CARDIFF CARDIFF UNIVERSITY
EXAMINATION PAPER

SOLUTIONS

Academic Year:	2000-2001
Examination Period:	Lent 2001
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.

1. *(a) Why is data compression necessary for Multimedia activities?*

Multimedia files are very large therefore for storage, file transfer etc. file sizes need to be reduced often as part of the file format. Text and other files may also be encoded/compressed for email and other applications.

3 Marks -- Bookwork

(b) Briefly explain how the LZW Transform Operates. What common compression methods utilise this transform?

Suppose we want to encode the Oxford Concise English dictionary which contains about 159,000 entries. Why not just transmit each word as an 18 bit number?

Problems:

* Too many bits,
* everyone needs a dictionary,
* only works for English text.
* **Solution**: Find a way to build the dictionary adaptively.

- Original methods due to Ziv and Lempel in 1977 and 1978. Terry Welch improved the scheme in 1984 (called LZW compression).
- It is used in UNIX *compress* and GIF compression

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.

10 MARKS – BOOKWORK

(c) Show how the LZW transform would be used to encode the following sequence of text based tokens

ABCBCABCEAB

w	**k**	**Output**	**Index**	**Symbol**
NIL	A			
A	B	‘A’	256	AB
B	C	‘B’	257	BC
C	B	‘C’	258	CB
B	C			
BC	A	257	259	BCA
A	B			
AB	C	256	260	ABC
C	E	‘C’	261	CE
E	A	‘E’	262	EA
A	B			
AB	EOL	256		

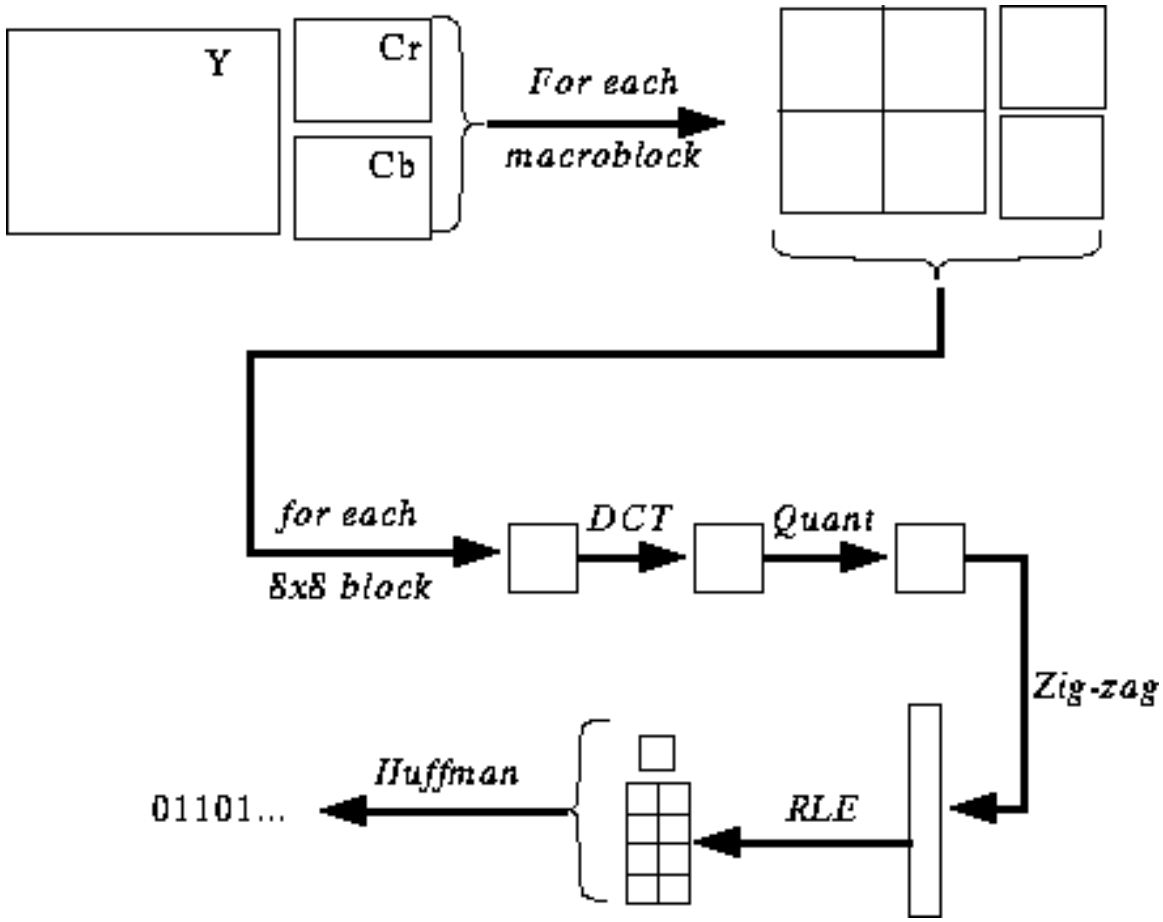
So Output is ‘A’ ‘B’ ‘C’ 257 256 ‘C’ ‘E’ 256

