SCHOLAR Study Guide

SQA Higher

Computing Unit 3c
Multimedia Technology

David Bethune
Heriot-Watt University

Andy Cochrane
Heriot-Watt University

Tom Kelly
Heriot-Watt University

Ian King
Heriot-Watt University

Richard Scott
Heriot-Watt University

Interactive University

Edinburgh EH12 9QQ, United Kingdom.

First published 2004 by Interactive University

This edition published in 2006 by Interactive University

SCHOLAR is a programme of Heriot-Watt University and is published and distributed on its behalf by Interactive University.

British Library Cataloguing in Publication Data

Interactive University

SCHOLAR Study Guide Unit 3: Computing

1. Computing

ISBN 1 904647 37 5

Typeset by: Interactive University, Wallace House, 1 Lochside Avenue, Edinburgh, EH12 9QQ.

Printed and bound in Great Britain by Graphic and Printing Services, Heriot-Watt University, Edinburgh.

Part Number 2006-1424

Acknowledgements

Thanks are due to the members of Heriot-Watt University's SCHOLAR team who planned and created these materials, and to the many colleagues who reviewed the content.

Programme Director: Professor R R Leitch

Series Editor: Professor J Cowan

Subject Directors: Professor P John (Chemistry), Professor C E Beevers (Mathematics), Dr P J B King (Computing), Dr P G Meaden (Biology), Dr M R Steel (Physics), Dr C G Tinker (French)

Subject Authors:

Biology: Dr J M Burt, Ms E Humphrey, Ms L Knight, Mr J B McCann, Mr D Millar, Ms N Randle, Ms S Ross, Ms Y Stahl, Ms S Steen, Ms N Tweedie

Chemistry: Mr M Anderson, Mr B Bease, Dr J H Cameron, Dr P Johnson, Mr B T McKerchar, Dr A A Sandison

Computing: Mr I E Aitchison, Dr P O B Holt, Mr S McMorris, Mr B Palmer, Ms J Swanson, Mr A Weddle

Engineering: Mr J Hill, Ms H L Jackson, Mr H Laidlaw, Professor W H Müller

French: Mr M Fermin, Ms B Guenier, Ms C Hastie, Ms S C E Thoday

Mathematics: Mr J Dowman, Ms A Johnstone, Ms O Khaled, Mr C McGuire, Ms J S Paterson, Mr S Rogers, Ms D A Watson

Physics: Mr J McCabe, Mr C Milne, Dr A Tookey, Mr C White

Learning Technology: Dr W Austin, Ms N Beasley, Ms J Benzie, Dr D Cole, Mr A Crofts, Ms S Davies, Mr A Dunn, Mr M Holligan, Dr J Liddle, Ms S McConnell, Mr N Miller, Mr N Morris, Ms E Mowat, Mr S Nicol, Dr W Nightingale, Mr R Pointon, Mr D Reid, Dr R Thomas, Dr N Tomes, Ms J Wang, Mr P Whitton

Cue Assessment Group: Ms F Costigan, Mr D J Fiddes, Dr D H Jackson, Mr S G Marshall

SCHOLAR Unit: Mr G Toner, M G Cruse, Ms A Hay, Ms C Keogh, Ms B Laidlaw, Mr J Walsh

Media: Mr D Hurst, Mr P Booth, Mr G Cowper, Mr C Gruber, Mr D S Marsland, Mr C Nicol, Mr C Wilson

Administration: Ms L El-Ghorr, Dr M King, Dr R Rist,

We would like to acknowledge the assistance of the education authorities, colleges, teachers and students who helped to plan the SCHOLAR programme and who evaluated these materials.

Grateful acknowledgement is made for permission to use the following material in the SCHOLAR programme:

To the Scottish Qualifications Authority for permission to use Past Papers assessments.

The financial support from the Scottish Executive is gratefully acknowledged.

Contents

Topic 1

Development process for multimedia applications

Contents

Prerequisite knowledge

Before studying this topic you should have:

- *Familiarity using a text editor (e.g. notepad)*
- *Basics of creating a presentation (e.g. PowerPoint)*
- *Basics of creating a web site (e.g. FrontPage)*

All topics require:

- *familiarity in working with the computer filing system - i.e. the ability to create, view the properties of, save and open files and folders;*

- *the ability to locate and run the necessary software for each topic;*
- *the ability to switch between multiple applications on the computer system;*
- *knowledge of units of storage (bits, Bytes, KB, MB);*
- *knowledge of the basic components of computer systems (mostly just input, output and backing store devices).*

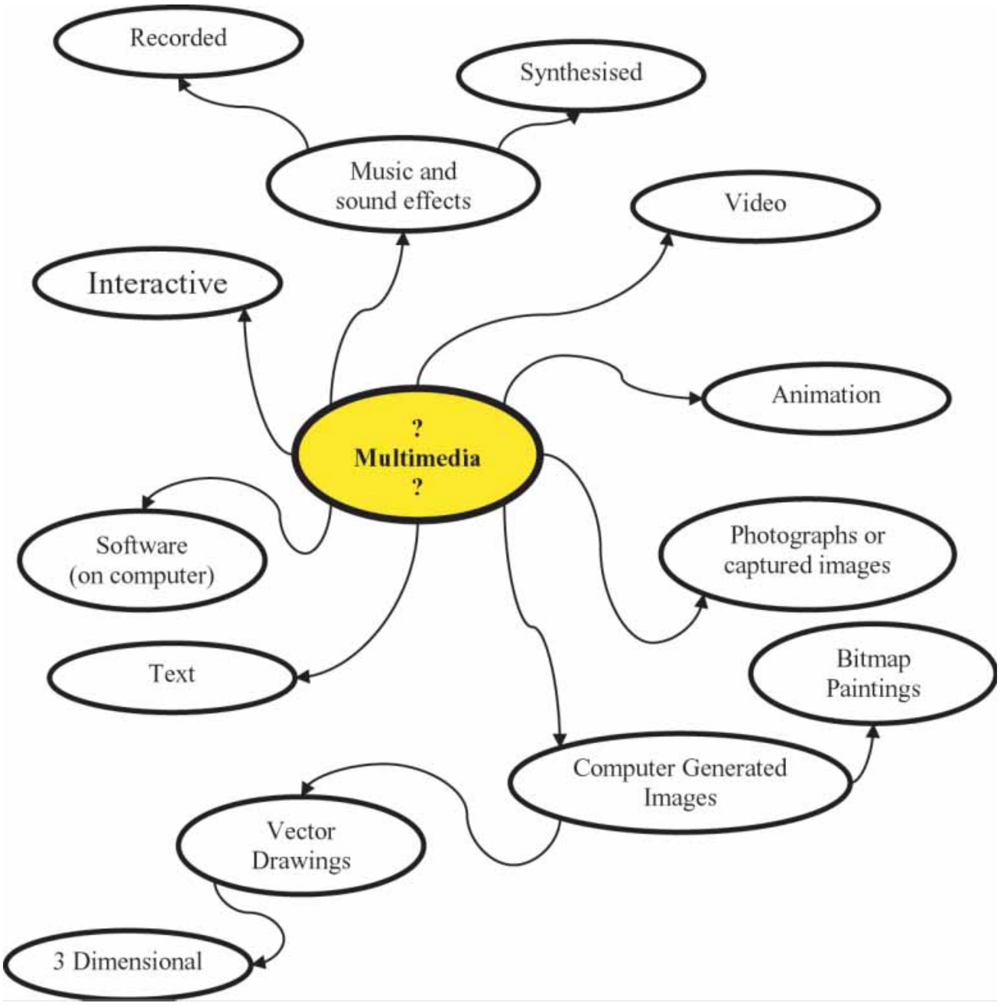
Learning Objectives

By the end of this topic you should be able to:

- *Describe the stages of development for multimedia applications;*
- *Compare WYSIWYG and Text editors;*
- *Compare Presentation and multimedia authoring software;*
- *Describe the hardware and software requirements for displaying multimedia;*
- *Give descriptions of the terms codecs and container files.*

1.1 Introduction: What is a multimedia application?

A multimedia application might be defined as an interactive piece of software communicating to the user using several media, for example text, images (photographs, illustrations), audio (music, sounds), video and animation.



Multimedia data is either:

- captured from real life (digitised) or created in the computer (synthesised)
- a bitmap (discrete) type of data or a vector (object oriented) type of data
- static (like an image) or time-dependent (like a video)

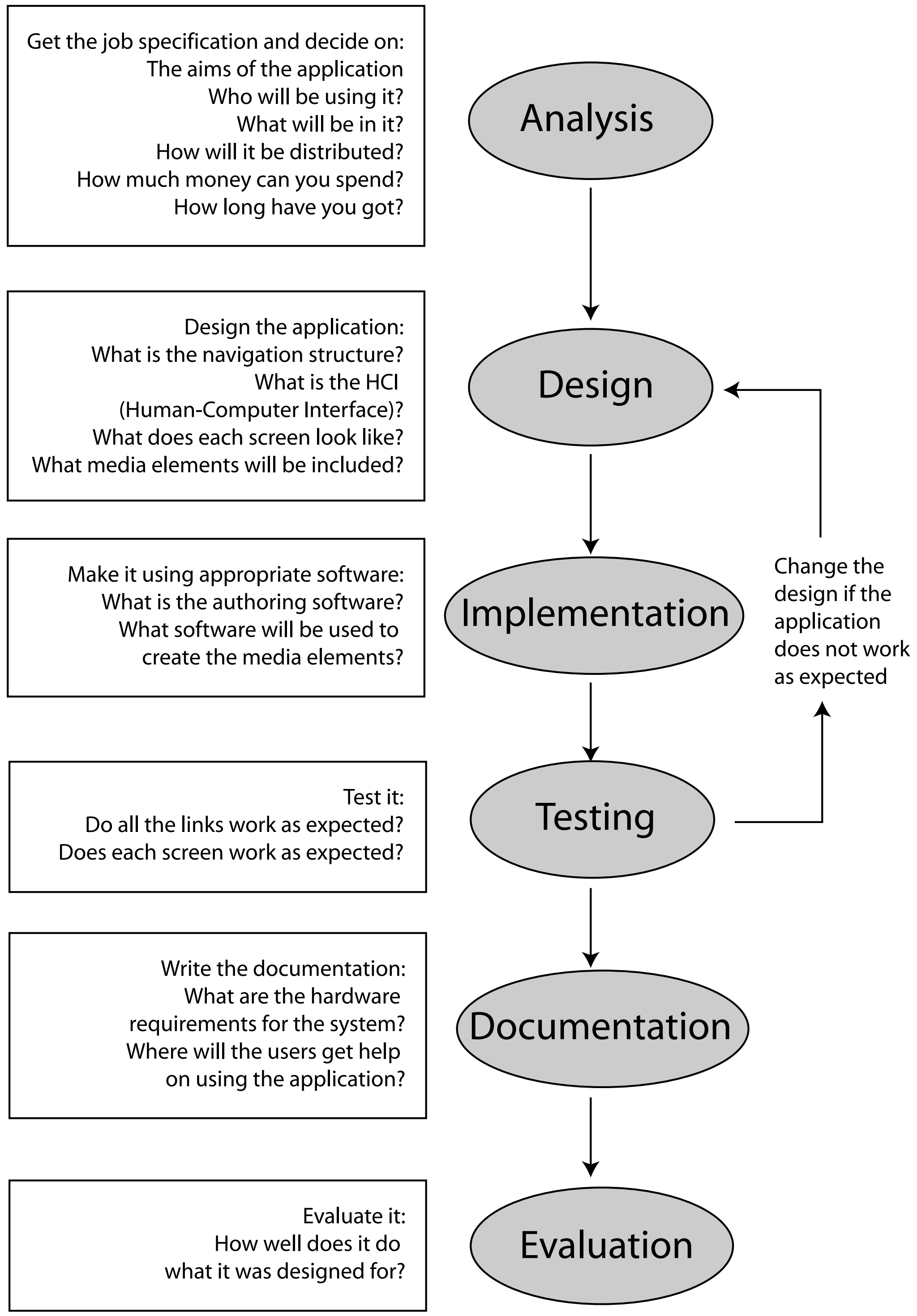
You should be able to apply 3 of these terms to every data type.

1.2 Description of the software development process

Multimedia applications range from simple linear slide show presentations to complex multimedia games or large, sophisticated web sites.

Whatever the target end-product, good planning and careful following of these stages of development are essential to a high-quality, successful outcome, delivered on time to your customer.

The software development process is as follows:



1.3 Methodologies used in the creation of a multimedia application

1.3.1 WYSIWYG and text editors

Software that is used to construct multimedia applications are usually either WYSIWYG ("what you see is what you get") or text editors.

WYSIWYG

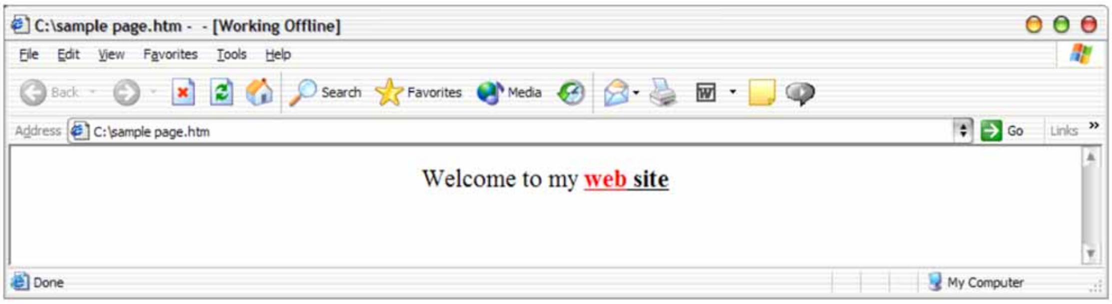
These allow the application to be viewed as it will finally be seen while it is being edited.

Text editors

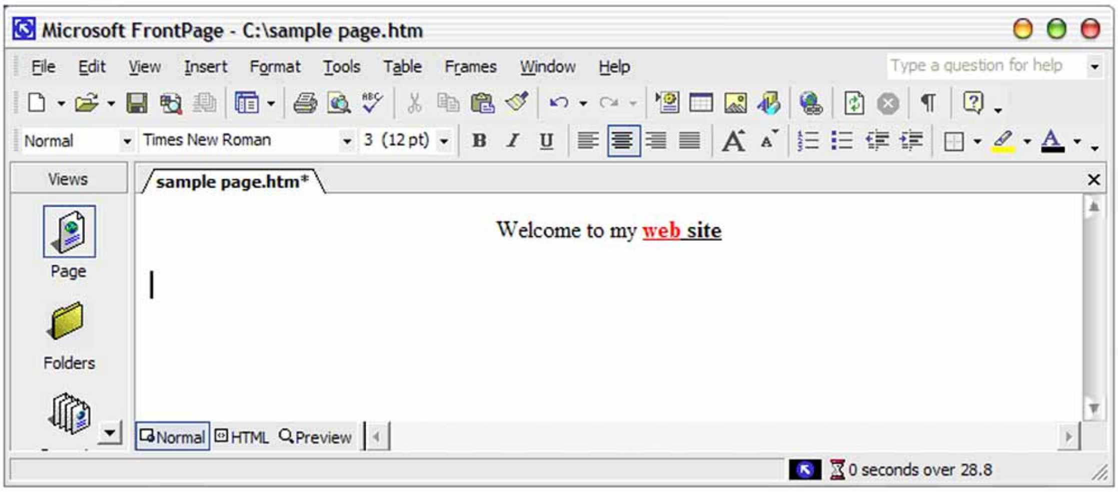
The application has to be structured using plain text.

Consider the following...

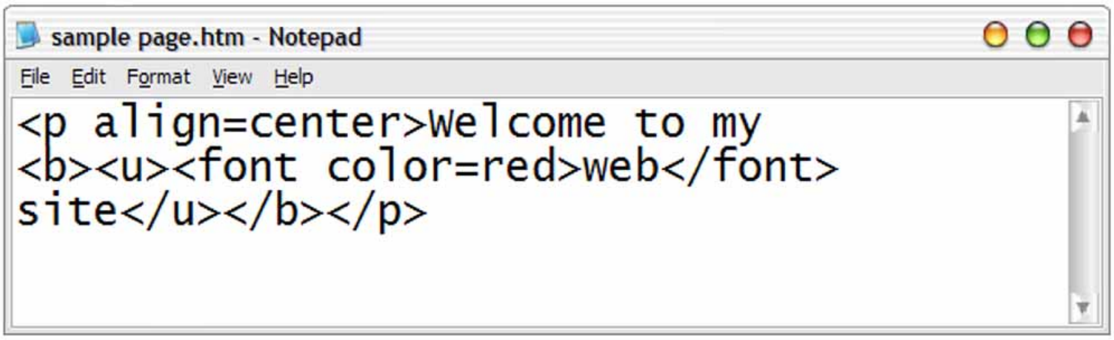
Viewed in a web browser:



Being edited in a WYSIWYG editor:



Being edited in a text editor:



Here is a short sample of selected software that can be used to create multimedia applications:

Software	**WYSIWYG, Text editor or both?**
Microsoft Word	Both
Microsoft PowerPoint	WYSIWYG
Microsoft FrontPage	Both
Notepad	Text editor

1.3.1.1 Advantages and disadvantages

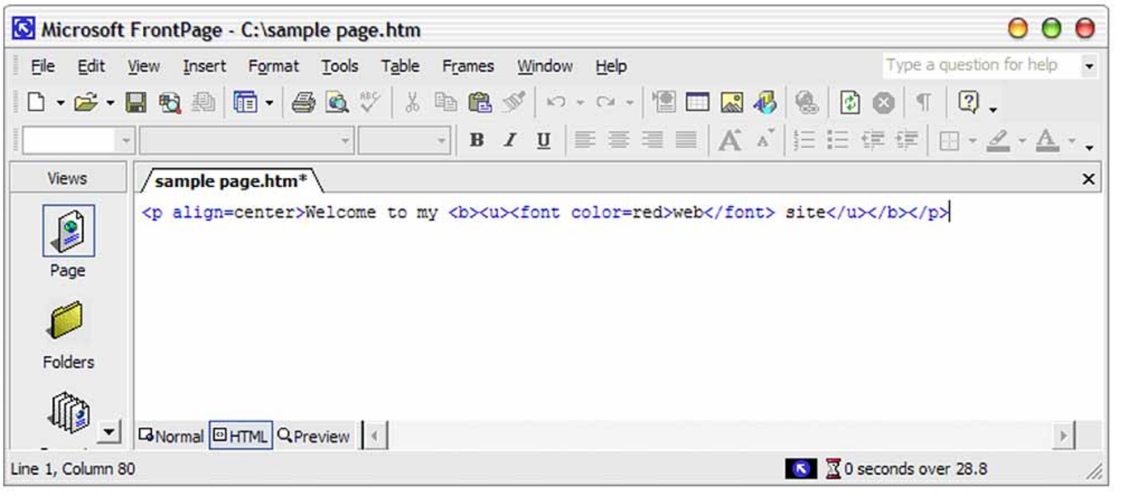
Why do the top authoring programs provide both text editing and WYSIWYG abilities? This is because there are advantages to both...

	Advantages	**Disadvantages**
Text Editor	Text editing software is simple, placing few demands on the computer hardware Provides greater control over the application	Output can be difficult to predict or calculate Knowledge of programming or text-based commands is required
WYSIWYG	Allows for easier editing of the application	More complex software places greater demands on the computer hardware

1.3.1.2 Applications providing both text and WYSIWYG editing

In practice having both capabilities is best. The WYSIWYG editor can be used to structure the HCI of the application but extra functionality and fine-tuning can then be carried out using the text editor.

Here is an editor shown in WYSIWYG mode above, but now switched to text mode. Note the colour coding - this is often applied by text editors designed for this purpose to help make the text more readable.



Comparing WYSIWYG with text editing

Use a text editor (e.g. notepad) to enter this into a new file, saving it as "page1.htm":

```
<html>
<head>
<title>BCS Definition of Multimedia</title>
</head>

<body bgcolor=Yellow>
<p align="center"><font color=Red size="6" face="Arial">Multimedia
</font></p>
<p><font face="Times New Roman">The BCS (British Computer Society)
definition of multimedia is:</font></p>
<p align="center"><font face="Times New Roman" color=Blue><em>
’Multimedia is the presentation of information by a computer system
using graphics, animation, sound and text.’</em></font></p>
</body>
</html>
```

Use a web browser (e.g. Internet Explorer) to open up the file and view it.

Now try to recreate exactly the same page using a WYSIWYG editor e.g. MS FrontPage or MS Word (can create web pages), save this as "page2.htm".

Check how this looks in the web browser as well.

Edit both files so that the background is Black and alter the colour of the rest of the text so that it is all readable.

Add the following text below the actual definition:

"from A Glossary of Computing Terms by the BCS, 10th Ed, page 85"

Make this new text 'Arial' font, right-aligned, blue and in italics.

1.3.2 Authoring software

1.3.2.1 What is authoring software?

Multimedia authoring software allows the developer to create an interactive, multimedia application that can be delivered to people using a variety of mediums, e.g. through web pages, on CDROMs or DVDs, kiosk terminals in public places or through a program that you run on your own computer system (be it a PC, games console or even a mobile phone). Entire computer games are regularly created using authoring tools and some authoring tools have been specifically developed for the creation of games or game content.

It provides the ability for the developer to customise the application using a variety of methods, e.g. using program code or scripting, or using an icon-based method to set up complex properties without having to learn programming code.

