensitive to red and green and blue (this is due to evolution need to see fellow humans where blue is not prevalent in skin hues)

So any Multimedia compression techniques should use colour representation that presents colour in a way that models Human Visual system. We can then encode luminence is high bandwith (more bits) than colour as this is much more perceptually relevant.

5 Marks -- Bookwork

*(b) List **three** distinct models of colour used in Multimedia. Explain why there are a number of different colour models exploited in multimedia data formats.*

Possible models:

- *RGB*
- *CIE Chromaticity*
- *YIQ Colour Space*
- *YUV (YCrCb)*
- *CMY/CMYK*

Different models reflect need to represent colour in a perceptually relevant model for effective compression.
Different models also due to evolution of colour from Video (YIQ,YUV), Display (RGB) and Print (CMYK) media requirements.

9 Marks Bookwork ---- 3 marks 1 per model, 6 marks for explanation of different models

(*c*) *Compression of colour has been exploited since analog video. How was colour compression achieved in analog video? Compare this colour compression technique to those used in digital video*

Analog Video colour compression:

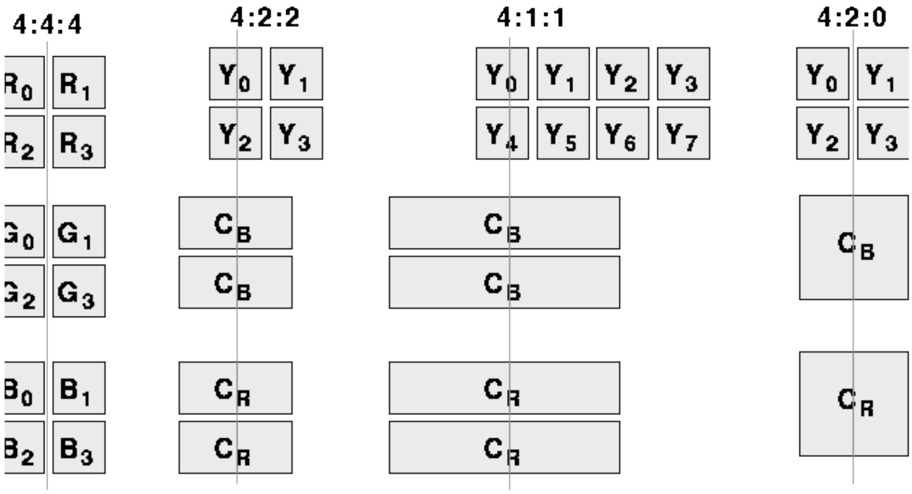
NTSC Video uses YIQ colour, PAL use YUV colour models

In NTSC, human ye is **most sensitive** to Y, **next** to I, **finally** Q., (Similarly for PAL with YUV).
SO use different frequency ranges (analog bandwisdth) in the signal to encode the channels using In NTSC, 4 MHz is allocated to Y, 1.5 MHz to I, 0.6 MHz to Q. In PAL Video U and V signals are lowpass filtered to about half the bandwidth of Y.

Digital Color Compression:

Similar idea but use different subsampling resolutions of each frame for each colour band. Called ***Chroma Subsampling***

E.G.



- 4:4:4 --- No subsampling in colour bands (Usually only if RGB model used)
- 4:2:2 -- Horizontally subsampled colour signals by a
- factor of 2. Each pixel is two bytes, e.g., (Cb0, Y0)(Cr0, Y1)(Cb2, Y2)(Cr2,
- Y3)(Cb4, Y4) ...
- 4:1:1 -- Horizontally subsampled by a factor of 4
- 4: :2:0 -- Subsampled in both the horizontal and vertical axes
- by a factor of 2 between pixels.

13 Marks --- Unseen. 7 Marks for Analog, 6 for Digital. Extended reasoning of notes. Analog compression briefly mentioned, digital more extensively dealt with.