Where ‘A’,’B’ are the ASCII codes for A, B etc.

14 MARKS UNSEEN

2. (a) *Briefly outline the basic principles of Inter-Frame Coding in Video Compression.*

Essentially Each Frame is JPEG endcoded



- Macroblocks are 16x16 pixel areas on Y plane of original image.

  A ***macroblock*** usually consists of 4 Y blocks, 1 Cr block, and 1 Cb block.

- Quantization is by constant value for all DCT coefficients (i.e., no quantization table as in JPEG).

The Macroblock is coded as follows:

Addr	Type	Quant	Vector	CBP	b0	b1	...	b5

- Many macroblocks will be exact matches (or close enough). So send address of each block in image -> *Addr*

- Sometimes no good match can be found, so send INTRA block -> *Type*

- Will want to vary the quantization to fine tune compression, so send quantization value -> *Quant*

- Motion vector -> *vector*

- Some blocks in macroblock will match well, others match poorly. So send bitmask indicating which blocks are present (Coded Block Pattern, or *CBP*).

- Send the blocks (4 Y, 1 Cr, 1 Cb) as in JPEG.

8 Marks BOOKWORK

(b) *What is key difference between I-Frames, P-Frames and B-Frames? Why are I-frames inserted into the compressed output stream relatively frequently?*

I-Frame --- Basic Refernce FRAME for each Group of picture. Essentially a JPEG Compressed image.

P-Frame --- Coded *forward* Difference frame w.r.t last I or P frame

B-Frame --- Coded *backward* Difference frame w.r.t last I or P frame

I-frame Needed regularly as differences cannot cope with drift too far from rerence frame. If not present regularly poor image quality results.

6 Marks BOOKWORK

(c) *A multimedia presentation must be delivered over a network at a rate of 1.5 Mbits per second. The presentation consists of digitized audio and video. The audio has an average bit rate of 300 Kbits per second. The digitised video is in PAL format is to be compressed using the MPEG-1 standard. Assuming a frame sequence of:*

IBBPBBPBBPBBI…..

and average compression ratios of 10:1 and 20:1 for the I-frame and P-frame what is the compression ratio required for the B-frame to ensure the desired delivery rate?

You may assume that for PAL the luminance Signal is sampled at the spatial resolution of 352x288 and that the two chrominance signals are sampled at half this resolution. The refresh rate for PAL is 25Hz. You should also allow 15% overheads for the multiplexing and packetisation of the MPEG-1 video.

