CARDIFF UNIVERSITY
EXAMINATION PAPER

SOLUTIONS

Academic Year:	2003-2004
Examination Period:	Postgraduate
Examination Paper Number:	CMP632
Examination Paper Title:	Multimedia Systems
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 100% 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 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) What are the key characteristics of a Multimedia System?

A ***Multimedia System*** is characterised by the processing, storage, generation, manipulation and rendition of Multimedia information.

A Multimedia system has four basic characteristics:

- Multimedia systems must be ***computer controlled*** .
- Multimedia systems are ***integrated*** .
- The information they handle must be represented ***digitally*** .
- The interface to the final presentation of media is usually ***interactive*** .

4 Marks Bookwork

(c) *Briefly describe eight hardware or software features that a Multimedia System should possess.*

The following features should be present:

- **Very High Processing Power**
  - needed to deal with large data processing and real time delivery of media. Special hardware commonplace.
- **Multimedia Capable File System**
  - needed to deliver real-time media -- ***e.g.*** Video/Audio Streaming. Special Hardware/Software needed ***e.g*** RAID technology.
- **Data Representations/File Formats that support multimedia**
  - Data representations/file formats should be easy to handle yet allow for compression/decompression in real-time.
- **Efficient and High I/O**
  - input and output to the file subsystem needs to be efficient and fast. Needs to allow for real-time recording as well as playback of data. ***e.g.*** Direct to Disk recording systems.
- **Special Operating System**
  - to allow access to file system and process data efficiently and quickly. Needs to support direct transfers to disk, real-time scheduling, fast interrupt processing, I/O streaming ***etc.***
- **Storage and Memory**
  - large storage units (of the order of 50 -100 Gb or more) and large memory (50 -100 Mb or more). Large Caches also required and frequently of Level 2 and 3 hierarchy for efficient management.
- **Network Support**
  - Client-server systems common as distributed systems common.
- **Software Tools**
  - user friendly tools needed to handle media, design and develop applications, deliver media.

4 Marks (half mark per feature) Bookwork

(d) The main types of multimedia data are: graphics, images, audio, s and video. What technical issues are associated when these data types are integrated in a Multimedia System?

For each media type briefly relate to the issues involved in generating, capturing, storing and transmitting the respective media components.

Multimedia issues (**2 Marks**):

- Sequencing within the media -- *playing frames in correct order/time frame in video*
- Synchronisation -- inter-media scheduling
- How to represent and store temporal information.

Data Types **(12 Marks 1mark per type per issue):**

Graphics

Source: Graphics input devices include: keyboard (for text and cursor control), mouse, trackball or graphics tablet.

Storage: Graphics are usually constructed by the composition of primitive objects such as lines, polygons, circles, curves and arcs. Graphics are usually generated by a graphics editor program (***e.g.*** Freehand) or automatically by a program (***e.g.*** Postscript usually generated this way). Graphics are usually editable or revisable (unlike Images).

Transmission: Graphics files usually store the primitive assembly and do not take up a very high overhead for bandwidth (e.g. Flash Vector Graphics)

Images

Source: Images may be generated by programs similar to graphics or animation programs. But images may be scanned for photographs or pictures using a digital scanner or from a digital camera. Some Video cameras allow for still image capture also. Analog sources will require digitising.

Storage: Images may be stored at 1 bit per pixel (Black and White), 8 Bits per pixel (Grey Scale, Colour Map) or 24 Bits per pixel (True Colour). Thus a 512x512 Grey scale image takes up 1/4 Mb, a 512x512 24 bit image takes 3/4 Mb with no compression.

Transmission: The storage overhead increases with image size so **compression** is commonly applied at source for bandwidth/storage e.g. JPEG

Audio

Source: They are first captured by a microphones and then digitised and stored on disk

Storage: usually compressed as CD quality audio requires 16-bit sampling at 44.1 KHz. So 1 Minute of Mono CD quality audio requires 60*44100*2 Bytes which is approximately 5 Mb.

Transmission/Bandwidth: Maybe be compressed (MPEG) perhaps streamed/copressed (Realaudio). Wav files may not be compressed

Video

Source:Analog Video is usually captured by a video camera and then digitised. May be digitised at source (Digital Video Camera)

Storage: Raw video can be regarded as being a series of single images. There are typically 25, 30 or 50 frames per second. Therefore a 512x512 size monochrome video images take 25*0.25 = 6.25Mb for a minute to store uncompressed. Digital video clearly needs to be compressed.