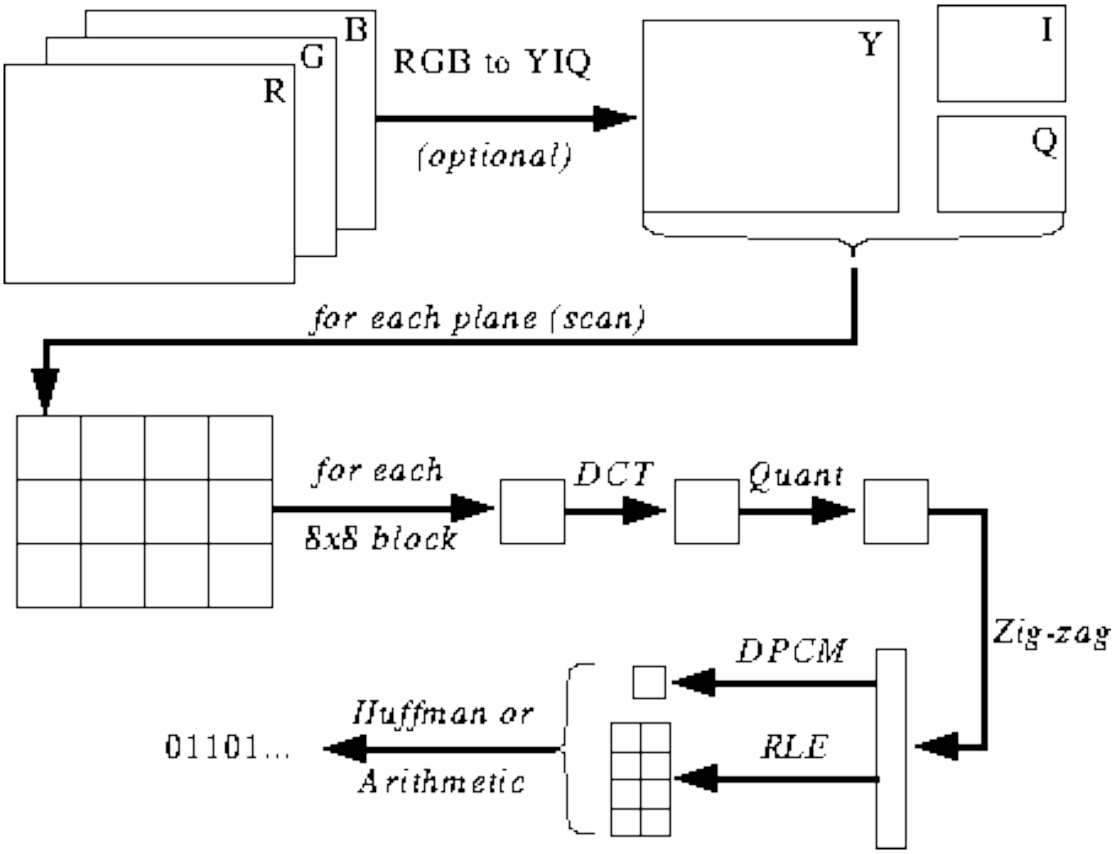
3. *(a) What is the distinction between lossy and lossless data compression?*

- **Lossless** Compression -- Where data is compressed and can be reconstituted (uncompressed) without loss of detail or information. These are referred to as bit-preserving or reversible compression systems also.

- **Lossy** Compression -- Where the aim is to obtain the best possible *fidelity* for a given bit-rate or minimizing the bit-rate to achieve a given fidelity measure. Video and audio compression techniques are most suited to this form of compression.

2 Marks Bookwork

(b) *Briefly outline the JPEG compression pipeline and the constituent compression algorithms employed at each stage in the pipeline.*



Stages of JPEG:

- **RGB to YIQ** --- colour subsampling into Y and subsampled I, Q channels
- **Discrete Cosine Transform** --- converiion to frequency domain (no compression yet)
- **Quantisation** --- “low pass” filter of DCT data using tables or uniform quantization

- **Zig-Zag Scan** -- writes out a vector from each 8x8 block low
- frequency coefficients first. (Maps 8 x 8 to a 1 x 64 vector)
- **Differential Pulse Code Modification (DPCM) on DC channel** --- DC component is large and varies but often close to previous value SO encode the difference from previous 8x8 blocks.
- **Run Length Encoding (RLE)** -- applied to
- the AC component. Each 1x64 vector has lots of zeros in it SO encode as RLE pairs
- **Entropy Coding (Huffman or Arithmetic)** -- DC and AC components finally need to be represented by a smaller number of bits so use Huffman or Arithmetic Coding compresson.

12 Marks Bookwork (5 For basic Outline diagram of pipeline, 7 marks for each stage

(c) (i) *Apply differential pulse code modulation to compress the following stream of integer numbers.*

8 7 4 6 3 4 5 6

Simple coding scheme Transmit the difference between successive tokens in the stream. (Can use less bits to encode the difference)

So coded stream is

0 +8 -1 -3 +2 -3 +1 +1 +1

3 Marks

If only 3 bits are used in the compressed stream encode what problems if any will occur with the above coding?

3 bits **cant** encode a ***difference of 8*** so there will be so error (overflow) in the coding.

3 Marks

TOTAL 6 Marks Unseen

(ii) *Apply run length encoding to compress following stream of alphabetical tokens:*

ABBAARNOOGOODEEEHHHHH

Comment on the efficiency of RLE encoding on the above token stream.