Desired Rate = 1.5 Mbits/Sec

Desired video rate = Rate – audio rate
= 1.5 – 0.3 = 1.2 Mbits/Sec

Physical rate = Video Rate less Headroom
= 1.2 / 1.15 = 1.044 Mbits/Sec

Each Group has 12 Frame: 1 I, 8 B and 3 P frames

So average frame rate = (0.1 + 3*0.05 + 8*x*)/12 = (0.25 + 8*x*)/12

Each frame has: 352*288*8 + 2*(176*144*8) bits (uncompressed) = 1,216,512 bits

So average Compressed bits per frame (average over 12 frames GoP) = 1216512*(0.25 + 8*x*)/12

Therefore Bits per second at 25 Frames per Sec rate=
25*1216512*(0.25 + 8*x*)/12

We require:

25*1216512*(0.25 + 8*x*)/12 = 1044000
2534400*(0.25 + 8*x*) = 1044000
(0.25 + 8*x*) = 0.412
8*x* = 0.16
x = 0.02

Or the compression ratio **is 50:1 for the B-FRAME**

13 MARKS UNSEEN

3. (a) *What key features of Quicktime have led to its adoption and acceptance as an international multimedia format?*

QuickTime is the most widely used cross-platform multimedia technology available today. QuickTime developed out of a multimedia extension for Apple's Macintosh(proprietry) System 7 operating system. It is now an international standard for multimedia interchange and is avalailbe for many platforms and as Web browser plug ins.

The following main features are:

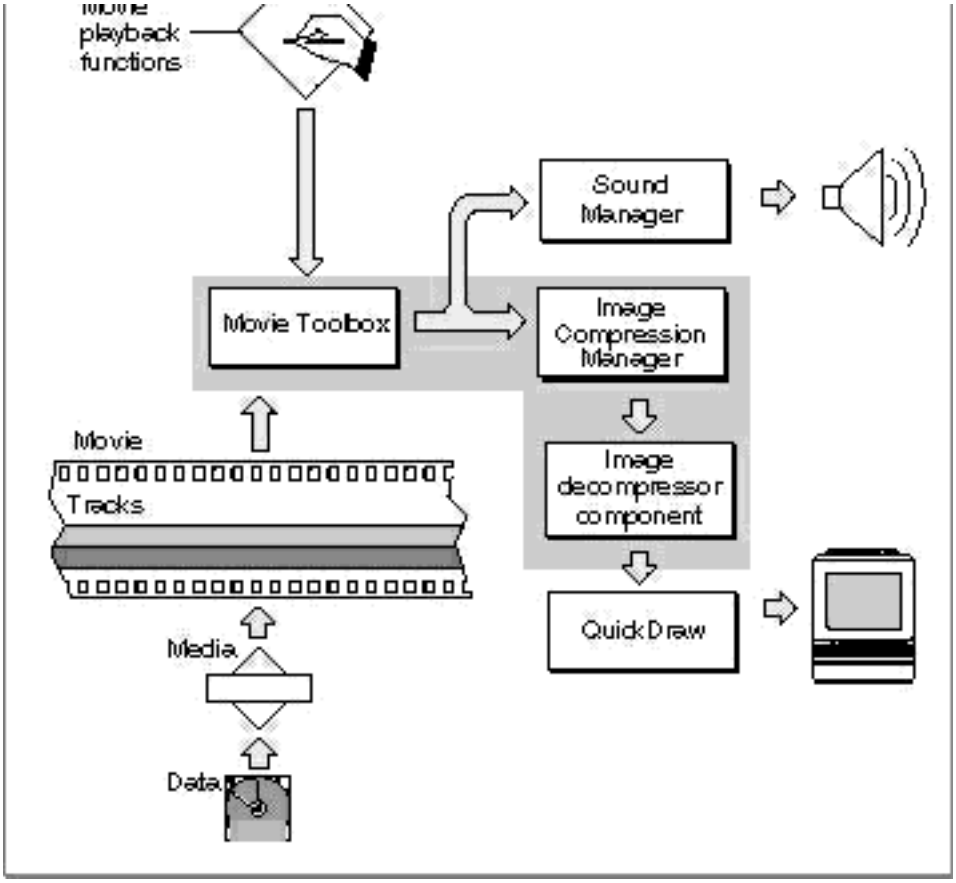
- Versatile support for web-based media
- Sophisticated playback capabilities
- Easy content authoring and editing
- QuickTime is an ***open standard*** -- it embraces other standards and incorporates them into its environment. It supports almost every major Multimedia file format

4 Marks – BOOKWORK

(b) Briefly outline the Quicktime Architecture and its key components.