1.3.2.2 Who makes multimedia authoring software?

Macromedia

- Flash - started out as an animation program, but now with the ability to incorporate many media types, provide complex user interaction and with support for scripting is now a feature-rich and popular multimedia authoring package in its own right.
- Dreamweaver - a website authoring and management tool
- Director - this is used for the creation of multimedia applications to be used, for example, in DVDs, CDROMs and kiosk applications. It can also create multimedia content for web pages by creating Shockwave objects.

Microsoft

- FrontPage - for website creation and management

Adobe

- GoLive - for website creation and management

Others

There are many, many other manufacturers producing multimedia authoring software, from full commercial programs, as those listed above, to shareware and freeware solutions. For example, *Mediator*, *Blender* and *HyperStudio* are just three more.

A quick search on the Internet for 'multimedia authoring software' provides a list of numerous authoring programs that are available.

1.3.3 Presentation software

1.3.3.1 What is presentation software?

Presentation software allows the developer to create a multimedia application that will deliver content to an audience. Originally a presentation consisted solely of screens (or slides) with text or images that moved through them from the first to the last (a linear presentation), as a visual aid to assist someone while giving a presentation. There is still presentation software that only allows this type of simple presentation to be created. We are more concerned with the more modern type of presentation software and the range of abilities it has developed as it has evolved.

Although modern presentation software is more feature-rich than the original versions of the programs, they are still very limited when compared to 'proper' multimedia authoring software.

At a basic level, presentation software allows the developer to create presentations consisting of screens (or slides) that contain multimedia data, allow interaction with the user and can be 'played' or viewed in a non-linear manner. Most packages will also allow the presentation to be saved in a variety of forms that allow it to be viewed in a variety of situations, e.g. as a stand-alone file that does not need the presentation software or as a web page, or web site (series of related web pages).

1.3.3.2 Who makes presentation software?

Microsoft

- PowerPoint - part of the ubiquitous Microsoft Office suite of programs. This has evolved almost to the point of being an authoring package in its own right. It can incorporate a wide variety of media, provide complex user-interaction and be further extended through the use of scripting.

Others

There are a few other manufacturers of presentation software, most with far fewer features than PowerPoint. A quick search on the Internet for 'multimedia presentation software' will turn up a list of a few others that can be investigated.

1.3.4 Presentation and authoring software compared

	Presentation Software	**Authoring Software**
Advantages	Lack of complexity means it is easy to learn. Lack of complexity allows simple presentations to be created quickly.	Advanced features give the developer more control over the final application.
Disadvantages	It is not possible to extend the features of the application beyond the simple set of those provided by the software.	The learning curve is much steeper as there are many more features to learn about. It can be more difficult to debug a complex application that is not working properly.

Create a multimedia application

Follow 3 of the stages of the Software development process to complete this task, omitting ’Analysis’, ’Documentation’ and ’Evaluation’.

You should create two versions of your completed design - one using authoring software, the other using presentation software.

You do not have long to complete this task, so keep your designs SIMPLE.

The purpose of this task is not to produce fabulous, complete multimedia applications, but to use both a Presentation and Authoring package to complete the same task. This should allow you to compare both programs.

Your brief:

Create a multimedia application to introduce people to you and two of your interests.

You must complete this task using only 3 screens or pages.

Each screen must not contain any more than 2 images (although you can use a background).

Your system should be easy to navigate, so design the navigation method carefully.

Step 1 Design

- Design the navigation structure, showing how all 3 pages link together. Design the rough layout for each page, mentioning the content and navigation features.

Step 2 Implementation

- Create your application using Presentation software.

Step 3 Testing

- Make sure that all the links and navigation features work. Make sure that each page is displayed/animated as you wish.

Step 4

- Repeat Steps 2 & 3, this time using authoring software.

1.4 Display of multimedia data

1.4.1 Distribution technologies

Multimedia applications can be distributed using these technologies:

- A web page, or series of web pages
- Embedded into a web page using any of the following technologies:
  - ActiveX object
  - Visual basic script
  - Java applet
  - Javascript program
- A stand-alone program, that is, an executable file
- A multimedia file run using player software e.g.:
  - Macromedia Flash or Shockwave
  - Microsoft PowerPoint presentation
  - Apple QuickTime movie
- ...or one of many others
- ...or a combination of two or more of the above!

1.4.2 Distribution medium

Web pages are distributed or accessed through the WWW on the Internet or through other networks e.g. a local area network within a business (LAN - Local Area Network) or a larger private network between two buildings several miles apart (WAN - Wide Area Network). Web pages are accessed through connections of increasingly greater bandwidth, this greater bandwidth means that more multimedia data can be used.

Mediums in popular use for distributing multimedia applications:

- Web pages on the Internet
- Web pages on Intranets
- CDROM discs
- DVD discs
- Digital broadcast systems

1.4.3 Embedded data

Multimedia applications usually require large data files for sound, images, video and so on. These data files are either stored along with the program file or are combined (embedded) into the actual program itself.

Web pages store the data files separately, which allows a web page to be viewed and read, even if all the images (for example) have not been loaded.

Many executable multimedia applications, for example games, also store the data separately. It is this data which is copied to the computer when a full install of the games is performed.

Multimedia applications which have the multimedia data files built into a single file are said to have the data embedded. This way there is only a single file to manage, but it may be large and the individual data files will not be accessible on their own - although this is what the developer may want.

For example Microsoft PowerPoint presentations embed the data into a single file, whereas web pages download every media element (pictures, sounds and so on) as separate files.

There are advantages and disadvantages to embedding data in the file:

Advantages	**Disadvantages**
File management is easier as there is just one file to worry about. It is easier to keep data secure to prevent access to the files, preventing others from easily copying them.	The file could be large so transferring it between computers could have problems. Often the entire file must be downloaded before it can be played. Individual data items can be updated without having to edit the actual application if the data is not embedded.

1.4.4 Streamed data

Data which is broadcast over a network is said to be streamed. Many radio stations around the World broadcast the radio sound over the Internet.

To access streamed data, you need the client software (Real Player or Windows Media Player are just two) and the data is played as it is received - you do not need to download an entire file before viewing it.

Viewing on-demand systems will allow user to purchase films and to then view them as the film is streamed to them - but high quality video needs faster network connections (greater bandwidths) than most people currently have.

1.4.5 Requirements for viewing multimedia applications

In order to be able to access a multimedia application certain requirements may have to be met (depending on the media and distribution technology):

- **Stand alone player software**

  (for example Flash Player, Apple QuickTime or PowerPoint Player)

  Player software is usually available to download for free from the manufacturers web site. This allows the multimedia application to be viewed, run or played but not edited

- **Browser Plug-in**

  The correct Plug-in may have to be installed in the web browser if the multimedia file is included within a web page. The player software mentioned above can usually also be installed as a plug-in for the web browser to allow the application to be run from within the web page.

- **Hardware and software requirements**

  If running a stand-alone file or executable multimedia application then there may be minimum hardware and software requirements that have to be met.

- **Client software**

  If receiving streamed media over a network then the client software must be installed on the receiving computer.

1.5 Codecs and container files

1.5.1 Codecs

Multimedia data is stored in a data file. The structure of the file varies with the type of file. For the application to be able to display the multimedia data it must be able to 'decode' the file. In other words it requires the relevant 'codec' that contains the method required to decode that specific type of data file. The codec can also be used to create that type of data file, or to be able to code that type of file.

Codec : COde DECode

Throughout this entire unit we will examine codecs for many types of data file. The main purpose of most codecs is to reduce the size of a file while keeping as much quality as is needed. (Sometimes codec is said to be shorthand for Compressor / Decompressor).

Codecs are often built into programs, but can often be downloaded as an add-in for many programs.

1.5.2 Container file

Data files on PCs have a suffix that is used to determine the file type and thus to work out how to display the file. Apple Macintosh computers have a similar system.

But quite often the data in the file could be any one of a number of types of data or it could be data that could be encoded in any of a number of ways.

Container files are files that can contain a variety of data or that the data could be encoded in a variety of ways.

So how does the application know how to handle a particular data file? All container files have a 'header' at the start of the file that gives information about the type of data and codec used.

A good example of container files are Microsoft RIFF multimedia files. This is not one file format but many. AVI (Audio video interleave) and Wav (wave) files are both types of RIFF file. Although we know from the file extension that an AVI file contains a video, it could be encoded in any number of ways, and although we may already be able to play some other AVI files with that computer, we may need to install a new codec to allow that new AVI file to be played.

1.6 Questions

Q1: Which of these is **not** a multimedia application?

a) An interactive page on Digital Television
b) A Hollywood block-buster film
c) A new rally racing game for a PlayStation
d) A new 'hole-in-the-wall' cash-line machine with colour display

Q2: Give two reasons to explain your answer.

Q3: Which of these describes part of the 'Design' stage of the software development process (SDP)?

a) Deciding who the audience is going to be
b) Deciding if this was the best way to make this application or if things could have been improved if done any differently
c) Deciding the layout of each page of the application
d) Creating an on-line help file to include with the application

Q4: What stages in the SDP do the other answers belong to?

Q5: Which of these describes an advantage of using Authoring software instead of Presentation software to create a multimedia application?

a) It is more flexible, allowing greater control over what happens in the application
b) It is easier to learn allowing applications to be developed more quickly
c) It has a WYSIWYG layout allowing you to edit the application and view it exactly as it will appear when being run
d) It has wizards and tools to allow the application to be developed more quickly

Q6: Describe all the *disadvantages* of using Authoring software instead of Presentation software to create a multimedia application.

Q7: A web user wished to listen to a radio station over the Internet while 'surfing'. What extra software is needed on their computer in addition to the 'net' surfing software?

a) web browser
b) presentation player
c) WYSIWYG multimedia editor
d) The client software for receiving the streamed broadcast

Q8: Name 2 programs that can receive streamed data from the Internet. Do these programs have any other capabilities?

Q9: New 'codecs' are being developed all the time. What is a likely reason for this?

a) New codecs are needed to work with all the new types of data that are coming out
b) New codecs are needed when new computer hardware is developed
c) Old codecs stop working after a while
d) New codecs are being developed that are better at compressing the data

Q10: Find the name of codecs used with: audio and video data.

Topic 2

Bit-mapped graphic data

Contents

Prerequisite knowledge

Before studying this topic you should have:

- *Familiarity using bitmap creation programs (e.g. Paint and Photo Editor)*
- *Familiarity with binary numbers*

All topics require:

- *familiarity in working with the computer filing system - i.e. the ability to create, view the properties of, save and open files and folders;*

- *the ability to locate and run the necessary software for each topic;*
- *the ability to switch between multiple applications on the computer system;*
- *knowledge of units of storage (bits, Bytes, KB, MB);*
- *knowledge of the basic components of computer systems (mostly just input, output and backing store devices).*

Learning Objectives

By the end of this topic you should be able to:

- *Describe the hardware required for capturing bitmap images;*
- *Describe the factors that affect the quality of the captured image;*
- *Compare four common bitmap image file formats;*
- *Demonstrate an understanding of bitmap images at a technical level;*
- *Describe common compression schemes;*
- *Use the terms 're-sampling', 'dithering' and 'anti-alias' correctly;*
- *Perform calculations involving the size of bitmap image files;*
- *Describe the hardware required for the display of bitmap images.*

2.1 Introduction

Images files which store the data needed to recreate an image dot-by-dot (pixel by pixel) are bitmap images. All images we capture from the 'real world' are bitmaps, for example from a digital camera or scanner.

Resolution

The term 'resolution' has several, all slightly different, meanings in Multimedia and is used throughout. So let's sort some of these out before we continue.

1. **The resolution of a printer:**

   the number of (microscopic) dots of ink that the printer can print in a single inch (dot pitch).

   Thus a printer with a resolution of 720 dpi (dots per inch) is capable of producing more detailed looking images than a printer only capable of 300 dpi

2. **The (optical) resolution of a digital camera:**

   the number of pixels of a captured image.

   A 3MP (Megapixel) camera will capture images consisting of around 3 million pixels, these images will not be as detailed as images captured using an 8MP camera. This is determined by the CCD in the camera.

   The software in the camera usually allows the actual captured resolution to be decreased if this maximum amount of detail is not needed.

3. **The (optical) resolution of a scanner:**

   the number of pixels that will be generated for every inch of the scanned document, measured in pixels per inch in both directions. This determines how much detail can be picked up from the image.

   The software allows the actual scanning resolution to be changed - altering the dimensions of the image captured, but is limited to the maximum physical (optical) resolution of the scanner.

   For example a scanner with a resolution of 2400dpi x 1800dpi scanning a 2 inch by 1 inch rectangle would produce an image of size (2 x 2400) x (1 x 1800) = 4800 x 1800 pixels

4. **The resolution of a computer display:**

   One of two meanings:

   - the physical resolution of a computer display. This refers to the maximum quality of the display and is given in two different ways:
     - the number of pixels in one inch, more usually given as the distance (or pitch) between each pixel. This has become known as dot pitch rather than resolution. Thus a display with a dot pitch of 0.24mm will be able to display finer images than a display with a dot pitch of 0.294mm, as the pixels are smaller and closer together.
     - the maximum number of pixels that the display is capable of showing. For example a display with a resolution of 1920 x 1440 pixels will be capable of displaying more detailed images than a display with a resolution of 1280 x 1024 pixels.

Note: This meaning must be considered together with the dot pitch when deciding on the quality of a display.

- The resolution of the current display settings. The computer software allows the computer display to be set at a number resolutions, determined by the graphics card. On a display with the current display resolution set to 1920 x 1200, images and text will appear smaller and more detailed than if the display was set at a resolution of 1280 x 800 pixels.

5. **The (spatial or screen) resolution of an image:**

   the dimensions or size of an image in pixels. E.g. to decrease the screen resolution of an image would reduce the number of pixels used to store it

6. **The (printing) resolution of an image:**

   the dot pitch to be used when printing an image. When taken in conjunction with the resolution of the image, the size of the printed image can be calculated. For example an image with a resolution of 900 x 600 pixels and printing resolution of 300dpi (or 300ppi - pixels per inch) will be 3 inches wide by 2 inches tall when printed

7. **The brightness resolution of an image:**

   another term for the colour depth of a bitmap image. This defines the dynamic range of a bitmap image - in other words, the colour range it can display.

8. **The sampling resolution of a sound:**

   the number of bits used when capturing each sample. This defines the dynamic range of the captured sound. *See Topic 3 on digitised sound data for more information.*

It is usually possible to tell from the context what meaning of resolution is meant. It has become fairly common to have the resolution of an image mentioned when actually meaning the screen resolution or dimensions of the image, as just 'resolution of an image' has traditionally meant the printing resolution. By looking at the units used we can see that if just pixels are mentioned (e.g. 800 x 600 pixels) then screen resolution is meant, but if a dot pitch is given (e.g. 72 pixels per inch or 28 pixels per cm) then printing resolution is meant.

2.2 Input (capture) hardware

2.2.1 CCDs

Devices which capture graphic data as an image produce a bitmap image. The sensors that capture the images are usually **Charge Coupled Devices** (CCDs). Although **CMOS** based sensors are a cheaper alternative, they are less effective. Cheap key-ring digital cameras and similar devices will usually use CMOS sensors.

These have a grid of sensors that give off an electrical signal when light is shone on them. Each sensor is actually a tiny photodiode.

The brighter the light, the bigger the signal. Each CCD sensor can (currently) only sense light levels, not colour.

There are several methods used to capture the colour of the light. Most of these record the image in RED, GREEN and BLUE, combining the three (just as in your computer monitor or television) to produce a full colour image.

2.2.2 Digital camera

Digital cameras have a rectangular array of CCD sensors to capture the image that is projected onto them by the camera lens.

2.2.2.1 Resolution of a digital camera

While the number of CCD sensors is usually greater than the actual number of recorded pixels (some of the sensors are not used for sensing light, but routing the electrical signals), we can assume that each CCD sensor records the data for a single pixel, although if you examine the actual specifications from the camera maker (the number of 'Megapixels' and the actual resolution of the image produced) you can see this discrepancy for yourself.

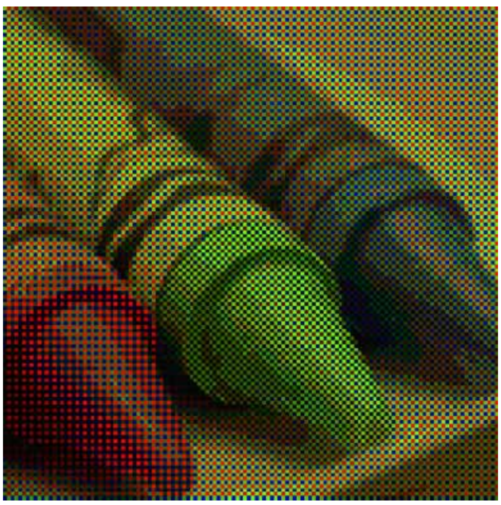
The resolution of a digital camera is the number of pixels in the image produced. The exact resolution is the width and height of the image in pixels. For example a 5.0 Megapixel camera (i.e. it has 5 000 000 light sensors on the CCD array) may produce an image of size 2560 pixels wide by 1920 pixels tall, a total of 4 915 200 pixels.

When the electrical signal is received from the CCD it needs to be turned into a digital signal by the ADC (Analogue to Digital Converter).

2.2.2.2 Capturing colour with a digital camera

The most common techniques for capturing colour involves placing a colour filter array over the CCD array, so that each sensor records light of one colour only, red, green, or blue.

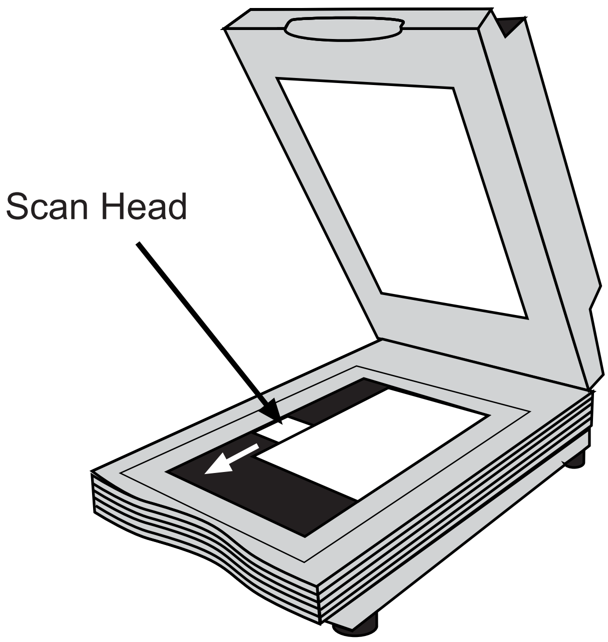
Here is an image as it is recorded directly from the CCD:



A demosaicing filter is then used to reconstruct the colour at each pixel by examining its neighbours.

2.2.3 Scanner

Scanners are used to capture an image from paper or a flat object. The object must be flat as scanners cannot focus to produce a sharp image of an object that is not against the glass. Some scanners can capture the image from a slide film or negative by shining light from behind the transparency.



Most scanners use a linear CCD (i.e. narrow strip) to capture the image in strips as it passes over (or under!) the image. Three images are needed (a red, green and blue one) which are then combined to produce a full colour image.

These three images (red, green and blue) can be obtained by the scanner in various ways (depending on how it has been designed), for example: scanning the image three times - using a different colour filter each time; or using a beam splitter to split the light from the image through three coloured filters.

The resolution of a scanner is physically determined by the number of light sensors along the scan head (the scanning bar that moves down the image while capturing it) and by the number of small steps that the scan head can be moved down the image by the motor.

2.2.4 ADC

Light, as with all naturally occurring signals, is analogue. It must be turned into a digital signal if it is to be stored in, or manipulated by, a computer.

The **Analogue to Digital Converter** is an essential part of any image capturing device. The signals given from the CCD are analogue. It is the job of the ADC to convert these signals into digital ones that can be stored and manipulated by computer.

Comparing input devices

Using a computer supplies catalogue or a web site (e.g. www.dabs.com), look up details for two digital cameras and two flat-bed scanners.

Compare a low quality (cheap?) one of each with a high quality (expensive?) one of each.

Avoid multi-function devices - choose only devices that are purely cameras or scanners.

Create a table like this, or similar, depending on the details you can get.

Digital Cameras

	Low Quality	**High Quality**
Optical Resolution		
Maximum colour depth		
Price		
What other features justify the higher price or quality?		

Scanners

	Low Quality	**High Quality**
Optical Resolution		
Maximum colour depth		
Price		
What other features justify the higher price or quality?		

Note: the term 'optical resolution' is used throughout because this relates to the actual physical resolution of the devices. Some cameras, for example, re-sample (see later) the captured image, producing an image of greater size than the actual CCD is capable of producing on its own; these cameras often quote this re-sampled resolution as the camera's output, which could be regarded as being misleading.