Transmission/Bandwidth: Maybe be compressed (MPEG) perhaps streamed/copressed (Realvideo/Quicktime).

14 Marks Total: Unseen --- Assimilation/Extended reasoning of a few parts of the course

2. (a) What does *Nyquist's Sampling Theorem* state?

In order to effectively sample a waveform the sampling frequency must be **at least** twice that of the highest frequency present in the signal

2 Marks --- Bookwork

(b) What are the implications of Nyquist's Sampling Theorem for multimedia data?

Sampling frequency affects the quality of the data ---- higher frequency equals better sampling hence representation of the underlying signal (given fixed frequency range of signal)

Sampling frequency affects size of digitized data --- higher frequency means more samples therefore more data.

4 (2 marks each above) Marks --- Bookwork

(c) For each of the following media types, graphics, images, audio and video, briefly discuss how Nyquist's Sampling Theorem affects the quality of the data and the form in which sampling effects manifest themselves in the actual data.

Graphics

- Quality: Not an issue with vector graphics
- Sampling Artifact: Rendering may lead to Aliasing effect in lines etc

Images

- Quality: Image size decreases so less detail or sampling artifacts
- Sampling Artifact: Aliasing effect in blocky images

Audio

- Quality: Lack of clarity in high frequencies, telephonic voices at low sampling frequencies
- Sampling Artifact: Digital noise present in signal, loss of high frequencies or poor representation of high frequencies give audio aliasing (should be filtered out before sampling)

Video

- Quality: Video Frame size decreases so less detail or sampling artifacts, motion blur or loss of motion detail
- Sampling Artifact: Aliasing effect in frame images, jittery motion tracking etc.

12 (3 Marks per media type) Marks --- Unseen: Extended reasoning on a few parts of course

(d) Calculate the uncompressed digital output if a video signal is sampled using the following values:

25 frames per second
160 x 120 pixels
True (Full) colour depth

True color = 24 bits (3 bytes) per pixel

So **number of bytes per second is**

3*160*120*25 = 144000 bytes or 1.37 Mb

3 Marks --- Unseen: Application of basic knowledge

(e) If a suitable CD stereo quality audio signal is included with the video signal in part d what compression ratio would be needed to be able to transmit the signal on a 128 kbps channel?

Stereo audio = 44100*2 (16 bit/s byte)*2 = 176400 bytes per second

So uncompressed bytes stream is 144000 + 176400 = 320400 bytes per second

128 kbps is kilo **<u>bits</u>** per second

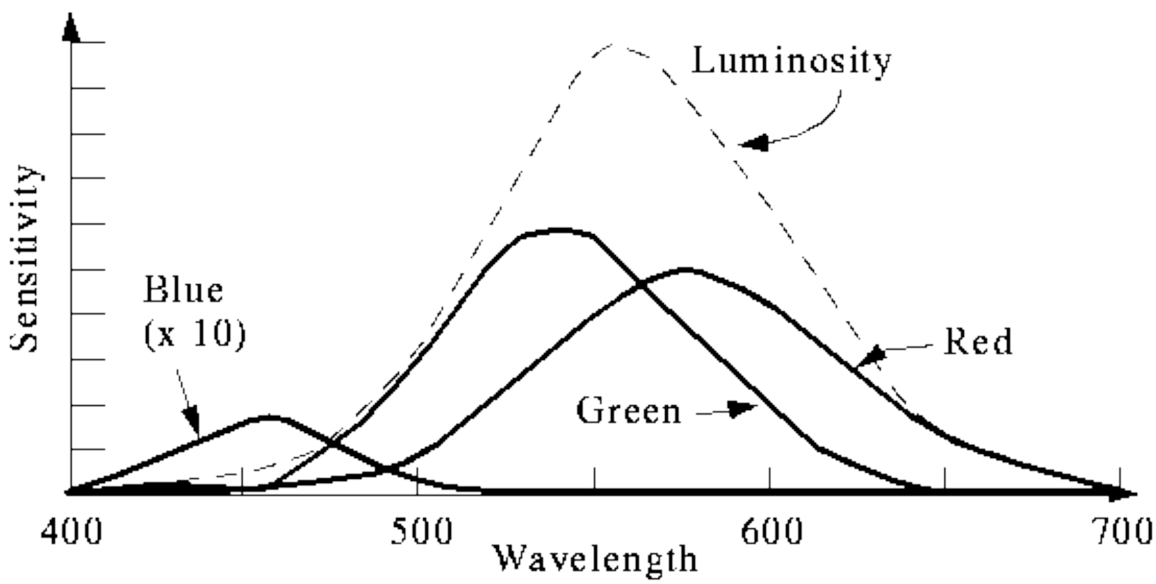
so compression ratio is (128*1024)/(320400 *8) = 0.05.