RLE: Simply code repeating tokens as pair of numbers that says number of repeats plus token.

Coded RLE stream:
(A,1), (B,2), (A,2), (R,1), (N,1), (O,2), (G,1), (O,2), (D,1), (E,3), (H,5)

4 Marks

Efficiency:

Orifinal Stream is 21 tokens (bytes in this case)

Compressed stream is 22 tokens/numbers (bytes if 8-but integer coding assumed)

SO COMPRESSED STREAM is actually larger by 1 byte!

RLE only gives good compression with a high number of repeated elemets

3 marks

Total 7 Marks Unseen

4. (a*) What is MIDI?*

Definition of MIDI: a protocol that enables computer, synthesizers, keyboards, and other musical device to communicate with each other.

2 Marks – Basic Bookwork

(b) *How is a basic MIDI message structured?*

Structure of MIDI messages:

- MIDI message includes a status byte and up to two data bytes.
- Status byte
- The most significant bit of status byte is set to 1.
- The 4 low-order bits identify which channel it belongs to (four bits produce 16 possible channels).
- The 3 remaining bits identify the message.
- The most significant bit of data byte is set to 0.

4 Marks – Basic Bookwork

(c) *What features of MIDI make it suitable for multimedia applications? Briefly justify your answer. What are the drawbacks of MIDI?*

MIDI is very low bandwidth when compared to audio. Sounds synthesised at client only "control data" transmitted

MIDI can control many performance aspects: what notes played, how they are played expression, dynamics, volume etc., what sound (instrument) makes the noise.

MIDI can support polyphonic (many sounds and many instruments)

General MIDI defines a standard sound set for consistent instrumentation across all platforms

Drawbacks:

Control of actual sounds made (audio synthesis) is delegated to client application so fidelity of sounds not guaranteed. General MIDI defines similar instrument set but NOT quality of sound. Cost (Computer resources, price etc.) define how high quality the sound generation is.

8 Marks Extended Reasoning

(d) *How is MIDI used within the MPEG-4 audio compression standard?*

MPEG-4 covers the the whole range of digital audio:

- from very low bit rate speech
- to full bandwidth high quality audio
- built in anti-piracy measures
- ***Structured Audio (MIDI PART OF THIS)***

Structured Audio Tools

MPEG-4 comprises of 6 ***Structured Audio tools*** are

- **SAOL** the Structured Audio Orchestra Language
- **SASL** the Structured Audio Score Language
- **SASBF** the Structured Audio Sample Bank Format
- **a set of MIDI semantics** which describes how to control SAOL with MIDI
- **a scheduler** which describes how to take the above parts and create sound
- **the AudioBIFS** part of BIFS, which lets you make audio soundtracks in MPEG-4 using a variety of tools and effects-processing techniques

MIDI is the “control” of the first 5 above:

SAOL is the central part of the Structured Audio toolset. It is a software-synthesis language; it was specifically designed it for use in MPEG-4. SAOL is not based on any particular method of synthesis. It is general and flexible enough that any known method of synthesis can be described in SAOL. Examples of FM synthesis, physical-modeling synthesis, sampling synthesis, granular synthesis, subtractive synthesis, FOF synthesis, and hybrids of all of these in SAOL.

MIDI controls how SAOL makes the sounds (like it controls synthesizers) through SASL variant and more ordinary MIDI semantics:

SASL is a very simple language that was created for MPEG-4 to control the synthesizers specified by SAOL instruments. SASL is like MIDI in some ways, but doesn't suffer from MIDI's restrictions on temporal resolution or bandwidth. It also has a more sophisticated controller structure than MIDI, you can write controllers to do anything.

SASBF is a format for efficiently transmitting banks of sound samples to be used in wavetable, or sampling, synthesis. The format is least partly compatible with the MIDI Downloaded Sounds (DLS) format.

MIDI Semantics

As well as controlling synthesis with SASL scripts, SAOL can be controlled with MIDI files and scores in MPEG-4. MIDI is today's most commonly used representation for music score data, and many sophisticated authoring tools (such as sequencers) work with MIDI.

The MIDI syntax is external to the MPEG-4 Structured Audio standard; only references to the MIDI Manufacturers Association's definition in the standard. But in order to make the MIDI controls work right in the MPEG context, some semantics (what the instructions "mean") have been redefined in MPEG-4. The new semantics are carefully defined as part of the MPEG-4 specification.

Scheduler: The scheduler is the "guts" of the Structured Audio definition. It's 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.

13 (2 marks for each part of MIDI in Structured Audio framework 1 MARK FOR OVERAL FRAMEWORK) Marks Extended Reasoning from bookwork