2.3 Storage of graphic data

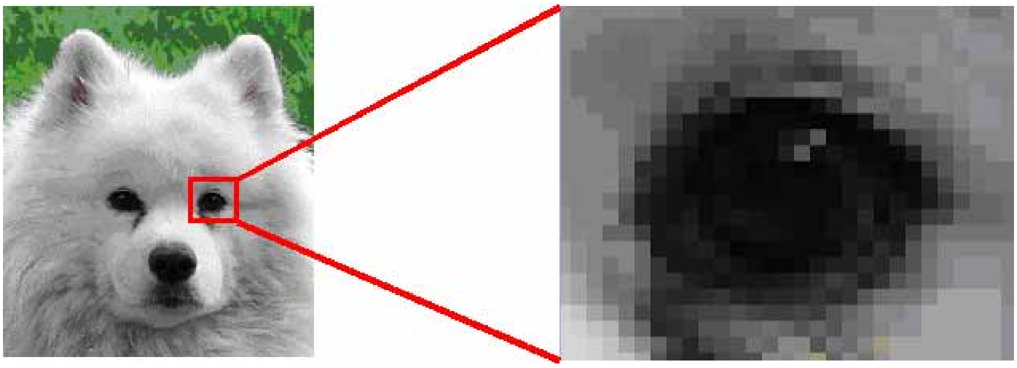
All of these file formats:

- Bitmap
- Graphic Interchange Format
- Joint Pictures Expert Group
- Portable Network Graphics

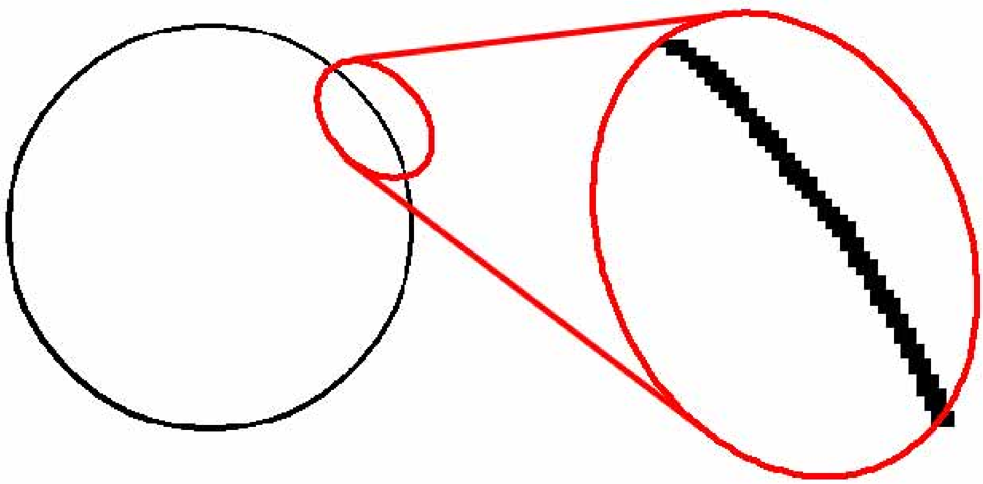
can be considered as bitmap file types, i.e. they store information on the colour and placement of each pixel in the image.

This means that all these image types will suffer from pixellation if enlarged too much, i.e. the individual pixels will become visible to the eye. Bitmap images are not suited to scaling.

For example:



or:



2.3.1 Bitmap

A bitmap file is a sequence of numbers, where each number identifies the colour of an individual pixel in the image. The numbers are stored on the disc as binary numbers (bits) and the order of them maps out the position of the colours on screen (map).

In order to identify the colour of a pixel a 24 bit RGB (Red, Green, Blue) code is needed, the colours used in the image are stored in the colour table (sometimes called the Colour LookUp Table or CLUT or simply the 'palette'). The use of a colour lookup table is also called indexed colour.

A 2 colour image would have 2 colours in the colour table, a 4 colour image would have 4 colours in the colour table and so on.

Consider this bitmap image (in colour on the course website):

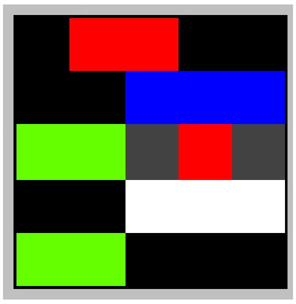


Figure 2.1: A bitmap image

Colours used in the image: 6

Colour depth of the image: 3 bits per pixel

Number of colours allowed: 8 (2 are unused here)

Resolution: 5 x 5 pixels (25 pixels in total)

Colour table for Figure 2.1

Colour Index (3 bits per pixel)	RGB colour code in decimal	24-bit RGB code in binary	Colour
00 (000)	000 255 000	00000000 11111111 00000000	green
01 (001)	000 000 000	00000000 00000000 00000000	black
02 (010)	255 255 255	11111111 11111111 11111111	white

Colour table for Figure 2.1 (continued)

Colour Index (3 bits per pixel)	RGB colour code in decimal	24-bit RGB code in binary	Colour
03 (011)	150 150 150	10010110 10010110 10010110	grey
04 (100)	255 000 000	11111111 00000000 00000000	red
05 (101)	000 000 255	00000000 00000000 11111111	blue
06 (110)	not used in this image	not used in this image	
07 (111)	not used in this image	not used in this image	

Image data for Figure 2.1

(starting on the bottom row, going from left to right)

In binary, as in the computer	In decimal, for comparison
000 000 001 001 001	00 00 01 01 01
001 001 010 010 010	01 01 02 02 02
000 000 011 100 011	00 00 03 04 03
001 001 101 101 101	01 01 05 05 05
001 100 100 001 001	01 04 04 01 01

The bitmap file contains the colour table **and** the image data.

2.3.1.1 Colour depth

To calculate the number of colours a particular colour depth can represent we use this calculation:

Number of colours = $2^{\text{bit depth}}$ (read as "two to the power...")

Examples

1.

Problem:

How many colours are represented by a colour bit depth of 3?

Solution:

Colour bit depth = 3

Number of colours = 2^3 (read as "two to the power three")

Do this on a calculator, or using the following method:

Number of colours = 2 x 2 x 2
= 4 x 2
= 8 possible colours

All 8 binary values for 3-bit numbers:

Binary number	Decimal equivalent
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

2.

Problem:

How many colours are represented by a colour bit depth of 8?

Solution:

Number of colours = 2^8
= 2 x 2 x 2 x 2 x 2 x 2 x 2 x 2
= 4 x 2 x 2 x 2 x 2 x 2 x 2
= 8 x 2 x 2 x 2 x 2 x 2
=16 x 2 x 2 x 2 x 2
= 32 x 2 x 2 x 2
= 64 x 2 x 2
=128 x 2
= 256 possible colours

2.3.1.2 RLE Compression

The file size of bitmap files can be reduced using a simple form of lossless compression called **Run Length Encoding** or RLE.

Compressed data consists of 2 numbers: Number of pixels : Colour index to repeat

If the first number is zero (0) this has a special meaning, depending on what follows: If the second number is 1 this means the end of the bitmap data; If the second number is 0 this means the end of the line.

For example:

Compressed data	Expanded data
04 07	07 07 07 07
03 01	01 01 01
00 00	End of line
00 01	End of bitmap data
01 00	00
08 08	08 08 08 08 08 08 08 08

This compression method works best for images with solid blocks of colour (i.e. long runs of a particular colour).

Although the RLE compression does not actually reduce the size of such a small image, here is the bitmap from the previous example shown using RLE:

Compressed data	Original uncompressed data
02 00 03 01 00 00	00 00 01 01 01
02 01 03 02 00 00	01 01 02 02 02
02 00 01 03 01 04 01 03 00 00	00 00 03 04 03
02 01 03 05 00 00	01 01 05 05 05
01 01 02 04 02 01 00 01	01 04 04 01 01

For comparison (and to illustrate that RLE does actually reduce the size of bitmap files) here is the data for a solid black square of 100 pixels (10 x 10) in an 8-bit image:

Compressed data (decimal numbers)	Original uncompressed data
10 00 00 00	00 00 00 00 00 00 00 00 00 00
10 00 00 00	00 00 00 00 00 00 00 00 00 00
10 00 00 00	00 00 00 00 00 00 00 00 00 00
10 00 00 00	00 00 00 00 00 00 00 00 00 00
10 00 00 00	00 00 00 00 00 00 00 00 00 00
10 00 00 00	00 00 00 00 00 00 00 00 00 00
10 00 00 00	00 00 00 00 00 00 00 00 00 00
10 00 00 00	00 00 00 00 00 00 00 00 00 00
10 00 00 00	00 00 00 00 00 00 00 00 00 00
10 00 00 00	00 00 00 00 00 00 00 00 00 00

The disk space and memory requirements have been reduced by using RLE compression, and the image quality remains exactly the same as this compression is lossless.

The length of run of each colour is given by a byte, so the maximum run of each colour using RLE is 255 repetitions.

Remember as noted, the colour index for this sample image is also a byte.

The storage requirements for the data in this image is then:

Compressed data: 40 bytes

Original data: 100 bytes

Bitmaps and compression

Q1: Calculate the file sizes of these uncompressed bitmap images (If you need to, read Section 2.4 first):

1. 2 colour image size 400 x 300 pixels
2. 16 colour image, size 600 x 400 pixels
3. 256 colour image, size 600 x 600 pixels

Estimate the file sizes, including the colour tables? (Do all three of these files need a colour table?)

Draw three pictures in a painting program, saving each with the attributes used in a, b & c as type 'Bitmap' images.

If using 'Paint', 'Image - attributes' can be used to set the image size in pixels.

(If given the option in your painting program, choose 'uncompressed' and compare the actual file sizes with the estimated ones)

Find out the file size of each, e.g. 'image - attributes' if using 'Paint' or Right-click the saved file and choose 'properties'.

Q2: Does the painting program you used use compression when saving 2 colour bitmap images?

Q3: How can you tell?

Q4: Does the painting program you used use compression when saving 16 colour bitmap images?

Q5: How can you tell?

Q6: Does the painting program you used use compression when saving 256 colour bitmap images?

Q7: How can you tell?

Q8: The 2 colour image is often just termed a 'bitmap' image. Why do you think this is?

2.3.1.3 24-bit bitmaps

Bitmap images with a colour depth of 24 bits per pixel do not have (or need) a colour table. Instead each set of 3 bytes (ie. 24 bits) in the bitmap data represents the colour of a single pixel, this is the RGB colour code for each pixel.

For example, Figure 2.1 could be stored as a 24-bit bitmap image as follows:

Image data (shown in decimal):

000 255 000	000 255 000	000 000 000	000 000 000	000 000 000
000 000 000	000 000 000	255 255 255	255 255 255	255 255 255
000 255 000	000 255 000	150 150 150	255 000 000	150 150 150
000 000 000	000 000 000	000 000 255	000 000 255	000 000 255
000 000 000	255 000 000	255 000 000	000 000 000	000 000 000

Because:

- 000 255 000 is a 24-bit code for Green
- 000 000 000 is a 24-bit code for Black
- 255 255 255 is a 24-bit code for White
- 150 150 150 is a 24-bit code for Grey
- 255 000 000 is a 24-bit code for Red
- 000 000 255 is a 24-bit code for Blue

No colour table is needed because the bitmap data contains the actual 24 bit RGB colour codes anyway!

24-bit images

Q9: Calculate the sizes of the following files

1. A 400 x 300, 24 bit image
2. A 600 x 400, 24 bit image
3. A 600 x 600, 24 bit image

Open the 3 files you created earlier and use 'Save As' to save them as 24-bit images. (Some painting programs need you to change the colour depth some other way *before* saving).

Compare the actual file sizes with your calculated ones.

2.3.2 GIF

The **Gif** (Graphics Interchange Format) was developed as an efficient graphical file format to transfer images around the Internet.

It compresses the file using a lossless method, i.e. the image quality remains exactly the same as the original file. The compression method is called **LZW** compression.

The compression method it uses (mathematical rules, called an algorithm) makes it best suited to storing images that have flat areas of colour, although it is more effective than RLE. It has one main drawback - it stores images using 8-bits per pixel, limiting it to 256 colours.

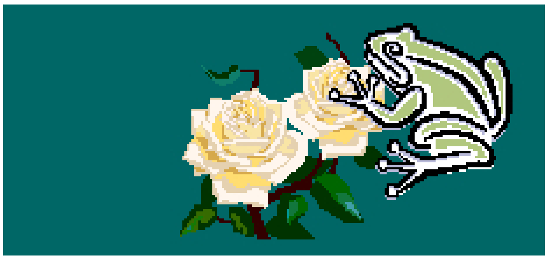
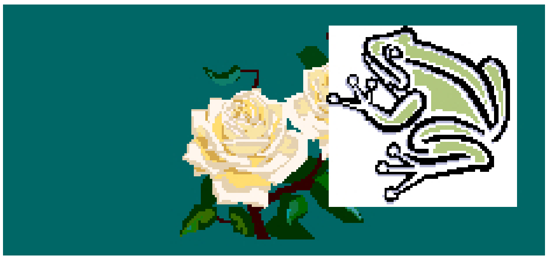
2.3.2.1 Extra features of GIF images

Animation

Gifs can also store several images in a single file, playing them back as a movie or animation. Animated Gifs are often seen on web pages.

Transparency

Gifs allow transparency, i.e. you can see through certain areas of the images, e.g.

Image of a frog with a transparent area	Image of a frog without transparency
	

Gif is a good format for:

- Images with few colours
- Images with flat areas of colour
- Animated images
- Images needing transparent areas

Gif is a poor format for:

- Photographs
- High-colour pictures
- High quality videos

2.3.2.2 LZW Compression

A lossless method for compressing files was developed by Abraham Lempel and Jacob Ziv, called LZ78. This was developed by Terry Welch creating a compression method called LZW (Lempel-Ziv-Welch). Although it is an efficient method for compressing files it is the main drawback to Gif files as the patent to the LZW algorithm is held by the company 'Unisys' who charge for its commercial use (although this patent has now expired in most parts of the world).

This method of compressing files is called 'string table compression'. This type of compression uses a dictionary and sends the code for that item in the dictionary rather than sending the actual data - this has similarities to the use of a colour table.

The steps (algorithm) for performing this compression are as follows:

```
1. Start with the string table for all the smallest parts
 of the file (e.g. A table with codes for all letters
in the alphabet or colours in the image)

2. Let [CurrentString] = empty

3. Let [Next] = next character in the data stream

4. Is [CurrentString].[Next] in string table?
    a. YES:
      i. Let [CurrentString] = [CurrentString].[Next]
      ii. Go to 3
    b. NO:
      i. Add [CurrentString].[Next] to string table
      ii. Store Code for [CurrentString]
      iii. Let [CurrentString] = [Next]
      iv. Go to 3
```

Example : Table compression (assuming a maximum code length of 3 characters)

Problem:

Say we wish to send this data:

ABABCABABC

The starting table is:

0=A, 1=B, 2=C (a code for all letters in this message)

Add 4=AB to table (A is in the table but AB is not)

0=A, 1=B, 2=C, 3=AB

Store 0 (A) Add 4=BA to table (B is in the table but BA is not)

0=A, 1=B, 2=C, 3=AB, 4=BA

Store 1 (B)

Add 5=ABC to table (AB is in the table but ABC is not)

0=A, 1=B, 2=C, 3=AB, 4=BA, 5=ABC

Store 3 (AB)

Add 6=CA to table (C is in the table but CA is not)

0=A, 1=B, 2=C, 3=AB, 4=BA, 5=ABC, 6=CA

Store 2 (C)

Add 7=ABA to table (AB is in the table but ABA is not)

0=A, 1=B, 2=C, 3=AB, 4=BA, 5=ABC, 6=CA, 7=ABA

Store 3 (AB)

Store 5 (ABC)

Solution:

So the table of codes is constructed as the file is compressed. In the above example, the letters A,B etc could represent the colour values for different pixels in the image.

You should understand that string table compression schemes such as this use a 'dictionary' or table of the most repeated sets of characters within the data stream in order to compress it.

As with most compression schemes it is difficult to see the compression effect unless examining large amounts of data. The above example only 'compresses' 10 characters so there is no actual compression of the data. However, it *is* effective for large amounts of data.

2.3.2.3 Interlacing

Interlacing is also known as 'progressive coding'.

This is used to allow an image to be viewed before it is entirely downloaded. Without interlacing an image file would be displayed line at a time (or remain blank until totally downloaded). Using interlacing an image is seen to be built up and have greater detail or resolution added as the file is downloaded.

There are different methods for interlacing image data. Gif files interlace the data by showing every 8th row, then every 4th row, every 2nd row and finally all the rows of image data. Non-interlaced Gif files start at the top row as the image appears row-by-row from the top of the image.

Saving a Gif file as an interlaced file increases its size slightly.

Figure 2.2 shows a screenshot of a non-interlaced (left) and an interlaced gif (right) downloading:

Figure 2.2: image from http://wp.netscape.com/eng/mozilla/2.0/relnotes/demo/pjpegdemo.html

Interlacing is simply used as a more appealing way to deliver images over networks (especially at low bandwidths / data transfer speeds).

2.3.3 JPEG

JPEG (pronounced "jay-peg") is a standardised image compression technique. JPEG stands for Joint Photographic Experts Group, the original name of the committee that wrote the standard.

JPEG is designed for compressing either full-color or gray-scale images of natural, real-world scenes. It works well on photographs, naturalistic artwork, and similar material; not so well on lettering, simple cartoons, or line drawings. JPEG handles only still images, but there is a related standard called **MPEG** for motion pictures.

Compression

JPEG is "lossy," meaning that the decompressed image isn't quite the same as the one you started with. (There are lossless image compression algorithms, but lossy JPEG achieves much greater compression than is possible with lossless methods.)

JPEG is designed to exploit known limitations of the human eye, notably the fact that small color changes are perceived less accurately than small changes in brightness.

Thus, JPEG is intended for compressing images that will be looked at by humans. If you plan to machine-analyze your images, the small errors introduced by JPEG may be a problem for you, even if they are invisible to the eye. They use a mathematical technique involving (among others) something called a Discrete Cosine Transformation (DCT).

A useful property of JPEG is that the degree of lossiness can be varied by adjusting compression parameters. This means that the image maker can trade off file size against output image quality. You can make *extremely* small files if you don't mind poor quality; this is useful for applications such as creating thumbnails of images. Conversely, if you aren't happy with the output quality at the default compression setting, you can increase the quality until you are satisfied, and accept lesser compression.

JPEG compression is only really effective on larger files (e.g. 200 x 200 pixels or larger), this is why small images used on web pages tend to be GIF files, even if they are photographic (although still restricted to 256 colours).

Example : A compressed JPEG

Problem:

Lowest quality	Medium quality	Maximum quality
		
File size: 41 Kilobytes	File size: 85 Kilobytes	File size: 288 Kilobytes

Solution:

The degradation of the image due to the JPEG compression is most noticeable at maximum compression.

If you look carefully you will see that the most noticeable marks in the image (called 'artefacts') occur where boundaries between contrasting colours happens (e.g. between the edge of the arm and the sky) - this makes JPEG unsuitable for images with text as the blurring at the boundaries makes it unreadable.

Standard, basic JPEG files do not support **interlacing**.

Standard, basic JPEG files do not support **transparency**.

JPEG files always store pixel colour information using 24 bits per pixel as they do not use colour lookup tables.

An extra note on Huffman Compression

An older but still common compression technique is worth mentioning here: 'Huffman Compression'. Although this is a non-lossy technique JPEG and MPEG files use it in combination with other forms of compression.

The idea behind Huffman encoding is:

Give each piece of data a code which could be a different length, pieces of data that are common and occur most times in the file are given the shortest codes and other pieces of data, occurring only on a few occasions are given longer codes.

Example : Huffman Encoding

Problem:

AAAAAAAABBBBBCCCCCC

Uncompressed	Huffman Compression
3 symbols (A,B,C) so 2 bits are needed for each	3 symbols (A,B,C) so 2 bits are needed for each
A=0 C=01 B=10	A=0 (A is most common so it is given a shorter code) C=10 B=11
00 00 00 00 00 00 00 00 11 11 11 11 11 10 10 10 10 10 10	0 0 0 0 0 0 0 0 11 11 11 11 11 10 10 10 10 10 10
Total: 38 bits	Total: 30 bits

Huffman compression has been shown to be less efficient than LZW compression.

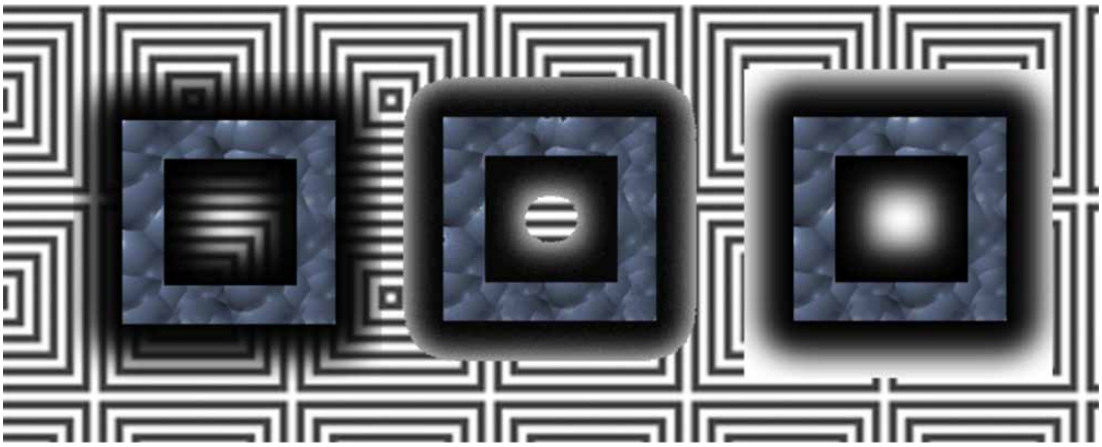
Solution:

2.3.4 PNG

PNG (officially pronounced 'ping'), the Portable Network Graphic file format attempts to incorporate the advantages from Gif files, without all the restrictions.

A range of colour depths is supported, including 24 bits per pixel.

Transparency support is better than in Gifs. Gifs only allow a pixel to be coloured or completely see-through, whereas PNG files allow pixels to be partially transparent.



PNG	GIF	JPEG
Pixels can be partially transparent, allowing the background to show through to different degrees	Pixels are either completely transparent, allowing background to show through or are completely opaque, hiding the background	All pixels in the image rectangle are opaque, completely hiding the background

PNG files can be saved as **interleaved** files.