3 Marks --- Unseen: Application of basic knowledge

3. (a) What characteristics of the human visual system can be exploited in related
to compression of colour images and video?

The eye is basically sensitive to colour intensity

- Each neuron is either a *rod* or a *cone* . Rods are not sensitive to colour.
- Cones come in 3 types: red, green and blue.
- Each responds differently --- Non linearly and not equally for RGBdifferently to various frequencies of light.



5 Marks --- Bookwork

(b) What is the *YIQ color model* and why is this an appropriate color model used in conjunction with compression methods such as JPEG and MPEG?

- YIQ is origins in colour TV broadcasting
- Y (luminance) is the CIE Y primary.
- $Y = 0.299R + 0.587G + 0.114B$
- the other two vectors:
- $I = 0.596R - 0.275G - 0.321B$ $\qquad Q = 0.212R - 0.528G + 0.311B$
- The YIQ transform:

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.275 & -0.321 \\ 0.212 & -0.528 & 0.311 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

How to exploit to compression:

- Eye is most sensitive to Y, next to I, next to Q.
- Quantise with more bits for Y than I or Q.

4 (2 for Transform (Matrix or Eqn) and 2 for Compression scheme) Marks --- Bookwork

(c) Given the following YIQ image values:

Y

128	*126*	*127*	*129*
124	*123*	*124*	*124*
130	*136*	*132*	*132*
154	*143*	*132*	*132*

I

55	*66*	*54*	*54*
56	*57*	*56*	*56*
45	*56*	*58*	*49*
34	*36*	*39*	*37*

Q

44	*44*	*55*	*55*
44	*44*	*55*	*55*
34	*34*	*36*	*35*
35	*35*	*34*	*34*

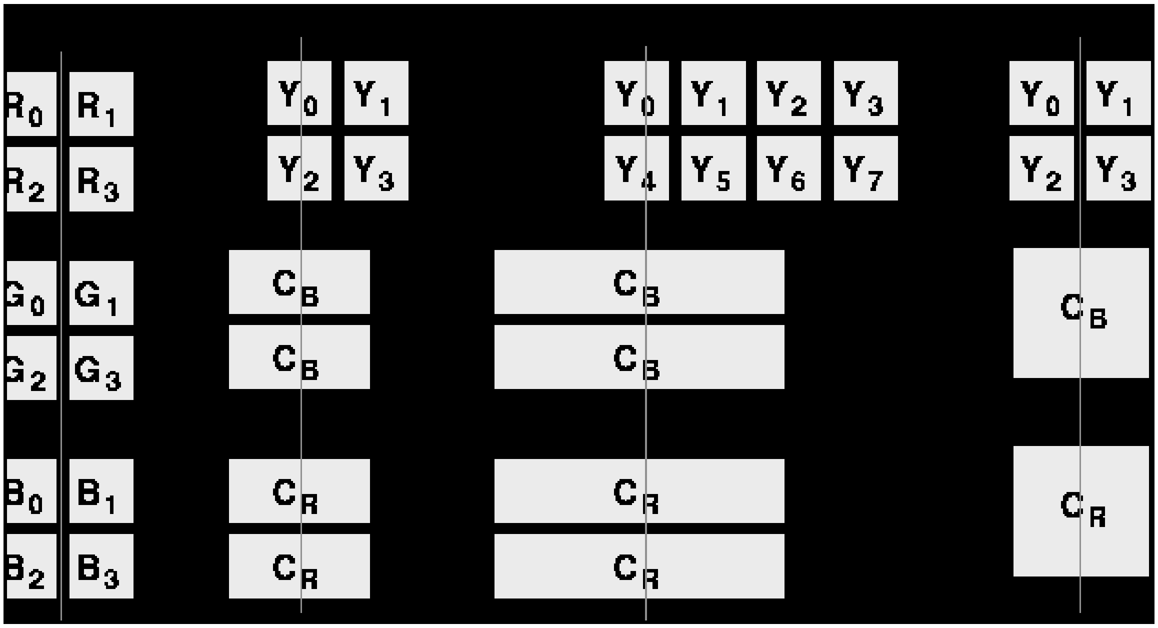
What are the corresponding chroma subsampled values for a