The QuickTime Architecture:

QuickTime comprises two managers: the Movie Toolbox and the Image Compression Manager. QuickTime also relies on the Component Manager, as well as a set of predefined components. Figure below shows the relationships of these managers and an application that is playing a movie.



The Movie Toolbox

-- Your application gains access to the capabilities of QuickTime by calling functions in the Movie Toolbox. The Movie Toolbox allows you to store, retrieve, and manipulate time-based data that is stored in QuickTime movies. A single movie may contain several types of data. For example, a movie that contains video information might include both video data and the sound data that accompanies the video.

The Movie Toolbox also provides functions for editing movies. For example, there are editing functions for shortening a movie by removing portions of the video and sound tracks, and there are functions for extending it with the addition of new data from other QuickTime movies.

The Movie Toolbox is described in the chapter "Movie Toolbox" later in this book. That chapter includes code samples that show how to play movies.

The Image Compression Manager

--

The Image Compression Manager comprises a set of functions that compress and decompress images or sequences of graphic images.

The Image Compression Manager provides a device-independent and driver-independent means of compressing and decompressing images and sequences of images. It also contains a simple interface for implementing software and hardware image-compression algorithms. It provides system integration functions for storing compressed images as

part of PICT files, and it offers the ability to automatically decompress compressed PICT files on any QuickTime-capable Macintosh computer.

In most cases, applications use the Image Compression Manager indirectly, by calling Movie Toolbox functions or by displaying a compressed picture. However, if your application compresses images or makes movies with compressed images, you will call Image Compression Manager functions.

The Image Compression Manager is described in the chapter "Image Compression Manager" later in this book. This chapter also includes code samples that show how to compress images or make movies with compressed images.

The Component Manager
--

Applications gain access to components by calling the Component Manager. The Component Manager allows you to define and register types of components and communicate with components using a standard interface. A component is a code resource that is registered by the Component Manager. The component's code can be stored in a systemwide resource or in a resource that is local to a particular application.

> Once an application has connected to a component, it calls that component directly. If you create your own component class, you define the function-level interface for the component type that you have defined, and all components of that type must support the interface and adhere to those definitions. In this manner, an application can freely choose among components of a given type with absolute confidence that each will work.

QuickTime Components :

- movie controller components, which allow applications to play movies using a standard user interface standard image compression dialog components, which allow the user to specify the parameters for a compression operation by supplying a dialog box or a similar mechanism
- image compressor components, which compress and decompress image data sequence grabber components, which allow applications to preview and record video and sound data as QuickTime movies video digitizer components, which allow applications to control video digitization by an external device
- media data-exchange components, which allow applications to move various types of data in and out of a QuickTime movie derived media handler components, which allow QuickTime to support new types of data in QuickTime movies
- clock components, which provide timing services defined for QuickTime applications preview components, which are used by the Movie Toolbox's

standard file preview functions to display and create visual previews for files
sequence grabber components, which allow applications to obtain digitized data from sources that are external to a Macintosh computer

- sequence grabber channel components, which manipulate captured data for a sequence grabber component

- sequence grabber panel components, which allow sequence grabber components to obtain configuration information from the user for a particular sequence grabber channel component

10 Marks BookWork

(c) JPEG2000 is a new image compression standard. Outline how this new standard might be incproprated into the Quicktime Architecture. Your answer need not consider the details of the actual compression methods used in JPEG2000 , instead it should focus on how given the compression format you could extend Quicktime to support it.

Sketch of ideas required by solution builds on QT Architecture knowledge above

JPEG is a still image format Need to add functionality to the following

Media Data Structure --- add knowledge of data structure on new format
Component manager --- add new component to component manage
Image Compression --- add compression **and** decompression routines to Compression manage

13 MARKS UNSEEN

4. (a) When designing multimedia systems, what two levels of functionality need to be considered? Briefly define these levels.