PNG files do not support moving images but this is by design. A modified PNG file format called MNG ('ming' Multiple-image Network Graphics) has been designed for this purpose. MNG files basically consist of many PNG images that can be played together to form an animation or video. MNG files also support different forms of compression - as there is a greater need for compression in video files.

24-bit or greater PNG files do not use a colour table, but PNG files with a colour depth of less than this *do* use a colour lookup table.

PNG files use a similar method of file compression to LZW (as used in Gif files), except this is royalty free.

In order to view PNG files in Internet Explorer (6) you will need to install this plug-in: http://entropymine.com/jason/mng4ie/ or http://www.libmng.com/downloadplugins.html

2.3.5 A summary of bitmap graphic types

	Compression			Transparency	Moving images	Interleaving	Colour Depth
	Type	Best for...	Lossy or lossless				
Bitmap	RLE	Very basic, not effective when compared to other forms of compression	lossless	No	No	No	Any
Gif	LZW	Images with areas of solid colour	lossless	Yes *fully transparent or fully opaque*	Yes	Yes	8-bit only
Jpeg	Involving DCT	Photographic images *Only effective only larger images*	lossy *Amount of compression can be controlled by the user*	No (*)	No *Only in related MPEG format*	No (*)	24 bit
Png	Variation of LZW	Images with areas of solid colour	lossless	Yes *levels of transparency supported*	No *Only in related MNG format*	Yes	Any

Note: image formats here do not use a colour lookup table (CLUT) when using colour depths of 24 bits per pixel and they all use a CLUT when using colour depths of less than this (except for monochrome - 1bit (2-colour) images, which do not need a colour table).

Bitmap formats

This is a web-based activity. 'Right-click' (PC) or 'CTRL-click' (Mac) these files and use 'Save Target As' (Internet Explorer) or 'Save Link Target As' (Netscape) to save them in your own folder.

SCOTTISH QUALIFICATIONS AUTHORITY

Open each of the images in bitmap editing software.

Use 'Save As' (or the method used by your own software) to make copies of these files in the formats shown in the table. Do not forget to re-load the original file before each 'Save As'.

If the program you are using does not support one of the formats above, try using another. If you get really stuck and you do not have the support for one of the file types, just miss that one out.

	Photograph		**Logo**	
Format/Setting to use	**Image quality**	**Size**	**Image quality**	**Size**
Max quality JPEG				
75% quality JPEG				
50% quality JPEG				
25% quality JPEG				
Min quality JPEG				
GIF				
8 bit PNG				
24 bit PNG				
24 bit bitmap				
256 colour bitmap				
256 colour, RLE bitmap				
16 colour bitmap				
2 colour bitmap				

Open each of the files you have saved and record your comments on the quality of each image. Also note the file size.

What is the best format to use for each image in a web page? In other words, which format offers the best size vs. quality ratio for each type of image?

Arrange all 22 images side-by-side for comparison using Publishing or Presentation software.

Q10: Make the white area in the logo transparent, saving it as a transparent GIF and PNG 8 bit. Does transparency affect the file size? Add these images to your comparison page.

2.4 Calculations

The file size of an uncompressed bitmap in bytes can be calculated using this method, it is an approximation to the actual file size as this calculation does not include the colour table (if used) nor account for any other information that is included in the file.

These calculations become more accurate for larger files, with higher resolutions.

If the colour depth per pixel is in bits:

$$\text{File size (bytes)} = \frac{\text{Number of pixels x colour depth per pixel (bits)}}{8}$$

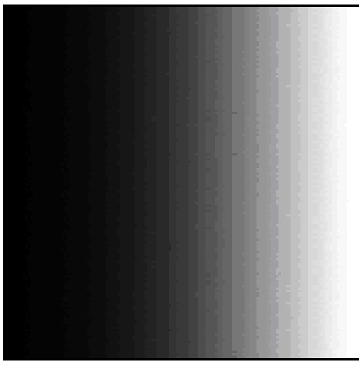
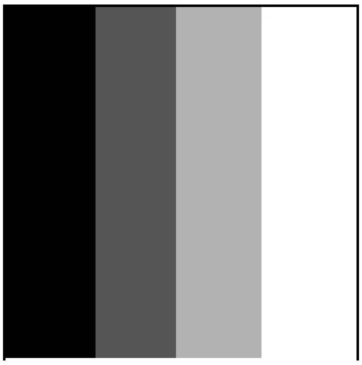
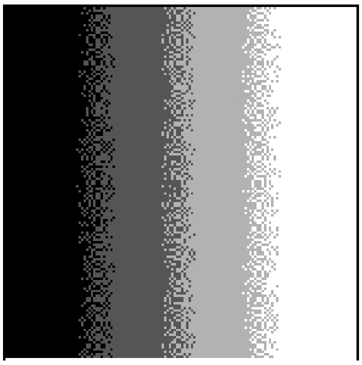
If the colour depth per pixel is in bytes:

File size (bytes) = Number of pixels x colour depth per pixel (bytes)

2.5 Technical terms

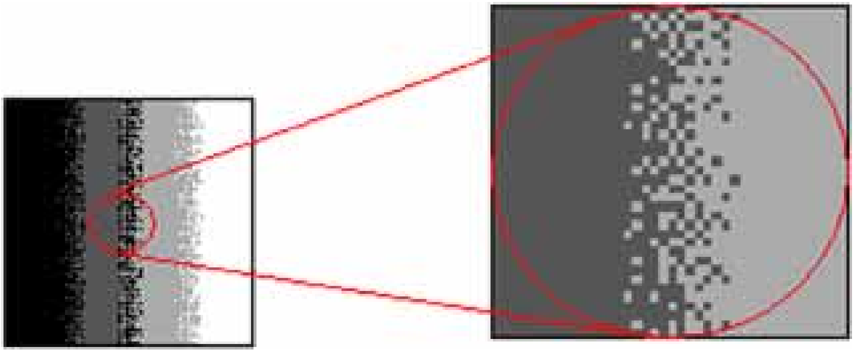
2.5.1 Dithering

Dithering simulates extra colours in an image by placing the allowed colours in the image close together. This is called optical colour mixing.

		
The image as we wish it to appear	4 colour image displays banding or **posterisation**	4 colour image using dithering to reduce posterisation (or banding) effect.

(Colour depth is 2 bits per pixel only allowing 4 shades of grey.)

In the above diagram you can see that extra shades of grey are simulated by mixing the allowed colours together:



Dithering never looks as good as a full colour image but it does reduce banding.

2.5.2 Anti-alias

Anti-aliasing simulates extra resolution in an image by adding extra colours along sharp edges to reduce the stepping effect or 'staircase effect' often called jaggies. The correct term for this staircase effect is **aliasing**.

The image as we wish it to appear.

Without anti-aliasing the low resolution of this image causes this diagonal line to display aliasing.

By anti-aliasing this image additional colours have been introduced to reduce the staircase effect and simulate a higher resolution than we actually have.

This image has the same resolution (number of pixels) as the previous one, yet the sloping boundary appears smoother.

Antialiasing

Create a new image in a bitmap editing program (e.g. Paint).

Add some text, using a large font size.

Use a pencil tool or sharp brush, draw a border around the text

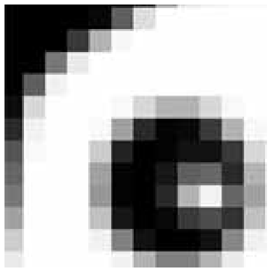
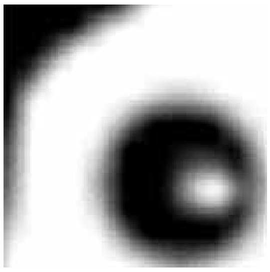
Zoom close in, to examine the edges of the text. Has anti-aliasing been used?

2.5.3 Resampling

If an image is captured using only 256 colours (for example), then the missing colours cannot be added in afterwards, in other words there is no point *increasing* the colour depth. The computer or software cannot *know* what colours are missing.

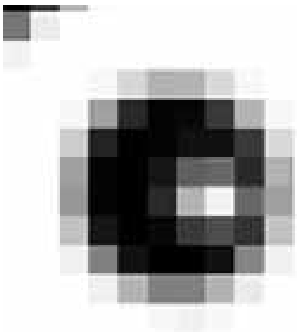
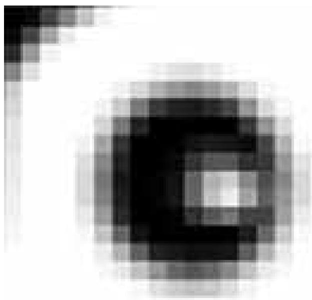
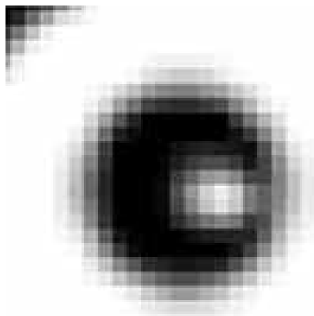
Likewise, if the image is captured using a low resolution , then the computer or software cannot *know* what pixels are missing and fill in the extra detail.

However, it *is* possible to increase the resolution of an image using re-sampling to reduce the effect of pixellation. Here the software fills in the missing pixels using best guesses. You cannot add in detail that was never in the image to start with, but you can reduce the effect of pixellation when the image is enlarged.

			
The original image(100 x 100 pixels)	Enlarged, the original image clearly shows pixellation	Enlarged, the re-sampled image, pixellation has been reduced	The re-sampled image (now 500 x 500 pixels)

How re-sampling works:

Although there are many methods, bicubic is one of the most common. It works by calculating the colours of the new pixels by giving them the average colour of their 4 neighbours.

		
The original image The circle is 8 pixels across	Re-sampled The circle is now 13 pixels across	Re-sampled further The circle is now 17 pixels across

Note: The most basic form of re-sampling is called 'nearest neighbour' and does not actually smooth the image, but only enlarges it, introducing pixellation. It is the equivalent of zooming in closer to a bitmap image and seeing the pixels. Although strictly speaking this is still re-sampling, we will not be regarding it as such in this course. You can use the 'Stretch' feature in Windows' Paint to see this in action (as in the activity 'Resampling').

Re-sampling of a low-resolution photograph:

		
Original image 50 x 67 pixels	Re-sampled to 100 x 134 pixels	Re-sampled to 200 x 268 pixels

As you can see, no extra detail is added, it is just the effect of pixellation that is reduced, this is why the re-sampled image appears blurrier than the original image.

If you need images at a higher resolution it is better to have captured it at that higher resolution in the first place. All re-sampling can do is increase the resolution of an image in order to reduce the effect of pixellation.

Here is how the above image would have looked if it had been captured at 200 x 268 pixels in the first place:

Resampling

Create a 100 x 100 image in Paint (or similar, basic software) and write your name.

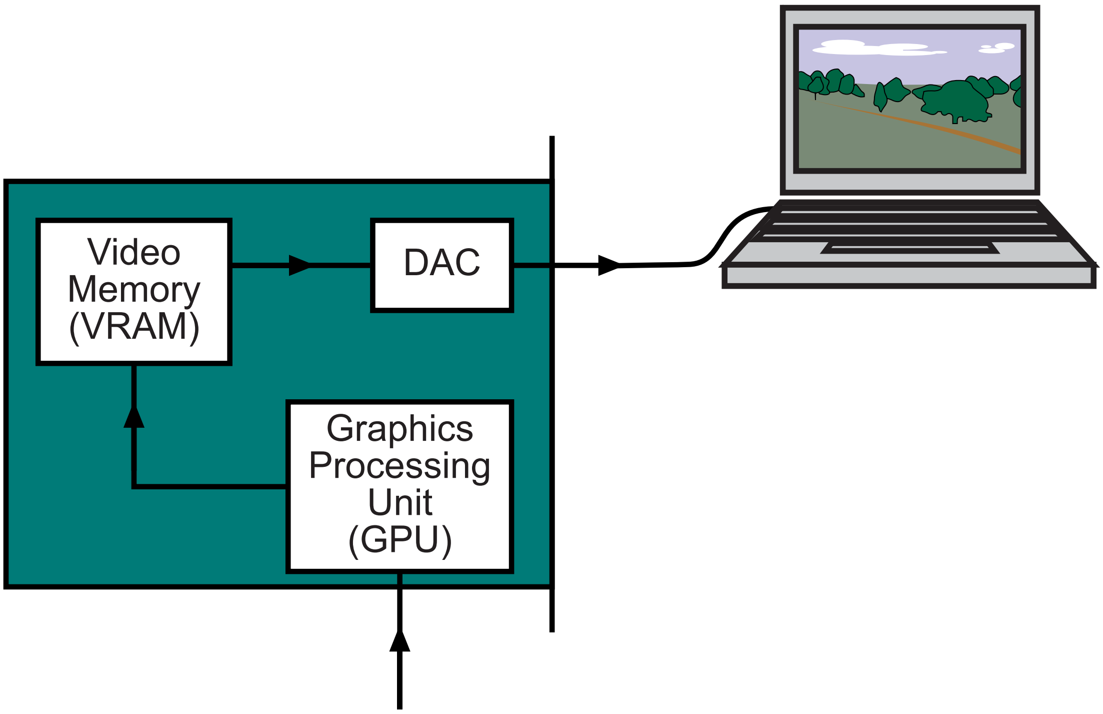
Use stretch/ skew (or other method) to increase the size by 1000% in both directions. Has the image been re-sampled?

Repeat steps 1 & 2 using different software (for example 'Microsoft Photo Editor')

Using knowledge from this section, prepare and add a bitmap image to one of the multimedia applications you created earlier.

2.6 Output hardware

The graphics card is the device in the computer responsible for generating images.



All images displayed are held in the memory of the graphics card (called **VRAM** - or Video RAM). The amount of memory available affects the maximum colour depth and resolution supported by that graphics card.

Example: to support a display of 1600 x 1200 pixels at 24 bit colour, the graphics card needs to have: 1600 x 1200 x 3(bytes) = 5.76MB of VRAM

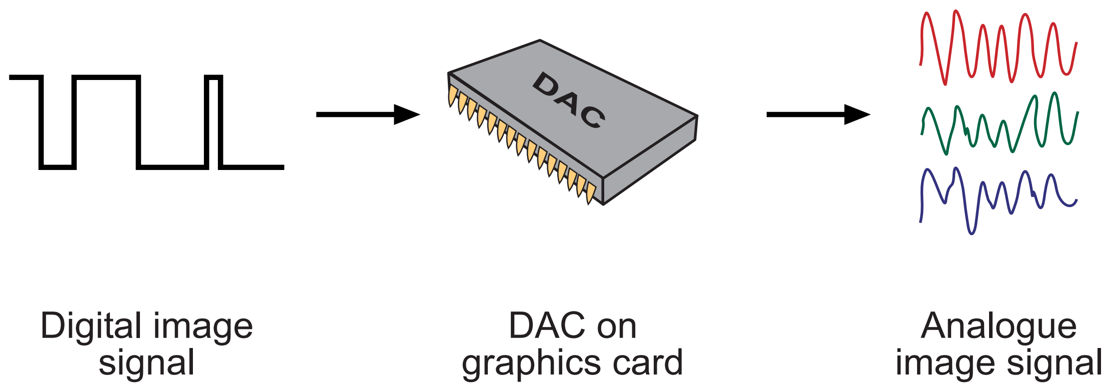
The size of VRAM and speed that it works has an effect on the speed that applications can generate images. As action games place big demands on the graphics processing capabilities of computers, it is not uncommon to find 'gaming' graphics cards with large amounts of VRAM (e.g. 128MB or greater). This also benefits other, graphics intensive, applications.

2.6.1 Digital to Analogue Converter (DAC)

Most computer monitors currently still need an analogue signal. The graphics card generates the image then outputs it as an analogue signal that can be viewed on normal computer monitors.

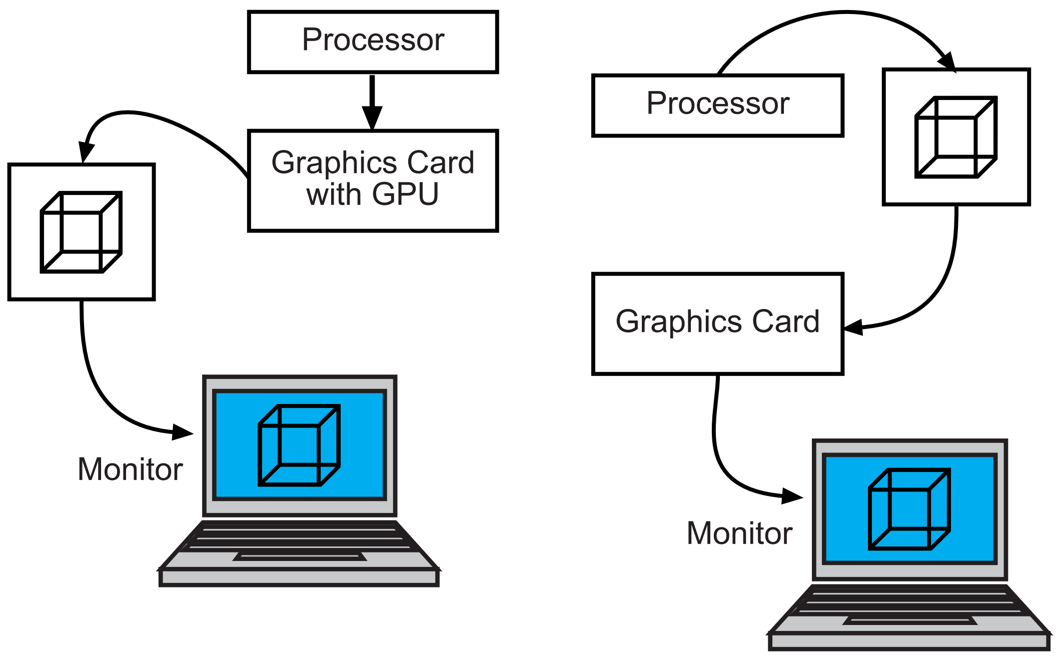
The **DAC** in a graphics card converts the digital computer signal into an analogue one that the monitor needs.

In the future it is likely that analogue displays will be replaced with displays that accept a digital signal. There are many LCD displays that currently allow digital signals (called DVI - Digital Visual Interface), this type of connection will become more common in the future.



2.6.2 GPU

The Graphics Processing Unit (GPU) is a digital signal processor (DSP). The GPU on the graphics card is often more advanced than the processor that is at the core of the computer system. The need for these has been driven on mainly by the requirements of modern computer games, with high resolution, detailed, complex 3D graphics. This is often 'overkill' on a computer which will only be used to process 2D (flat) images, although the processing power of these graphics cards can be helpful when working with large images, animations or video.



As the GPU has been created only for the purpose of generating images, it generates images faster and with better quality than the images that would be generated by the processor.

Games and other multimedia applications for Microsoft Windows usually employ a technology called DirectX, which is basically a set of commands to instruct computers to generate images. The GPU is programmed to be able to generate images from the DirectX commands, whereas if the processor were to generate images from DirectX commands, it must decode them using time-consuming programming instead of a special computer chip.

There is a special subset of DirectX used purely for 2D images, called DirectDraw.

The GPU not only generates images when given the geometry but manipulates the images and applies special effects (for example anti-aliasing) - all of which is very time-consuming (slowing down the computer) if it needs to be performed by software, if there is not a GPU present that can perform those tasks using hardware.

2.7 Questions

Q11: High colour digital images of medical scans are stored in a hospital computer. These images must be stored exactly as they are received from the medical scanner. Which of these file formats would be suitable?

a) Bitmap
b) JPEG (maximum compression)
c) GIF
d) JPEG (minimum compression)

Q12: Explain why each of the other file types would not be suitable.

Q13: How many bits per pixel are required to store an image needing 200 colours?

a) 4 bits
b) 8 bits
c) 24 bits
d) 200 bits

Q14: Write out a table showing the maximum number of colours that can be represented using these colour depths: 1bpp, 2bpp, 4bpp, 8bpp, 16bpp, 24bpp.

Q15: A photograph with transparent areas is needed for a web page. Which would be the best file format to use?

Bitmap, JPEG, GIF, PNG

Q16: Describe when the other 3 file formats would be most suitable.

Q17: A logo is placed on a web page but suffers from pixellation. The file size must not be increased any more. What can be done to reduce this effect?

a) Re-sample the image
b) Anti-alias the image
c) Use dithering
d) Save it in JPEG format

Q18: Explain how your answer achieves this.

Q19: Here is a line of a bitmap image stored using RLE compression:

02 03 01 04 00 00

Which answer show the uncompressed data?

a) 03 03 04 00
b) 02 02 02 01 01 01 01
c) 01 01 04 04 04
d) 03 03 04

Q20: What is the purpose of the '00 00' at the end of the compressed data line?

Topic 3

Digitised sound data

Contents

Prerequisite knowledge

Before studying this topic you should have:

- *Basics of using sound editing software (e.g. sound recorder or GoldWave)*
- *Familiarity with binary numbers*

All topics require:

- *familiarity in working with the computer filing system - i.e. the ability to create, view the properties of, save and open files and folders;*
- *the ability to locate and run the necessary software for each topic;*

- *the ability to switch between multiple applications on the computer system;*
- *knowledge of units of storage (bits, Bytes, KB, MB);*
- *knowledge of the basic components of computer systems (mostly just input, output and backing store devices).*

Learning Objectives

By the end of this topic you should be able to:

- *Describe the hardware required for capturing sound data;*
- *Describe the process of digitising sound;*
- *Compare factors affecting the quality and size of the captured sound file;*
- *Describe common file formats for storing sound data;*
- *Describe how MP3 files compress the sound data;*
- *Correctly use technical terms related to the editing of sound files;*
- *Perform calculations involving the size of sound files;*
- *Describe the hardware required for the output of sound data.*

3.1 Introduction

Digitised sound can be thought of as a recording of a real-life sound. That is, sound that has been captured and converted into a digital signal. When sound is in a digital format it can be stored and manipulated by computer.