(i) 4:2:2 subsampling scheme
(ii) 4:1:1 subsampling scheme
(iii) 4:2:0 subsampling scheme

Basic Idea required (from notes):

Chroma Subsampling

- 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



(i) 4:2:2 subsampling scheme

Take average of every two horizontal pixels in I Q Space

128	126	127	129
124	123	124	124
130	136	132	132
154	143	132	132

55	66	54	54
56	57	56	56
45	56	58	49
34	36	39	37

44	44	55	55
44	44	55	55
34	34	36	35
35	35	34	34

Full YIQ

128	126	127	129
124	123	124	124
130	136	132	132
154	143	132	132

661	54
57	56
551	54
335	38

44	55
44	55
34	36
35	34

4:2:2
YIQ

(ii) 4:1:1 subsampling scheme

128	126	127	129
124	123	124	124
130	136	132	132
154	143	132	132

55	66	54	54
56	57	56	56
45	56	58	49
34	36	39	37

44	44	55	55
44	44	55	55
34	34	36	35
35	35	34	34

Full YIQ

128	126	127	129
124	123	124	124
130	136	132	132
154	143	132	132

57
56
52
37

50
50
35
35

4:1:1
YIQ

(iii) 4:2:0 subsampling scheme

1126	1127	129
1123	1124	124
1136	132	132
1143	132	132

55	66	54	54
56	57	56	46
45	56	58	49
34	36	39	37

44	44	55	55
44	44	55	55
34	34	36	35
35	35	34	34

6	127	129
3	124	124
6	132	132
3	132	132

59	55
43	46

44	55
35	35

15 Marks --- Unseen: Practical Application of Bookwork Knowledge

4. (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 describe two repetitive suppression algorithms and give one practical use of each algorithm.

1. **Simple Repetition Suppresion**

If in a sequence a series on *n* successive tokens appears we can replace these with a token and a count number of occurences. We usually need to have a special ***flag*** to denote when the repated token appears

For Example

89400000000000000000000000000000000

we can replace with

894f32

where fis the flag for zero.

Compression savings depend on the content of the data.

Applications of this simple compression technique include:

- Silence in audio data, Pauses in conversation ***etc.***
- Bitmaps
- Blanks in text or program source files
- Backgrounds in images
- other regular image or data tokens

2. Run-length Encoding

In this method, sequences of (image) elements X1, X2 … Xn are mapped to pairs (C1,l1), (C2,L2) …… (Cn,Ln) where *ci* represent image intensity or colour and *li* the length of the *i*th run of pixels.

For example:

Original Sequence:

111122233333311112222

can be encoded as:

(1,4),(2,3),(3,6),(1,4),(2,4)

The savings are dependent on the data. In the worst case (Random Noise) encoding is more heavy than original file: 2*integer rather 1* integer if data is represented as integers.

Applications:

- This encoding method is frequently applied to images (or pixels in a scan line).
- It is a small compression component used in JPEG compression

10 Marks Bookwork

(c) *Briefly state the LZW compression algorithm and show how you would use it to encode the following steam of characters*:

MYMEMYMO

You may assume that single character tokens are coded by their ASCII codes, as per the original LZW algorithm. However, for the purpose of the solution you may simply output the character rather than the ASCII value.

The LZW Compression Algorithm can summarised as follows:

```
w = NIL;
   while ( read a character k )
       {
         if wk exists in the dictionary
          w = wk;
         else
           add wk to the dictionary;
           output the code for w;
           w = k;
       }
```

Original LZW used dictionary with 4K entries, first 256 (0-255) are ASCII codes.

Encoding of ***MYMEMYMO:***

W	**K**	**Output**	**Index**	**Symbol**
nil	M			
M	Y	‘M’ (ASCII)	256	MY
Y	M	‘Y’	257	YM
M	E	‘M’	258	ME
E	M	‘E’	259	EM
M	Y			
MY	M	256	260	MYM
M	O	‘M’	261	MO

So Token Stream is

MYME<256>M

12 Marks Unseen: Application of Algorithm