There are two levels of description for an application:

* ***Functional level*** -- what the application does and source and display data formats. ***E.g.*** An application may deal with the retrieval and display of movies. It retrieves movies and displays them in a window.
* ***System Level*** -- The entities that the application deals with and how it deals with them. In the above example we deal with video frames and voice samples and need to process these accordingly.

4 Marks Bookwork

(b*) For each common multimedia data type discuss what common functionalities should be supported by a multimedia system.*

The following functionality should be provided by a multimedia system:

- Digital Representation of Media -- Many standardised formats for many media
- Capture: Digitisation of Media -- special Hardware/Software
- Creation and editing -- assemble media and alter it
- Storage Requirements -- significant for multimedia
- Compression -- related to above and below, ie can save on storage but can hinder retrieval
- Structuring and retrieval methods of media -- simple to advanced DataBase Storage
- Display or Playback methods -- effect of retrieval must view data
- Media Synchronisation -- display multimedia as it is intended

Typical Range of operations required for common media include:

Text

--

- Editing
- Formatting
- Sorting
- Indexing
- Searching
- Encrypting

The above operations involve basic processing such as:

- Character Manipulation
- String Manipulation

Audio

--

- Audio Editing
- Synchronisation
- Conversion/Translation
- Filtering/ Sound Enhancing Operators
- Compression
- Searching
- Indexing

The above operations involve basic processing such as:

- Sample Manipulation
- Waveform Manipulation

Graphics

--

- Graphic primitive Editing
- Shading

- Mapping
- Lighting
- Viewing
- Rendering
- Searching
- Indexing

The above operations involve basic processing such as:

- Primitive Manipulation
- Structural/Group Manipulation

Image

--

- Pixel operations
- Geometric Operations
- Filtering
- Conversion
- Indexing
- Compression
- Searching

The above operations involve basic processing such as:

- Pixel Manipulation
- Area Manipulation, copy image areas, delete area, process areas of interest

Animation

--

- Primitive/Group Editing
- Structural Editing
- Image Overlaying
- Rendering
- Synchronisation
- Searching
- Indexing

The above operations involve basic processing such as:

- Pixel Manipulation
- Area Manipulation, copy image areas, delete area, process areas of interest
- Frame Manipulation, Copy frame(s) to frame(s), Process frame(s)

Video

--

- Pixel Operations
- Frame Operations
- Editing
- Synchronisation
- Conversion
- Mixing
- Indexing
- Searching
- Video Effects/Filtering

The above operations involve basic processing such as:

- Pixel Manipulation
- Area Manipulation, copy image areas, delete area, process areas of interest
- Frame Manipulation, Copy frame(s) to frame(s), Process frame(s)

8 marks Bookwork

(c) *You have been commissioned to produce a Multimedia mail system. What media should be supported in such a mail system and how should an application facilitate assembly, delivery and reading of the mail.*

SKECTH OF SOULTIOIN REALISATION OF BANDWIDTH/MESSAGE SIZE and how deal with it important. Also how the application deals with media is important.

Multimedia mail an extension of test-only mail:

- Basic content still test
- Annotated with richer media:
  - Image
  - Speech/Audio
  - Video Clip

Possible problems

- Send all data at once ---- large mail message
  - High Bandwidth
  - High Server Overheads
  - Mailboxes need large storage
  - Mail apps need large memory
  - Low quality audio and images not a problem if small in number?
- Terminal being used the read mail supports all media?
- Printing etc may lose their Message

Solution Annotations (tags/links) in main body

- Annotations can be sent with main message (Explicit Media Inclusion)OR
- Get requested specifically by recipient when message is read or indeed if media is required.(Implicit Media Inclusion)
- Latter method allows reading of mail on more platforms etc
- Could send previews of data (i.e. lower resolution audio, images, image snapshots of video, thumbprints of larger media?)

Application Requirements

- Multimedia enabled
- Good user friendly Interface
- Good network connection
- Good media support of formats
- Input ---- support for Multimedia Authoring
- Input ---- support for Graphics/Image/Audio/Video input devices

15 Marks Unseen