Audio CDs, digital television, digital satellite, Digital Audio Broadcasting (DAB), minidiscs and DVDs all store or transmit sound in a digital format so all of these use digitised sound.

3.2 Input (capture) hardware

Sound, as all other naturally occurring signals, is analogue.

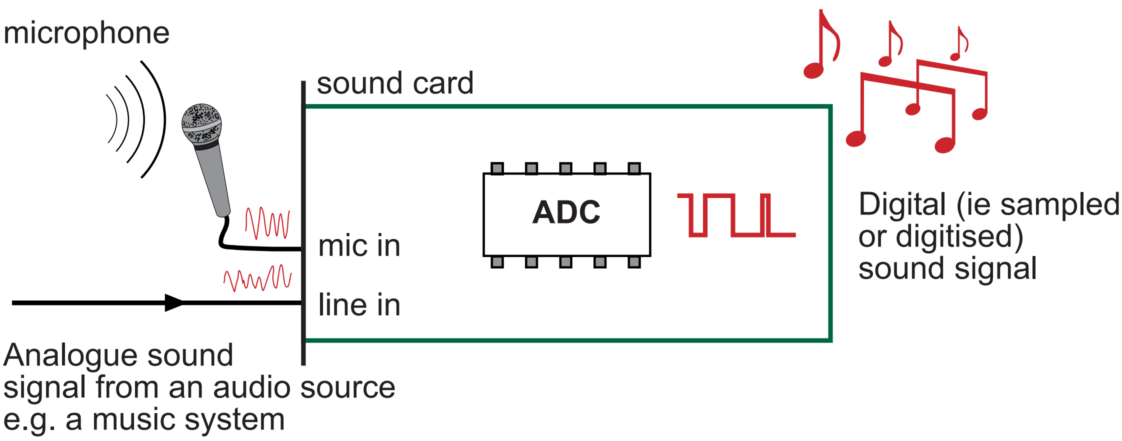
Sound is generally captured using a microphone. Microphones turn the analogue sound into an analogue electrical signal that can be transmitted along a wire.

3.2.1 ADC

An Analogue to Digital Converter (ADC) is needed to turn the analogue sound signal into a digital signal that can be stored and manipulated by computer.

This is known as Sampling or Digitising the sound.

The ADC needed to capture sound is generally part of the sound card.



Most modern soundcards also have a digital input (SP/DIF - Sony & Philips / Digital Interconnect Format). This can be used to transfer digital sound data from other digital devices, such as a CD or DVD player, **eliminating** interference from analogue noise and the degrading of the sound by having to digitise it - it is already digital and does not pass through the ADC.

3.2.1.1 How sound is captured, digitised or sampled

The analogue electrical sound signal produces a signal like this:

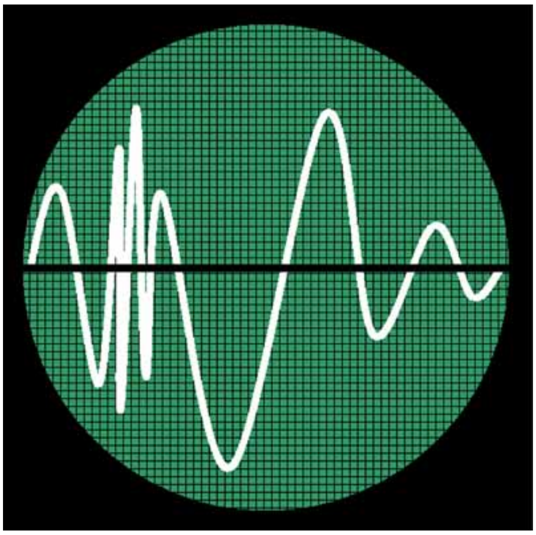


Figure 3.1: The electrical sound signal as it may be seen on an oscilloscope

The computer can only handle digital signals, so this analogue signal has to be turned into numbers. This is done by measuring it.

3.2.1.2 Sampling frequency

Taking measurements of the signal is called sampling it.

An example is given in Figure 3.2.

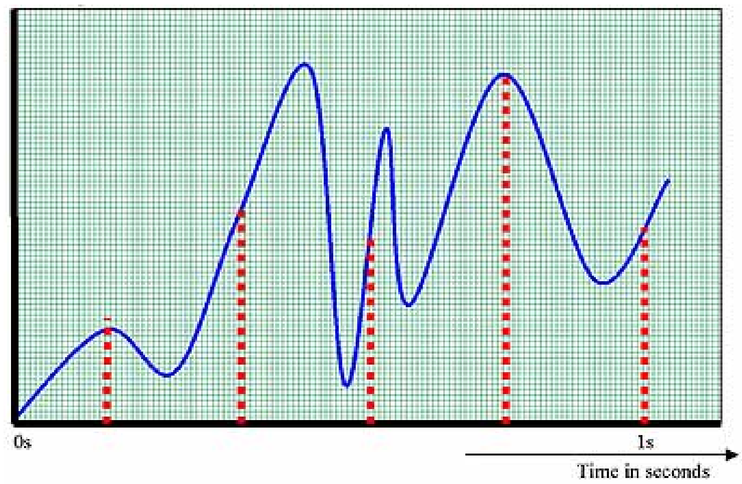


Figure 3.2: Sampling frequency

In Figure 3.2, 5 measurements, or samples, are taken in 1 second, so we say the signal is sampled using a sampling frequency of 5Hz (5 Hertz).

The number of times each second that the signal is sampled (measured) is called the **sampling frequency**.

The sampling frequency should be greater than twice the frequency of the highest pitch sound that we want to capture. Since humans can only hear frequencies of up to 20000 Hz (20KHz), then we need a sampling frequency of greater than 40000 Hz (40KHz).

Human speech is all below 4000 Hz so a sampling frequency of 8000 Hz is used in telephone systems, this is why speech sounds fine but music sounds poor quality when played down the telephone.

3.2.1.3 Sampling depth

The measurements are taken by reading the strength of the signal for each sample point. This is when we turn the signal into a number, called quantising the signal.

In this example each measurement is stored using 2 bits, giving us 4 possible levels to record the signal. The number of bits used to record each measurement is called the sampling depth. The sampling depth determines the dynamic range of the recording.

In Figure 3.3 the sampling depth is 2 bits per sample.

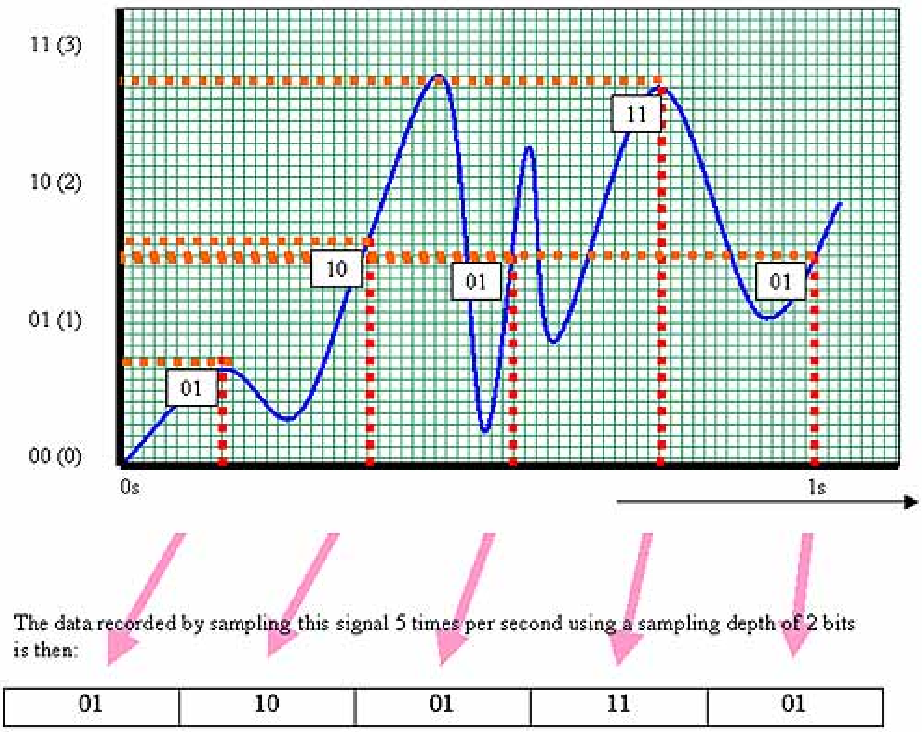


Figure 3.3: Sampling depth of 2 bits per sample

If this data is now used to try and recreate the sound wave it looks like Figure 3.4:

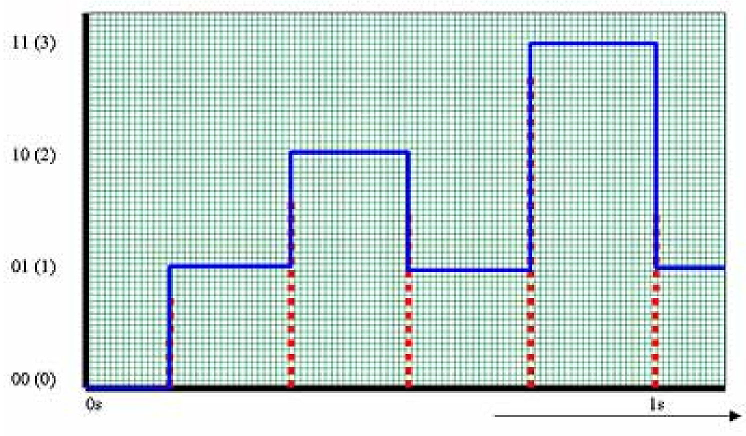


Figure 3.4: Recreated sound wave

The signal is now sent to the speakers as an analogue signal, in this case, even although the sharp edges of the digital signal may be rounded by the circuitry of the sound card, it is still a poor representation of the original signal (Figure 3.5).

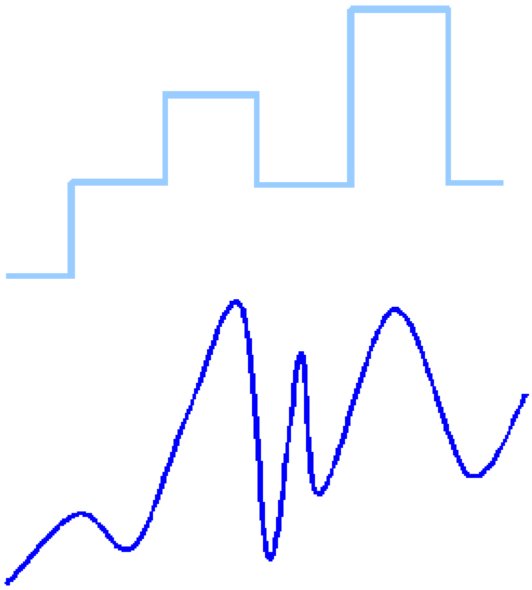


Figure 3.5: Representation (top) vs original (bottom) signal

The quality / accuracy of this recording would be improved by:

- Using a greater sampling frequency

  Audio CDs are sampled 44100 times every second; we sampled 5 times per second

- Using a greater sampling depth

  Audio CDs use a sampling depth of 16 bits, we used 2 bits (giving a dynamic range of 65536 levels of signal that can be recorded, we had 4 levels available)

The digital waveform of the letter ’S’ being spoken is shown in Figure 3.6. It was captured using the following settings:

- Sampling frequency: 22050Hz
- Sampling depth: 8 bits per sample

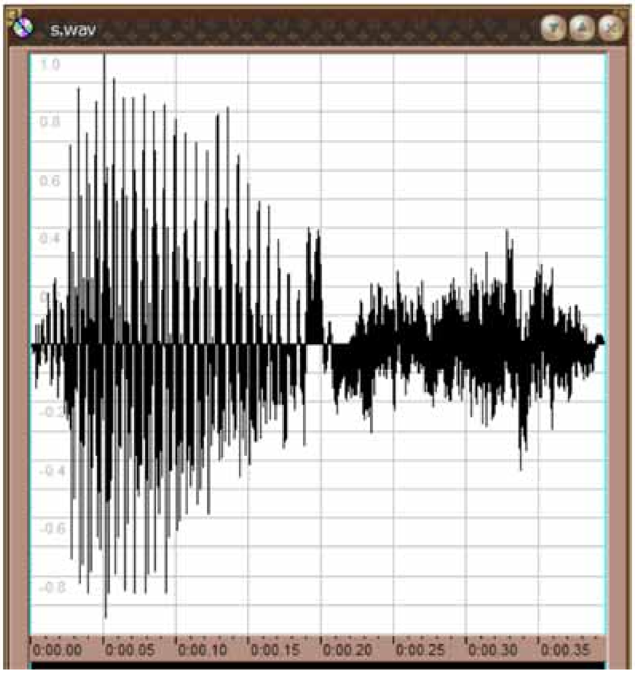


Figure 3.6: Digital waveform of the letter ’S’ being spoken

If we zoom in on a small part of the waveform (Figure 3.7), we can see the staircase effect caused by digitising the signal. This **always** happens and is the equivalent of seeing the pixels in a bitmap image if it is enlarged enough.

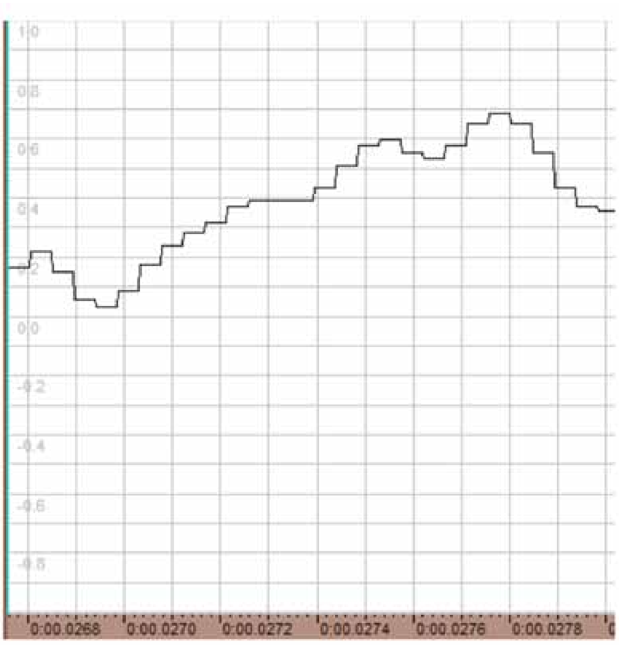


Figure 3.7: Zoom view of the digitised waveform

The extra disc space offered by DVDs is being used for a new format of audio disc with a quality that surpasses that of Audio CDs, called DVD-Audio.

	Audio CD	**DVD-Audio**
Sampling frequency	44100 Hz	192000 Hz (using 2 channels) 96000 Hz (using 6 channels)
Sampling depth	16 bits (dynamic range: 65536 - nearly 66,000 levels)	24 bits (dynamic range 1677216 - nearly 17 million levels)
Audio channels	2	6

Due mainly to the high sampling frequencies needed, DVD-Audio discs can only be played in special DVD-Audio players or in computers with a sound card that has a **DAC** fast enough to cope with this sampling frequency.

Sound activity

In this activity you will compare the effects of different sampling frequencies and sampling depths on a sound.

Record your own sound using these settings:

- Sampling Frequency: 44.1KHz
- Sampling Depth: 16bits (i.e. CD quality, but mono - 1 channel)
- Format: PCM or RAW) - we will cover this in the next section.

Keep the recording to around 10 - 15 seconds, this should give enough time to be able to judge the quality of the sound without the file size becoming too large.

Use ’Save As’ (or method used in your software) to save the wave file using different sampling frequencies and depths, recording you results in the table.

File no.	**Sample Rate**	**Sample Depth (sample Resolution)**	**File Size**	**Quality**
1	44.1 KHz			
2	44.1 KHz			
3	32 KHz			
4	32 KHz			
5	24 KHz			
6	24 KHz			
7	12 KHz			
8	12 KHz			
9	8 KHz			
10	8 KHz			

3.3 Storage of sound data

3.3.1 RAW

Files which store the sound data just as it is received during the sampling process - with no other processing involved are termed ’RAW’ files.

The process of sampling the analogue signal and quantising it (turning it into numbers) depending on the strength (amplitude) of the signal is called **Pulse Amplitude Modulation** (PAM).

When the data from the PAM process is stored in a sequential manner then it is called **Pulse Code Modulation** (PCM). PCM allows the digital sound data to be transmitted, for example over a network or to a sound card, as it can be easily ’decoded’ and used to construct the original sound as it is being received..

PCM is the format of the sound data stored on CD-Audio, DVD-Audio, Digital Audio Tapes (DAT) and often on DVD-Video. AIFF (Audio Interchange File Format) files on Apple Macintosh Computers only contain sound data in raw PCM format.

3.3.2 RIFF files

RIFF (Resource Interchange File Format) is an example of a container file format, that is, it can contain data of various types, for example video, bitmapped sound, midi sound and many others.

RIFF is a multimedia file format. The type of data stored in the file is stated by information in the first part of the file, called the header. For RIFF files that contain bitmapped sound (wave files), the ’format’ part of the header simply contains the letters "WAVE". Wave files (on PCs with the suffix .wav) are RIFF files containing digitised sound (wave) data.

3.3.2.1 Wave files and ADPCM

When a RIFF file contains ’wave’ data then, again, that data could be stored in various formats, RAW-PCM or ADPCM to name two.

Wave files on the Microsoft Windows PC, and many other digitised sound files are commonly encoded using Adaptive Differential Pulse Code Modulation (ADPCM).. ADPCM is used in many other places, for example in digital telephone systems and on Minidisc players.

3.3.2.1.1 Adaptive Differential Pulse Code Modulation (ADPCM)

ADPCM is a codec which is used to compress the sound data. It is a lossy method of compressing the data.

When encoded using ADPCM, it is not numbers representing the strength (volume) of the signal for each sample that is stored, but it is the difference between that sample and the last.

Its compression reduces the disc space requirements for the sound file by about half.

Data encoded using ADPCM is still stored and transmitted sequentially. This means that it can be reconstructed and played as it is being received. As the data is compressed then the bit-rate is lower and a lower-bandwidth network can be used to transmit the sound.

3.3.3 MP3

Digital satellite and DVD movies store the sound and video data in a compressed format called MPEG. MPEG is actually the name of the group that created the standard (The Moving Pictures Experts Group). Part of this system of compressing data involves compressing sound and this is contained in the MPEG specification called **MPEG Audio Layer-3**. This is now known as just MP3.

It has become a popular sound format due to its ability to compress raw sound files to around 10% of their original size.

A 3 minute CD quality song will take up around 30 MB of disc space, encoded using MP3 this same song file will only take up around 3 MB.

Home DVD players, portable players and car CD players are all available with the ability to play back MP3 files.

3.3.3.1 How MP3 works

To achieve such high rates of compression, a lossy compression scheme is used. This means that data is removed from the original sound in order to reduce the file size - this data can never be recovered.

So what data does the MP3 method throw away? This is all based on what humans hear best, called Perceptual Audio Compression.

- High or low pitched sounds which humans cannot hear are removed
- When two similar sounds occur at the same time in a sound file and one is significantly louder, then the quiet one is removed
- The emphasis is placed on the frequencies of sound that humans hear best

3.3.3.2 Quality vs compression

The quality and compression of MP3 files is adjustable. Two main factors are used to control the quality and compression of MP3 files, bandwidth and bit rate.

3.3.3.3 Comparing bitmap image files and digitised sound files

It may be helpful to compare terms when discussing these two file types, as they both have similarities.

Many samples of the analogue signal are taken

Each sample of a bitmap records the data for a pixel. So the image will be split into many pixels in order to get an accurate recording of it. The number of image samples taken (pixels) is called the image **resolution**.

Sound signals are also sampled many times in order to get an accurate recording. The number of samples taken in one second is the sampling **frequency**.

The image resolution and sampling frequency both decide the number of measurements that are taken of a signal.

A number is recorded for each sample

Each time a pixel in an image is recorded using a number, which consists of a certain number of bits. The number of bits determines the number of colours that can be recorded for that image. The number of bits used for each sample is called the **colour depth**.

Samples of sound signals are recorded also using a number, again consisting of a certain number of bits. The number of bits determines the number of signal levels that can be recorded for that sound. The number of bits used for each sample is called the **sampling depth**.

The colour depth and sampling depth both determine the number of levels (dynamic range) that can be used for each sample (i.e. sound sample or pixel) within the signal.

Comparing the quality of sound files

Here you will compare the file sizes and quality of three formats of sound file.

Open File 1 from the sound activity you completed earlier in this topic

By using ’Save As’ in Sound Recorder, or the equivalent method to change the sound format in your software, save this sound using these different formats (re-open the original before changing the format each time). Record your findings in the table.

Format	Settings	File Size	Quality of sound
Wave PCM (RAW)	44.1 KHz, 16 bit		
Wave ADPCM	44.1 KHz, 4 bit		
Wave ADPCM	22 KHz, 4 bits		
Wave ADPCM	8 KHz, 4 bits		
MPEG Layer-3 (MP3)	24 KHz, 56 Kbps		
MPEG Layer-3 (MP3)	24 KHz, 32 Kbps		
MPEG Layer-3 (MP3)	8 KHz, 8 Kbps		

3.4 Technical terms

3.4.1 Bit-rate

As a sound is played, digital signals are constantly having to be converted to analogue in order for us to hear it. The bit rate is the number of bits each second that have to be processed in order for a digital sound to be played.

If the sound is high quality, then there will be a greater number of bits (1s and 0s) as there will be a greater number of samples each second to be converted back to analogue plus each sample itself will be a larger number of bits as the sampling depth will be greater.

If a digital sound is being streamed over a computer network then the number of bits each second is important. If the number of bits each second is very high then this will also place demands on the computer hardware, as it all needs to be processed.

The bit rate for sounds can be calculated as follows:

Bit Rate (bits per second) = sampling depth (bits) * sampling frequency (Hz)

Consider the following examples:

Example 1

A CD quality sound recorded using a sampling frequency of 44KHz and a sampling depth of 16 bits.

Bit rate = 44000 * 16

Bit rate = 704000 bits per second

Bit rate = 704K bits per second

This would need a broadband connection as ISDN lines only allow data transfer speeds of up to 512K bits per second.

Example 2

A low quality sound recording using a sampling frequency of 2KHz and a sampling depth of 8 bits.

Bit rate = 8 * 2000

Bit rate = 16000 bits per second

Bit rate = 16K bits per second

This could be easily transmitted over a dial-up network connection using a modem, as modem speeds transmit data at up to 56Kbits per second.

These figures refer to the streaming of the audio signal over the network. In other words, transmitting the signal over the network in order that it can be listened to as it is sent.

The quality settings of MP3 files specify the bit-rate and sampling frequency to be used. We cannot calculate the sampling depth for this because the data is compressed.

This allows you to select a quality setting based on the speed that the data could be transferred, rather than the end size of the entire file - which is much better for time-dependent data like **sound** or **videos**.

3.4.2 Normalising

A captured signal may not use the full dynamic range that is available. This means that the sound does not use the available range of volumes.

The signal is clearer if the full dynamic range is used, although this not only makes the sound louder, but also makes any noise in the sound louder. Professional musicians use a slightly different technique to this in order to expand the sound to use the full range of volumes, while minimising the effect of noise.

For example, Figure 3.8 shows the wave form of someone speaking:

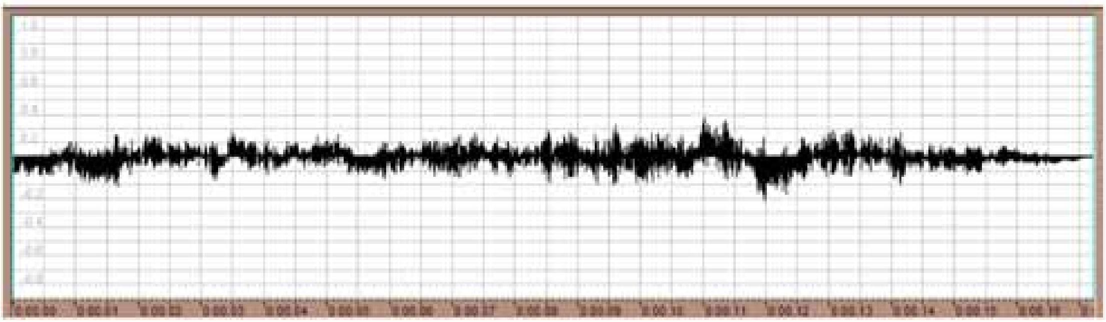


Figure 3.8: Waveform of human speech

The recorded signal is not using the whole dynamic range available.

After normalising the signal (Figure 3.9), it has been 'stretched' and the full dynamic range is now used.

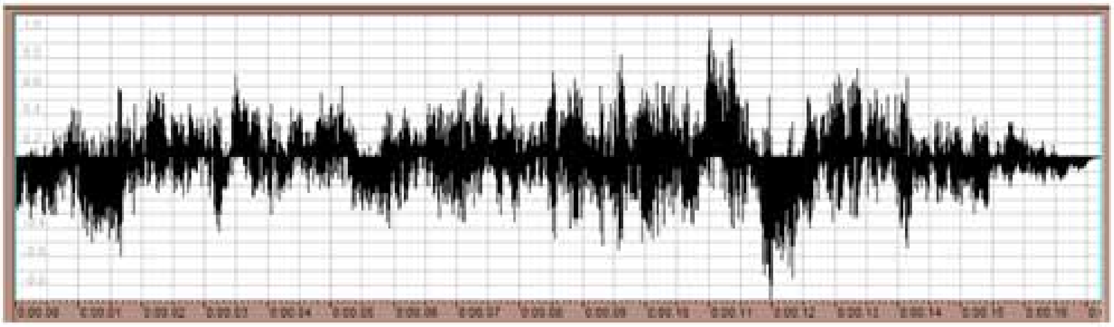


Figure 3.9: Normalised signal

This edited sound is now louder overall and should be clearer than the original.

Although normalising the signal improves it, as the noise is also increased in volume, it is best to record the sound using as wide a dynamic range as possible to begin with. The problem is that if the sound is recorded too loud then the sound becomes distorted due to clipping.

Some programs have the ability to automatically adjust the gain of a sound as it is recorded, to try and maximise its dynamic range while avoiding clipping - this is called Automatic Gain Control or AGC.

3.4.3 Clipping

If the volume of a sound signal is beyond the dynamic range then clipping occurs. This can be caused by recording a sound using too high a gain setting, if the sound is louder than expected or if the volume of the sound is edited and increased beyond the dynamic range of the file.

Sometimes clipping can be introduced for a special effect, but it is usually unwanted, as it reduces the quality of the sound, which just sounds unclear and distorted.

For example, Figure 3.10 shows part of a waveform of someone saying 'hello':

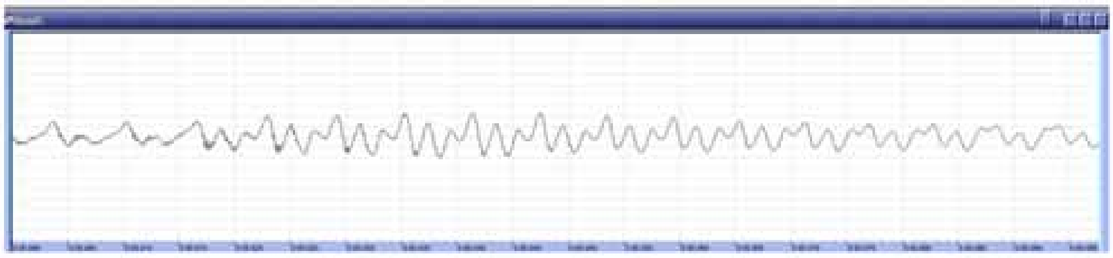


Figure 3.10: "Hello"

Figure 3.11 shows the waveform as the volume is increased (this sound has been normalised):

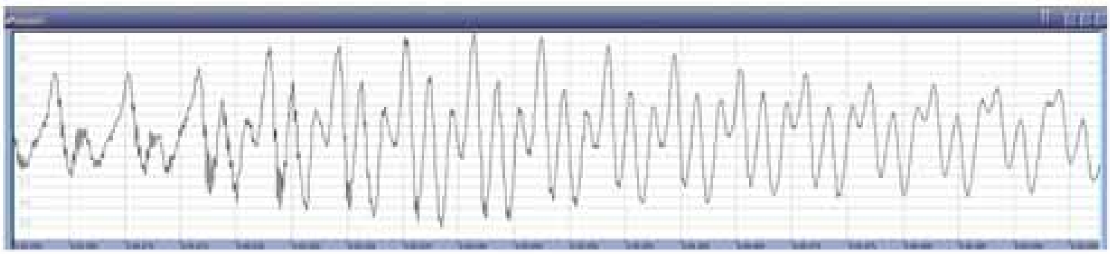


Figure 3.11: "Hello": normalised

Figure 3.12 shows the waveform showing clipping, as the volume has now been increased beyond the dynamic range of the file:

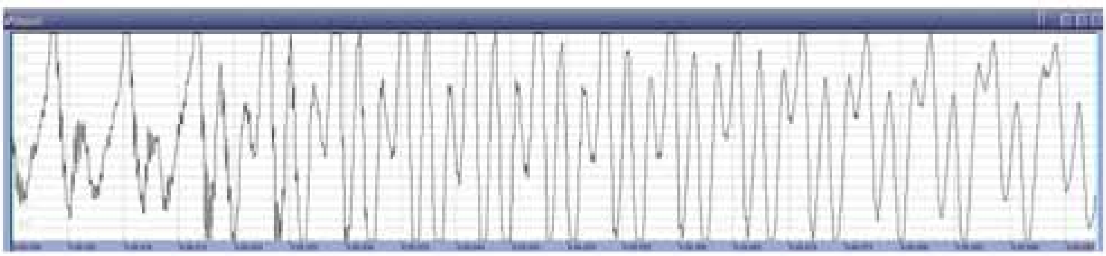


Figure 3.12: "Hello": clipped

Clipping can be spotted by looking for flat areas at the top or bottom of the waveforms (Figure 3.13).

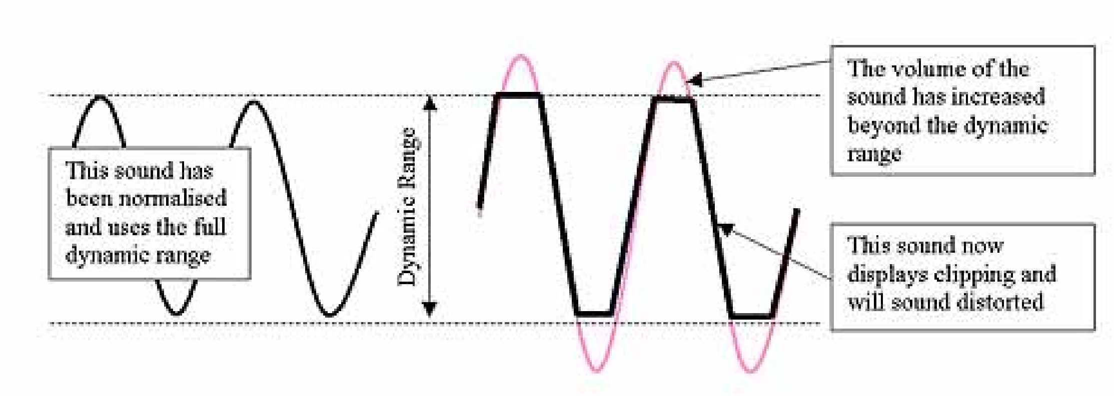


Figure 3.13: Identifying clipping

3.4.4 Stereo

Mono sound only contains one channel of sound, or one audio track. Mono is actually short for monaural sound. Stereo (or stereophonic) sound contains many tracks or channels of sound, binaural sound is the term for sound that contains two channels.

Through common use, stereo sound has become to mean the same as binaural sound, so this is what we will use here.

A stereo sound is actually two separate sound tracks. In order to record in stereo you need two microphones and two inputs to the sound card - most sound cards have a stereo input anyway, that is, a single socket that allows two microphones to be connected to it, or allowing the two sounds from a stereo source to be captured. These two channels within a stereo sound file are known as **left** and **right** - referring to the ear that each is intended.

As a stereo sound is just two separate sounds in a single file they take up twice the disc space of a **mono** sound file.

When editing a stereo sound in a waveform editor, both sounds are displayed as separate waveforms:

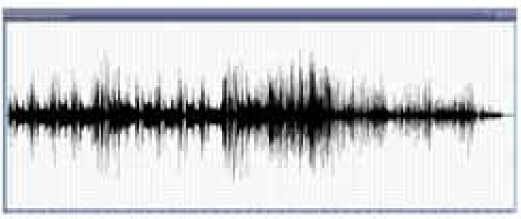
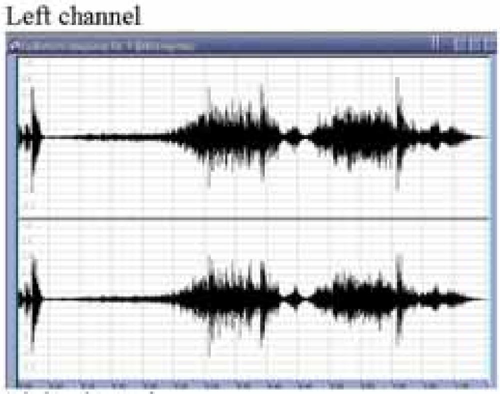
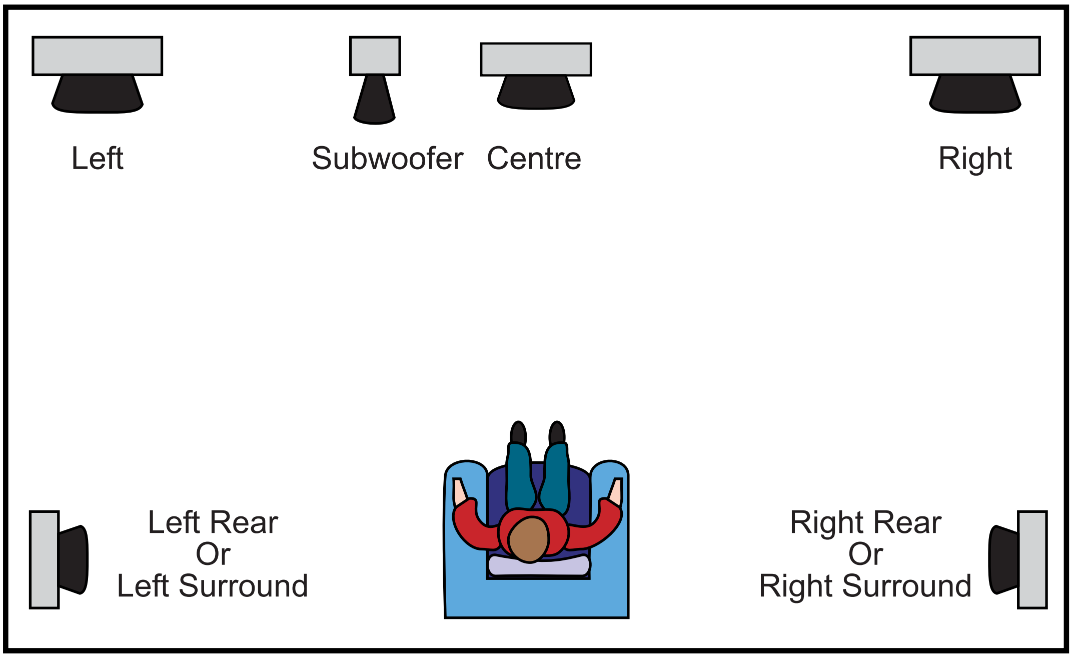
Editing a mono sound	Editing a stereo sound
	Left channel  Right channel

Figure 3.14: Editing mono and stereo sounds

3.4.5 Surround sound

Most surround sound systems are termed **5.1 systems**. The speaker setup is usually as follows:



DVDs, digital television broadcasts and computer games often include surround-sound. There are two main standards for surround sound: Dolby Digital, DTS (Digital Theatre Systems). Dolby Digital is the standard on these systems with DTS, for example, being optional on DVDs.

Surround sound makes the film, game or multimedia application more immersive and life-like.

The '5.1' refers to 5 main speakers - see the above diagram, with an extra subwoofer (the '.1' bit). The subwoofer channel is only for low frequency effects (LFE) like making the room rumble when there is an explosion, or a dinosaur is approaching. This channel is called .1 because it only needs to carry sound data on the very low frequencies, so has a much narrower bandwidth than the other 5, proper, channels.

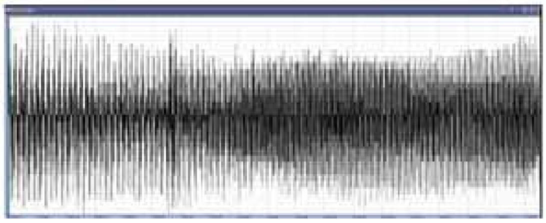
So we need to have 6 channels of sound for a proper surround sound setup. Digital surround sound simply saves 6 separate channels of sound, in the same way that a stereo sound file contains 2 channels of sound. DTS surround supports up to 8 separate audio channels.

Only compression can allow this much sound data to be used, otherwise the bit-rates would be just too high for most equipment to cope with. A lossy compression codec is used that has similarities to MPEG (it uses perceptual compression) but uses a combination of methods. These surround sound codecs can support sampling frequencies of up to 192KHz with sampling depth of 24bits, but using compression these high-quality sounds can achieve the same bit-rate as standard audio CDs.

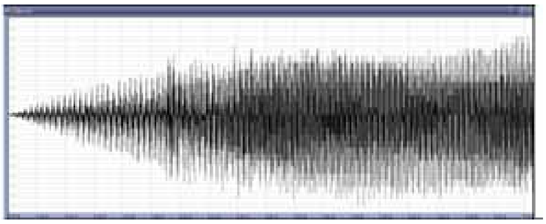
Surround sound decoders have hardware surround sound codecs that decode this audio stream and generate all the necessary channels of analogue sound for the speakers.

3.4.6 Fade

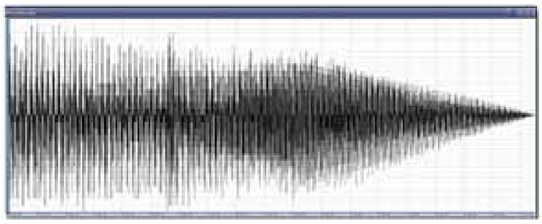
Fade is the term applied to a change in volume. It is used to prevent abrupt starts and finishes of sounds.



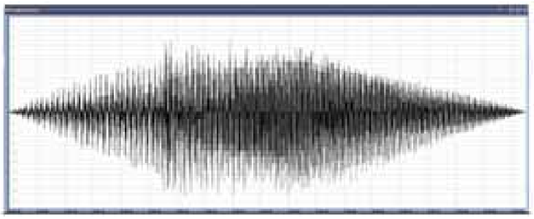
A short sound with an abrupt start and finish.



To prevent an abrupt start a **fade in** is used.



To prevent an abrupt finish a **fade out** is used.



The sound has a fade-in and a fade-out applied.

3.5 Calculations

The size of an uncompressed audio file can be estimated using the following calculations:

File Size (bits) = Sampling Frequency (Hz) x Sampling Depth (bits) x Length of sound (s) x channels (1 for mono, 2 for stereo).

$$\text{File Size (bytes)} = \frac{\text{File Size (bits)}}{8}$$

Example

A stereo song has to be recorded at CD quality. The song is 3 minutes and 12 seconds long. How much disc space would the captured song occupy?

Settings for CD quality are:

- Sampling frequency = 44100 Hz (44.1KHz)
- Sampling depth = 16 bits

Stereo so:

- Channels = 2

Length of song in seconds: = (3 × 60) + 12 = 192 s

File Size (bits) = 44100 × 16 × 192 × 2 = 270,950,400 bits

File Size (Bytes) = 270950400 / 8 = 33,868,800 Bytes

File Size (Kilobytes) = 33,868,800 / 1024 = 33,075 KB

File Size (Megabytes) = 33,075 / 1024 = 32.3 MB

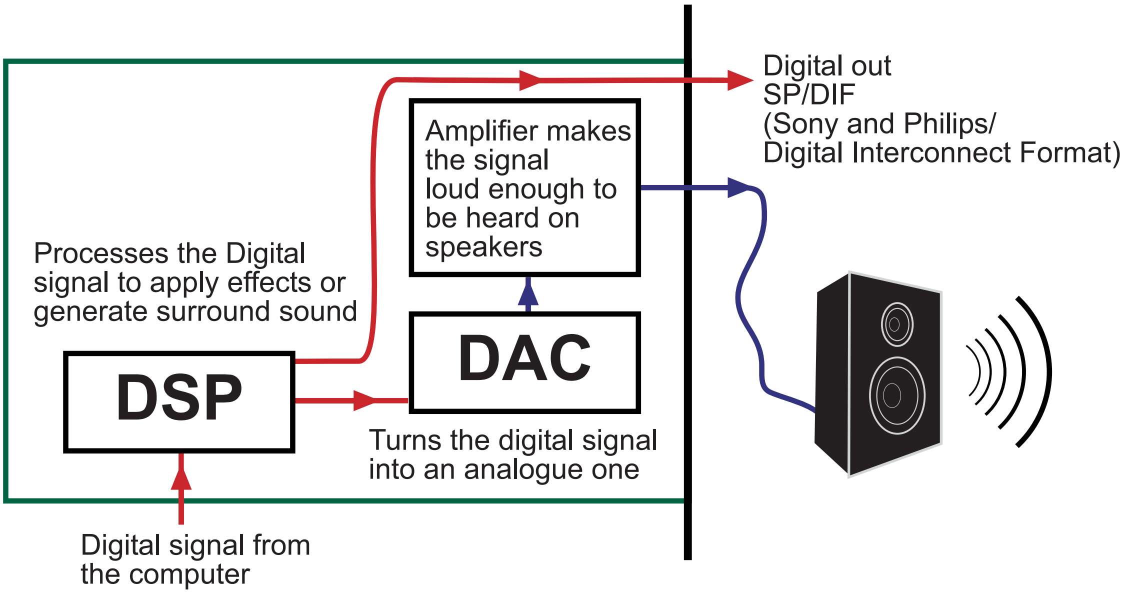
Calculating file sizes

Calculate the file sizes for files 2 - 10 in the earlier sound activity in this topic, if the settings in that table were used to capture a stereo song 3 minutes and 12 seconds long, as shown above.

Using knowledge from this section, prepare and add a digitised sound file to one of the multimedia applications you created in Topic 1.

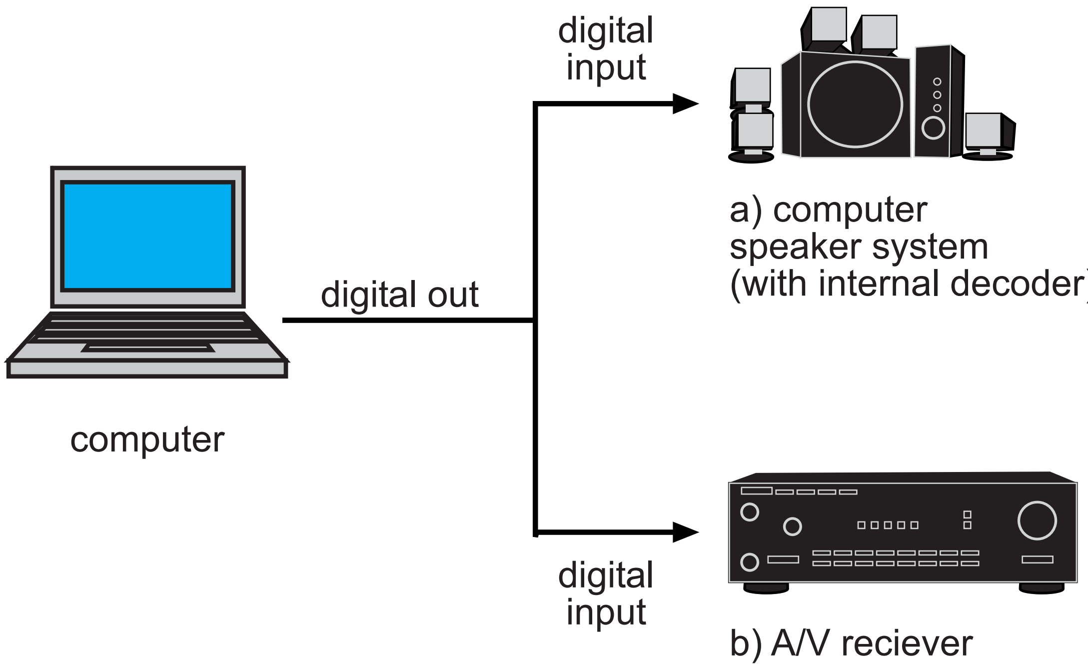
3.6 Output hardware

A soundcard is needed to generate sounds that can be fed to speakers or headphones in order that we can hear it.



Many modern soundcards include the ability to decode surround sound audio. The **DSP** does this and there are sockets for extra speakers.

Alternatively the digital surround sound signal can be sent directly to an external surround sound decoder such as this.



3.6.1 Digital to Audio Converter (DAC)

The Digital to Audio Converter in a sound card is used to change the digital sound signal into an analogue signal that can be sent to loud speakers.

3.6.2 Digital Signal Processor (DSP)

The digital signal processor or DSP is used to process the sound signal as it is captured or as it is output. The DSP in a soundcard can apply effects to sounds. If the soundcard does not have a DSP, then any processing needs to be done by the computer's CPU, slowing down the overall system.

3.7 Questions

Q1: Which of these describes the term 'Sampling Frequency'?

a) The number of times each second that a measurement is recorded of the sound
b) The number of times each second that the sound signal changes (oscillates)
c) The number of bits used to store each sample
d) This is the format that is used to store the file

Q2: Write down a meaning of the term 'Sampling Depth'.

Q3: What Sampling Depth is needed to capture a sound in order to give it a maximum dynamic range of 65536 levels?

a) 4 bits
b) 8 bits
c) 16 bits
d) 24 bits

Q4: Calculate the dynamic range for the other 3 answers in the above question.

Q5: Most of the common compression codecs used for audio data are lossy. What could be a reason for this?

a) This makes the audio sound better quality
b) Audio data files are so large that only lossy compression produces reasonable file sizes
c) Audio files cannot be compressed using any form of non-lossy compression
d) It takes too long to compress the data using non-lossy compression

Q6: Find out what audio codecs are supported by your audio software.

Q7: A song is recorded at 44.1KHz and 16 bit CD quality and stored using PCM. Which of these would have greatest effect in reducing its file size?

a) Sample it using 20KHz instead of 44.1KHz
b) Sample it using 8 bits instead of 16 bits
c) Store it in MP3 format (keeping equivalent audio quality)
d) Store it using ADPCM (keeping equivalent audio quality)

Q8: Calculate, or find out by creating the files, the sizes of all five audio files mentioned in the above question.

Q9: A sound is captured, which of these will ensure that the sound uses the whole dynamic range available, without distorting it?

a) Save using a non-lossy codec
b) Use a greater sampling depth
c) Apply clipping
d) Normalise it

Q10: Explain what effect the other three answers in the above question would have on an audio file.

Topic 4

Video data

Contents

Prerequisite knowledge

Before studying this topic you should have:

- *Basics of using video editing software (e.g. Windows Movie Maker)*

All topics require:

- *familiarity in working with the computer filing system - i.e. the ability to create, view the properties of, save and open files and folders;*
- *the ability to locate and run the necessary software for each topic;*
- *the ability to switch between multiple applications on the computer system;*
- *knowledge of units of storage (bits, Bytes, KB, MB);*

- *knowledge of the basic components of computer systems (mostly just input, output and backing store devices).*

Learning Objectives

By the end of this topic you should be able to:

- *Describe the hardware required to capture video data;*
- *Describe the structure of video files;*
- *Describe a method of compressing video files;*
- *Perform calculations involving the size and bit rates of video files;*
- *Use technical terms involved in the editing of video files;*
- *Describe the hardware factors involved in the output of video data.*

4.1 Introduction

A video is just a collection of bit-mapped images that when played quickly one after another give the illusion of a moving image.

As video files are then just lots of bitmap images, you should be familiar with Topic 2 on bit-mapped graphics before starting this topic.

We will concentrate on how video files store image data but you should remember that video files also store the sound data alongside the image data in the same file.

Just like a flick-book, a video file is simply a collection of images, except that sound is nearly always included.

Animations are often stored as video files. The only difference between a frame-by-frame animation like this one and a video is that each frame of the animation is drawn, whereas each frame of the video is captured.

Animated GIF files can store videos, but the restriction of 256 colours makes the quality very poor - this restriction has no effect here when the animation only consists of 3 colours anyway.

The LZW compression in GIF files shows its benefits for this type of image with few, flat areas of colour.

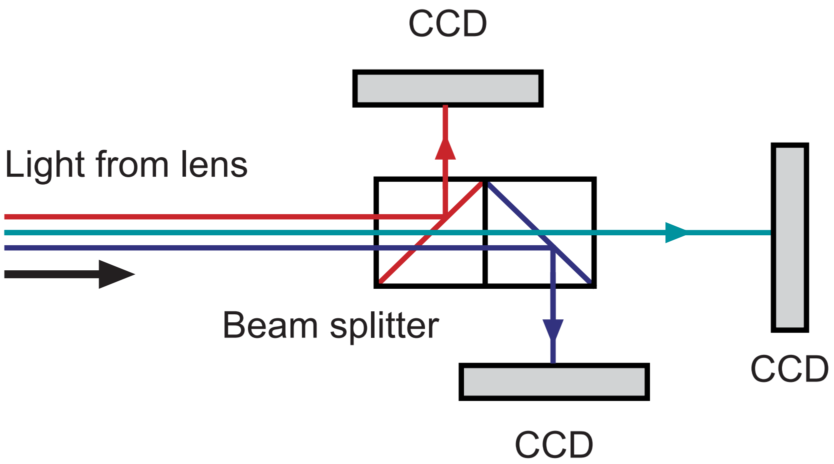
4.2 Input (capture) hardware

4.2.1 Digital video camera

Digital video cameras (from now on termed ’camcorders’) work in the same way that digital cameras (from now on termed ’cameras’) do. They use the same light sensors, called CCDs. The main difference is that camcorders have lower resolution CCDs than cameras and are designed to capture lots of images very quickly, capturing lots of images (frames) to form a film. For camcorders to be able to do this the ADC needs to process a lot of information very quickly, so has to be a lot faster than the ADCs used in cameras.

Camcorders do not need as high a capture resolution as cameras as the images are only designed for display on televisions or on the film screen. Moving images do not need as much detail as a still image for us to perceive similar qualities.

In order to capture the RGB image, most ’consumer’ camcorders use a colour filter as in digital cameras this produces a poorer quality video than the method used by professional camcorders. Professional camcorders use 3 CCDs, each with a different filter of red, green or blue, the images from these 3 CCDs are combined to produce the full colour image. As 3 CCDs are used, rather than 1, the price of these professional camcorders is greater. (Some professional digital cameras also use this method to capture still images)



A note on convergence

Modern digital cameras include the ability to record video, albeit at lower resolutions, often including sound. They can only capture videos at lower resolutions, even although their CCDs are capable of much higher resolutions, as their ADCs cannot cope with the high bit rates required to capture higher resolution videos. Another factor preventing the use of digital cameras for capturing videos is that the storage requirement for video data is much greater than of still images and digital cameras are not usually equipped with the necessary storage space for capturing videos of any useable length.

Likewise digital video cameras often have CCDs capable of capturing images at higher resolutions than used for video (currently some have 3MP CCDs). These high resolutions are only used when capturing still images. So digital video cameras can often be used as a medium quality digital camera for capturing still images.

Perhaps by the time you read this a device will be available that can capture high quality still images and also broadcast quality videos?

4.2.2 Webcam

Webcams use low resolution array CCDs and low quality lenses, keeping costs down. As the videos captured are low resolution, there is not the need for a fast ADC as the bit rates will be lower.

A typical resolution of videos captured from a webcam is 640 x 480 at 30 frames per second (fps).

As webcams are designed and sold for the purpose of creating videos to be transmitted over the Internet, they do not need to be high resolution, keeping down the bit rates and price.

Webcams, as with digital video cameras, can also be used to take low-quality still images. The low-quality components means they are no challenge to digicams or cameras, however the quality is often more than adequate for the purpose of streaming over networks.

4.2.3 Video capture card

In order to capture videos from analogue sources such as video tape players, television broadcasts and analogue video recorders, a video capture card is needed. Some modern graphics cards include the ability to capture video, but for the best results a special video capture card is needed.

Specialised video capture cards offer several advantages over using a standard graphics card with video capture (input) capability:

- They often have faster ADCs and can capture videos at higher bit rates (i.e. better quality)
- They usually capture sound as well. This enables the sound to be fully synchronised with the video. Standard graphics cards have to be used in combination with a separate sound card, which can cause discrepancies between the video and sound tracks
- They usually have a hardware codec to allow the video to be processed and stored in a compressed format as it is being captured.

4.3 Storage of video data

Here are the approximate settings needed to capture a film to be shown on analogue television or film. Note, as these are for analogue systems, there are no pixels but the approximate resolution of each system has been calculated to be as shown below.

These are the settings you would need to capture a video in the given format at in order to maintain the quality.

	Resolution	**Frame Rate (images each second)**
Current analogue UK Television (PAL)	768 x 576	25 fps
Current USA Television (NTSC)	640 x 480	30 fps

Modern video systems are digital and improve on the image quality of analogue systems, even if the image resolution is not any greater. In digital video systems there is no snow or ghosting interference so the pictures look clearer.

Video systems of the future will support greater resolutions than these. The Star Wars Episode II film was shot on digital camcorders using an image resolution of 1920 x 1080 and 24fps.

Most camcorders are not capable of resolutions this high, these only being readily available to professional television and film makers due to their high cost.

4.3.1 Uncompressed AVI

Audio Video Interleave (AVI) files are actually a type of RIFF file (see topic 2 for more on RIFF files). As such, this container file can store videos in a variety of formats as defined by information at the start of the file (in the header).

Uncompressed video files are so large that they are almost never used. AVI files are generally used to store videos in more compact formats than un-compressed data.

The name Audio-Video Interleave is a reference to the way that audio and video data are stored in this file. When saving AVI files the audio-video interleave ratio can be set (for example embed audio data every 3 frames of the video data). This just means that the audio and video data is mixed up in the data file. This enables the videos with the sound to be played as the file is being received without having to transmit the entire file before it can be viewed.

4.3.2 MPEG

The Moving Pictures Experts Group (MPEG) have defined a series of standards for compressing video and audio using compression based on DCT (Discrete Cosine Transform) (see JPEG images). Each frame in an MPEG video is compressed as a JPEG. The data that stays the same in successive frames is then removed.

There have been a series of standards based on this:

MPEG-1 (VHS video quality with 353 x 240 pixels and 30 fps frame rate support)

MPEG-2 (The standard for DVD-Video and Digital Television -to name two. Widely used)

MPEG-3 (Intended for HDTV but these revisions were incorporated into MPEG-2) (Not the same as MPEG-Layer 3, or MP3 used for audio - this is actually the audio subset (layer) of the MPEG-2 standard)

MPEG-4 (Designed for low-bandwidth networks - e.g. video phones) (Part used by DivX)

MPEG-7 (Builds on the interactive and extra data capabilities of MPEG-4 and is a full multimedia description format) - named "Multimedia Content Description Interface"

4.3.2.1 How does MPEG work?

Not all frames are stored - just a few key frames called ’i-frames’. These are JPEGs. The next set of frames does not store images, they just store data on what has changed since the last i-frame.

First (i) frame which is the base for the following (b) frames

A scene which can use most of the inform-ation form the first

A new scene which requires a whole new (i) frame

MPEG does not store each image separately, only key frames are stored as JPEG images, the rest of the data consists of predictions or actual changes since the last (or next!) key frame.

MPEG is a lossy compression codec and, as with JPEG images and MP3, has adjustable compression depending on the desired quality, file size or bit-rate.

There are different implementations of the MPEG-4 codec (for example). The playback compatibility and compression/quality gained depends on the actual version of the codec that is being used.

4.4 Calculations

As video files are simply a collection of bitmap images, all we need to do is calculate the size of one image (i.e. one frame) and multiply that by the number of frames in the entire video.

File Size (Bytes) = Frame Size (Bytes) x Frame Rate (frames per second [fps]) * Video Time (s)

As video data is time-dependent, the term bit-rate is often applied to it.

Activity

Calculate the Bit-rates for the two video files shown in section 4.2.1

Use the above formula to calculate the size of the uncompressed file - verify that it matches (approx) the actual size of this file.

The uncompressed file was initially captured using 24 bit colour depth. What would have been the size of this file?

What would the file size have been if it was captured at appropriate settings for display on UK TV? (24bit colour depth)

These figures are all for a nine-second video clip. Perform the calculations again, this time for a 3 minute 12 second video (long enough to store a music video for instance).

These calculations above are all regarding a video file without any sound. What would the uncompressed file size of the 3 minute 12 second music video be if it were recorded for maximum quality on UK television with CD quality stereo sound in PCM format?

Find or create a video file, you could use one of the video files from this section of work, capture your own video or use any other video file

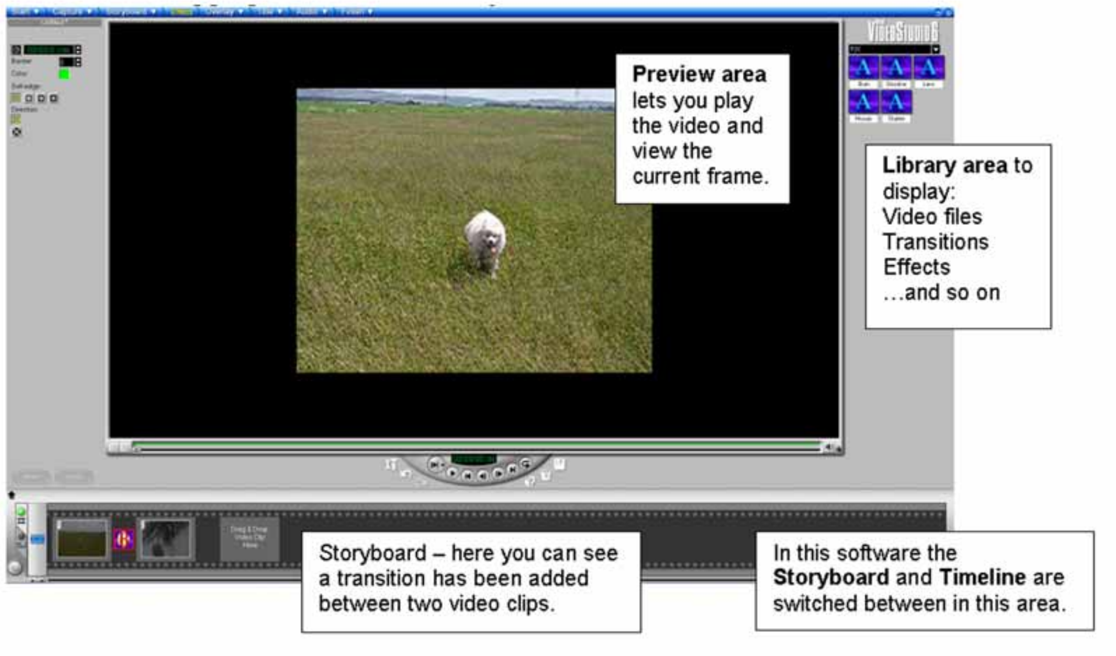
- Open up the video file in video editing software
- Use 'Save As' (or the method appropriate to your software) to save the video using different settings. Remember to reload the original before each save.
- Fill in this table showing your findings

Codec used	Frame Size	Frame Rate	Colour Depth	Duration (all the same)	File SIze	Quality

Try at least 2 different codecs (uncompressed AVI and MPEG?) Use 6 different settings to see the effect on file size and video quality.

4.5 Technical terms

Most video editing programs have a similar layout:



4.5.1 Timeline

The timeline is a key item in a video editor. Here, each frame of the video is displayed as a thumbnail image. This allows each frame to be manipulated separately or overlays and effects applied to a group of frames.

An overlay could be another video, animation, title or effect such as a colour gradient. Effects can also be applied to a single or number of frames - effects that are available in bitmap editing programs are also available to be applied to video.

Figure 4.1: A timeline

A library of effects that can be applied to frames on the timeline.

4.5.2 Transitions

Transitions are effects that are used to join two video clips (or shots) together. Common transitions are Fade and Wipe.

4.5.3 Sequencing

Sequencing just means joining together video clips into different orders. This is done on the 'Story board' of the video editor as shown above. Transitions can be added to connect different clips on the storyboard.

Applying effects to videos

Open two videos in editing software, apply a transition, titles and an effect. Save them as MPEG video files.

4.6 Output hardware

As mentioned in the topic on bitmap graphics, the graphics card is responsible for the output of image data.

Due to the complex codecs and high bit-rates needed to output video data, video output cards used to be needed in order for a computer to be able to display videos at full quality.

Graphics cards have become much more powerful (driven mainly by the games market) and any modern graphics card should be able to comfortably decode and display full-quality videos on a computer.

4.6.1 Digital to Analogue Converter (DAC)

While most people have analogue displays (LCD displays with digital (DVI) inputs are available) then graphics cards must be able to generate the analogue signals needed for the monitors.

The DAC changes the digital video signal in the computer into an analogue video signal that the monitor can use to display the image.

4.6.2 Digital Signal Processor

The Digital Signal Processor (or **GPU**) plays a key role in allowing computer to display full-quality videos.

The GPU on the graphics card is responsible for decoding the video signal and may even have hardware codec built in. There are fewer calculations needed to output video data than to capture it (it is an asymmetrical process) - for MPEG video files.

Preparing a video file

Using knowledge from this unit, prepare a video file and insert it into one of the multimedia applications you created in Topic 1.

4.7 Questions

Q1: Which of these does **not** describe a why a video capture card is a better choice for capturing video data than a standard, all-round graphics card with 'video in'?

a) The 'video in' on the graphics card does not allow analogue video to be captured
b) Video capture card has a faster ADC and can capture the video data at higher quality
c) Video capture card also allows sound to be captured, which the standard graphics card does not
d) Video capture cards can compress the video data as it is received

Q2: Look up a computer catalogue (paper or web site - e.g. www.dabs.com) and compare the features of a standard graphics card with video capture (look for VIVO - video-in, video-out) and compare the price and features with a Video capture card. Create a table to summarise your findings.

Q3: Highest quality PAL television is approximately: 768 x 576 pixels at 25fps in 24bit colour. What bit-rate is needed to broadcast this, uncompressed, over a digital network?

a) 52Kbps
b) 4Mbps
c) 32Mbps
d) 440Mbps

Q4: What bit-rate is needed to broadcast NTSC television?

Q5: AVI (Audio Video Interleave) is a file format that can be used to store video. Which codec is used to encode the video data in an AVI file?

a) It is uncompressed, so no codec is needed
b) MPEG
c) AVI
d) Any, information on the codec to use is contained in the header of the container AVI file

Q6: Explain how MPEG codec works.

Q7: A video is captured using 24fps and 600 x 400 pixels, colour depth 24bits. Which of these settings would **half** the file size of the uncompressed video?

a) 24fps, 600 x 400 pixels, 12 bits
b) 6fps, 300 x 200 pixels, 12 bits
c) 12fps, 600 x 200 pixels, 24 bits
d) 12fps, 300 x 200 pixels, 24 bits

Q8: Calculate the bit-rates and file sizes of all five files in the above question.

Q9: When a news report is shown it appears by sliding down from the top of the screen. What has been used to create this effect?

a) A transition was used
b) Sequencing was used
c) The frames were edited on the timeline
d) The video was saved using a special codec

Q10: What is the difference between an effect and a transition? Make a list of all the effects and transitions supported by your video editing software.

Topic 5

Object Oriented Data

Contents

Prerequisite knowledge

Before studying this topic you should have:

- *Familiarity with drawing software (e.g. AppleWorks Draw)*
- *Familiarity with painting software (e.g. AppleWorks Paint)*
- *Familiarity using a text editor (e.g. notepad)*
- *Familiarity with web browser and ability to open local files (with SVG and VRML capability)*
- *Familiarity with geometry and the co-ordinate system*
- *Basics of using MIDI editing software*

All topics require:

- *familiarity in working with the computer filing system - i.e. the ability to create, view the properties of, save and open files and folders;*
- *the ability to locate and run the necessary software for each topic;*
- *the ability to switch between multiple applications on the computer system;*
- *knowledge of units of storage (bits, Bytes, KB, MB);*

- *knowledge of the basic components of computer systems (mostly just input, output and backing store devices).*

Learning Objectives

By the end of this topic you should be able to:

- *Describe common features of object oriented file types;*
- *Give examples of objects from object oriented data files;*
- *Describe factors affecting the output of object oriented data;*
- *Describe methods used to create object oriented data files;*
- *Describe the conversion between object oriented and discrete data types;*
- *Compare object oriented data with the associated discrete data format;*
- *Demonstrate an understanding of the structure of SVG files and associated attributes;*
- *Demonstrate an understanding of the structure of VRML files and associated attributes;*
- *Describe the techniques involved in editing MIDI data and attributes involved.*

5.1 Introduction

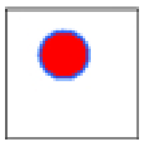
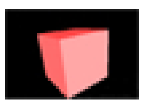
Object oriented data types are quite different from digitised data. Object oriented data is usually created on the computer and is not normally captured by sampling real world information.

As the name suggests, these data types store details on the properties (attributes) of each object that is stored in the file. Each object can be altered or edited simply by changing its attributes. For example, here are some object oriented data types, with examples of how they can be changed by editing the attributes:

- **Text**: a character in a text file can be given a different size, colour or font
- **MIDI** music: a note in a MIDI sound can be given a different volume, length or pitch
- **Drawing**: a shape in a drawing can be given a different size, fill colour or layer
- **3D image**: an object in a 3D image can be given a different location, rotation or texture

All vector data is stored as a group of objects within a file. Each object can be edited individually without affecting any others in the file.

All aspects of each object are decided by its properties or attributes. As object oriented files are just collections of objects with details about them (attributes) then they can often be stored as plain text files. For example:

Type of data (*example of object*)	**Example of object oriented storage giving details on some attributes**	**How the data would be displayed**
Text (*character*)	<u>H</u>ello This is HTML data	Hello
MIDI (*musical note*)	Note Start=00 End=01 Vol=50 Pitch=C# Instr=01 This is an example of how MIDI data could look if converted to text.	Would play a C-sharp note 1 second long at 50% volume using instrument 1 (usually a piano)
Drawing (*a shape*)	circle cx="60" cy="50" r="20" fill="red" stroke="blue" This is SVG data.	 A red circle with blue line at position 60 along, 50 down and radius 25.
3D Image (*a 3D object*)	Shape { appearance Appearance { material Material {emissiveColor 255 0 0} } geometry Box {} This is VRML data.	 A box shape of standard size, colour glowing red.

5.2 Input and output

Input/creation of object oriented data

All the above could be created using nothing but a simple text editor, but there are specialised input devices used to help users create and edit object oriented data on computers:

- We are familiar with keyboards to help us enter text
- To create drawings or 3D images a graphics tablet and puck provides the designer with a more accurate method of drawing than using a mouse
- To enter musical notes a MIDI keyboard is easier than using the computer keyboard or mouse to enter them on screen
- 3D digitisers can measure points on an object and generate a 3D object from the data

Output / display of object oriented data

The quality / appearance / sound of object oriented data is very dependent on the hardware or software used to generate / output it.

Text	Web browser settings can cause the same text to be displayed in different ways on different computers. For example 'Tools - Options' in Internet Explorer will allow you to change the font / colour of standard text on web pages.	Affected by the software
Drawings	These are generated by the software - probably the easiest to keep consistent across different computers of the 4 data types named here.	Affected by the software
3D images	Graphics cards with special hardware codecs for the rendering of 3D images on screen are available. Each may produce a different quality of image and support different features. For example one may not display fog, while another does.	Affected by the software
MIDI sound	The soundcard generates the sound. Two soundcards could make the sound very different. For example one soundcard could make a very realistic piano sound, the other could only produce a very synthesised sound.	Affected by the software

5.3 Converting between Discrete and Object Oriented data types

Converting from Object Oriented data

It is always possible to convert from an object oriented file **to** a discrete (bitmap) file type. This is sometimes called **Rasterisation**. Converting data to a discrete format is the equivalent of capturing the data (see sections on digitised sound or bitmap image), except that no input hardware is needed.

In these conversions, remember all data about each object in the file is lost. For example the text can no longer have it font changed, the layers of shapes cannot be changed, 3D objects cannot be rotated and individual musical notes can no longer be edited.

Object Oriented data	...how it can be converted to ...	Discrete data
2D Drawing	Often available in the ’Save As’ or ’Export’ menu of the drawing program - just choose a bitmap file type.	Bitmap image
3D Image	This is called ’rendering’. The software uses the 3D file to generate a flat (2D) image from a given point of view. As monitors display 2D images then this is what 3D graphics cards have to do. Their special hardware codecs allow them to do this very quickly.	Bitmap image
Text	A picture of text is often obtained scanning in printed text, but can also be achieved by capturing an image of text on the screen by doing a ’screen shot’ (e.g. pressing the ’Print Screen’ button and pasting the image into bitmap image software.	Bitmap image
MIDI music	Often available using ’Save As’ or an ’Export’ function in the MIDI software but you can also ’record’ the sound as a MIDI file is playing - the sound can be recorded internally, within the computer.	Digitised sound file

Converting to Object Oriented data

It is very difficult to convert discrete data into an object oriented type because conversion must recognise each object from the discrete (bitmap) data and, for example, generate a mathematical rule describing the shape of that object.

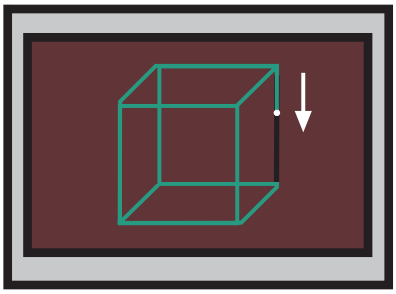
You will not always end up with an accurate version of the data that was converted. This is sometimes called vectorisation. This requires ’intelligent’, pattern-matching software.

Discrete data	...how it can be converted to ...	Object Oriented data
Bitmap image	The software must spot the patterns that make up single objects. (for example 'WinTopo')	2D Drawing
Bitmap image	With the user's help, to 'tell' the software where important points and features are, generates 3D objects from photographs or other bitmap images. (e.g. 'PhotoModeler')	3D Image
Bitmap image	This is called 'OCR' - Optical Character Recognition. The software uses pattern matching to recognise shapes of letters in the image and to convert them to text.	Text
Digitised sound file	Must 'recognise' musical notes and chords but cannot (yet?) recognise imstruments. (for example 'intelliScore')	MIDI music

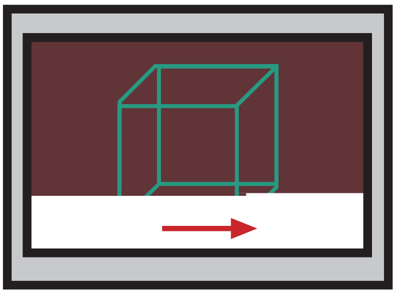
5.4 Vector graphics data

A vector is a mathematical term for a line with a specific length and direction. Computer displays did not used to be able to show images (they displayed text characters only) and the only displays capable of graphics were specialist vector displays. The electron beam (which scans across our televisions and computer displays building up the picture) did not scan across the image like current displays but drew out shapes directly on the screen (like an oscilloscope). Vector images contained the data to tell the electron beam which path to take, so vector images contained data for the creation of line images (containing real vector data - in the mathematical sense)

Vector displays use the electron beam to draw out the shapes. (no colour).



A (now common) raster monitor. Creates the image by drawing rows of dots.



The term vector graphics has been derived from these vector image files.

5.4.1 Comparison of Vector and Bitmap Graphics

Vector		**Bitmap**	
Can be scaled to large sizes, keeping original quality.	☺	Scaling causes pixellation.	☹
Individual objects can be edited, allowing an object to be altered without affecting the rest of the image.	☺	Only the image as a whole can be edited.	☹
Are easily converted to bitmap formats.	☺	Are very difficult to convert to vector formats, with unpredictable results	☹
File sizes are relatively small.	☺	File sizes can be large.	☹
Difficult to create realistic images	☹	Images can be very realistic (e.g. digital photograph).	☺
Size of image can be increased keeping quality and file size the same.	☺	Increasing the image size needs re-sampling and increases the file size.	☹
Only individual objects can be edited (it is sometimes impossible to edit only part of the object).	☹	Pixel level editing is allowed - allowing effects such as spray paint, blur, effects and so on.	☺
Dependent on output hardware or software for appearance & quality.	☹	Same appearance in all systems, regardless of hardware or software.	☺
Suitable for graphic, unrealistic images and designs.		**Suitable for natural, hand-drawn looking, realistic images.**	

Exploring graphic formats

Create an image in a drawing program - say design a logo for... a new rock band!

Save the image in a drawing format and in a bitmap format - give each a slightly different name.

Open each and try to edit both the images in these ways:

- Change the line thickness of a shape
- Create a new shape and place it behind other shapes in your image
- Make a blur effect so that part of the images look out of focus
- Edit the images, adding paintbrush effects etc, to make the logo appear as if it has been hand painted in art class.

5.4.2 Object Oriented Data Storage

5.4.2.1 Drawing

Drawing files store details on each shape within the 'image'. Here are some objects that may exist within drawing files (there are plenty more) and examples of attributes that each object may have.

5.4.2.1.1 Common attributes

Attributes are the properties that determine what and how objects will appear.

Attributes

Start a Drawing program (for example the Drawing software within MS Office Suite is very powerful). Drawing programs can be very similar and similar objects will have similar attributes in different software. Record the attributes that apply to each object in the table.

	Attributes														
Object	Layer	Start Point	End Point	Line Thickness	Line Colour	Line Pattern	Centre Point	Location	Width	Height	Fill Colour	Fill Pattern	Number of sides	Rotation	
Line															
Point															
Circle / ellipse															
Square / rectangle															
Rounded rectangle															
Polygon															
Path															
Bitmap image															

You will notice that Drawing files can include a bitmap image as a separate object - some even let you edit this image using a bitmap editing program built into the drawing program (for example, AppleWorks allows this, just use the PaintBrush tool to add a bitmap image to a drawing and see the tools change to bitmap editing tools, click away from it and they change back to Drawing tools).

Look carefully at a drawing program and find 3 other objects that it supports, find one extra attribute that could be included in the above table.

In your drawing program, add one or more of each of the above objects, make sure you can alter all the attributes that you ticked for each object in the table.

5.4.2.1.2 Drawing file formats

There are many types of drawing file, for example windows metafile (wmf), cgm or drawing exchange file (dxf), but most are binary, proprietary formats. This caused a problem for web graphics as there many advantages to using vector images on web pages - scalability and file size to name two, but there was not an openly available image standard that would allow vector images to be distributed so SVG files were developed.

Scalable Vector Graphics

Scalable Vector Graphics (SVG) is a format for encoding vector graphics in XML (eXtensible Markup Language), designed to provide vector graphic content in Web pages.

We will try to ignore most of the XML content as this is outwith this course! This section of work will only introduce you to the most basic aspects of creating simple drawings using the SVG framework. Remember that there is much more to SVG than we have time to cover here.

As with XML, HTML, WRL and X3D files, SVG files are plain text files. We will be using a text editor (Notepad is recommended - be careful if using a more advanced text editor, like MS Word, to save the file as a plain text file) and saving the files with the suffix '.svg'. These SVG files can then be viewed in any modern browser (tested in Internet Explorer 6).

There are plenty of WYSIWYG drawing programs that can export the drawings as SVG files, but few currently support all the functionality of the SVG language that can be accessed by creating the files 'by hand'.

Viewing SVG Files

You must have an SVG capable Browser, a plug-in for your browser or a stand-alone SVG viewer. The web component for this topic was tested with the Adobe SVG viewer plug-in for Internet Explorer 6.

Right-clicking on a SVG image gives options, for example, to zoom into the image.

SVG tutorial 1: Testing for compatibility

You will learn how to add these elements (shapes) to a drawing:

- *Rectangle (including a Square)*
- *Rounded Rectangle*

- *Circle*
- *Ellipse*
- *Text*

and how to alter these attributes of each shape:

- *Size*
- *Position*
- *Rotation*
- *Line thickness*
- *Line Colour*
- *Fill Colour*

SVG is CaSe SeNsItIvE, so be careful when copying the SVG code...

Create this basic SVG file in Notepad (or other text editor):

```
<svg>

<circle cx="200" cy="200" r="150" fill="red"/>

</svg>
```

Save the file as a plain text file called 'svgtest.svg'

Test the file by opening it in your web browser (drag it into your browser, click the file or do 'File-Open' from your browser).

If you cannot see a red circle then try entering this instead (after checking carefully for spelling errors - e.g. cx is NOT the same as cX!):

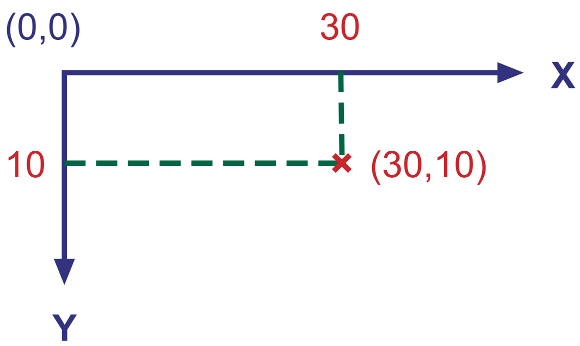
```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 20010904//EN"
"http://www.w3.org/TR/2001/REC-SVG-20010904/DTD/svg10.dtd">
<svg xmlns="http://www.w3.org/2000/svg"
xmlns:xlink="http://www.w3.org/1999/xlink">

<circle cx="200" cy="200" r="150" fill="red"/>

</svg>
```

If this then works, you will have to enter all SVG files this way (you can copy & paste) - placing the desired SVG commands in the lines before the tag (that is, replacing the indented, circle, line in this example).

Note: in general errors are not reported, if an SVG attribute is spelled wrongly it will just be set to the default value (usually 0) or nothing will be displayed at all.

Co-ordinates

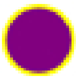
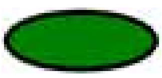
The top left of the window is the origin. Along is X and down is Y.



SVG tutorial 2: Basic shapes

This will introduce you to: Placing, setting the size of, setting the line thickness and setting the line colour of: Rectangles, Rounded Rectangles, Circles, Ellipses, Lines and Text

Create each of these files and experiment by altering the attributes for each one in some way:

	<rect x="5" y="5" width="40" height="30" fill="lightblue" stroke="black" stroke-width="2"/> *x and y is the upper left co-ordinate for the rectangle.*
	<rect x="10" y="10" width="70" height="40" rx="10" ry="10" fill="orange" stroke-width="6" stroke="purple"/> *rx and ry determine how rounded the corners are.*
	<circle cx="30" cy="30" r="25" fill="purple" stroke="yellow" stroke-width="3"/> *(cx, cy) is the co-ordinate of the centre of the circle, r is its radius.*
	<ellipse cx="60" cy="35" rx="50" ry="20" fill="green" stroke-width="4" stroke="black"/> *rx is the 'radius' of the ellipse in the x direction, ry is the same for y axis.*
	<line x1="100" y1="10" x2="10" y2="50" stroke="grey" stroke-width="10"/> *(x1,y1) is the start of the line, (x2,y2) is the end.*
	<text x="5" y="25" stroke-width="1" stroke="red" font-weight="bold" fill="black" font-size="20pt">SVG</text> *(x,y) is the start of the text.*

	<rect x="10" y="20" width="100" height="40" rx="10" ry="10" fill="rgb(255,200,200)" stroke-width="6" stroke="rgb(0,200,210)"/> *You can create the exact colours you want using RGB codes as shown.*
	<rect x="0" y="15" width="100" height="5" fill="red"/> <text x="5" y="25" font-weight="bold" fill="black" font-size="20pt">Layers</text> *Shapes placed first in the SVG file will appear beneath later shapes.*
	<text x="5" y="25" font-weight="bold" fill="black" font-size="20pt">Layers</text> <rect x="0" y="15" width="100" height="5" fill="red"/> *Shapes placed last in the SVG file will cover earlier shapes.*

SVG tutorial 3: Rotation

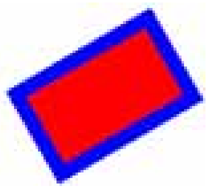
Setting the 'transform' attribute can be used to rotate an element (shape), for example:

transform="rotate(angle cx cy)"

Where 'angle' is the angle of rotation (in degrees) and (cx,cy) are the co-ordinates of the centre of rotation.

Negative angles rotate anti-clockwise; positive angles rotate clockwise.

Create these shapes:

	<rect x="20" y="40" width="100" height="60" transform="rotate(-30 60 50)" fill="red" stroke="blue" stroke-width="10" /> *This rectangle is rotated 30176; anti-clockwise around its centre at (60,50)*
	<text x="5" y="25" fill="darkblue" font-size="20pt" transform="rotate(45 5 25)">SVG </text> *This text is rotated clockwise by 45176; around the start of the text at (5,25)*

Edit each of the SVG files from tut1 and apply a rotation - think carefully about the centre of rotation and the angle, otherwise you may just find that the shape disappears from view!

SVG tutorial 4: Putting it all together

Design a simple logo, consisting of **at least 3** different shapes and text, making sure at least 1 element needs rotated.

Create the logo in a SVG file, saving it as 'tut3.svg'

Draw your name, using nothing except the basic shapes covered here - without using Text.

There is much more to learn about SVG: shapes can be grouped, shapes can be re-used, shapes and attributes can be animated, shapes can be partially transparent, effects like blur and others can be applied, hyperlinks can be created...and more.

If you have time, look up a tutorial on the www, a quick search for 'svg tutorial' should turn up lots of choices.

5.4.2.2 3D images: VRML/WRL

Virtual Reality Modelling Language / World Representation Language VRML is a standard for describing interactive 3D objects and worlds. VRML is capable of representing static and animated dynamic 3D and multimedia objects with hyperlinks to other media such as text, sounds, movies, and images.

VRML is a standard for structuring text files. These VRML text files are called WORLD files with the file suffix '.wrl'.

5.4.2.2.1 Why are we not studying X3D?

X3D (extensible 3D) is an ISO open standard for 3D on interactive web and broadcast media. It can integrate multimedia in many forms, including MPEG and SVG (being XML based also). X3D is the next step up the evolutionary ladder of interactive 3D technologies, extending on the capabilities of VRML.

X3D is essentially VRML placed into an XML (Extensible Markup Language) framework and as such requires an understanding of XML. This is an entire course in its own right and something beyond the scope of this course.

Whereas knowledge of VRML will prepare you for the step into X3D as the techniques are all very similar.

5.4.2.2.2 Creating / authoring VRML

There are a wide variety of authoring tools available, from professional, expensive 3D modellers to freeware utilities.

As WRL files are text based, any text editor can be used - but make sure that the file is saved as a plain text file with the suffix '.wrl'. Notepad in Windows is recommended, although there are freely downloadable text editors written especially for VRML which include syntax colouring. MS Word can be used but be careful to save the file as a 'Plain Text' file.

5.4.2.2.3 Viewing VRML

You need a VRML plug-in for your web browser, a stand-alone VRML viewer or a special VRML web browser.

Here is a useful link for adding VRML capabilities to your web browser: (VRML plug-in and browser detector)

http://cic.nist.gov/vrml/vbdetect.html

5.4.2.2.4 Navigating in VRML

Each VRML 'player' has slightly different features but there are many similarities.

At least three navigation modes will be available:

- **Walk** (you can move around but not leave the ground, being affected by gravity)
- **Fly** (you can move around, including up and down, unaffected by gravity)
- **Examine** (or study) (you can rotate the whole VRML world to view it from different angles)

Navigation in these modes is by clicking on the screen with the pointer. The place you click determines in which direction the movement or rotation will occur and the distance you click from the centre of the screen will determine the speed.

Individual players may have additional methods of navigation and control - see the help files. Right-clicking on a VRML world usually brings up a list of options (including access to the help file).

Hyperlinks are often embedded into VRML files and are usually indicated by the cursor shown in Figure 5.1.

Figure 5.1: VRML hyperlink cursor

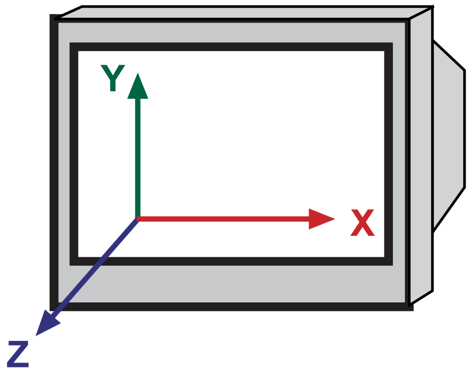
5.4.2.2.5 Thinking in 3D

A 3 co-ordinate system is used: X, Y, Z.

The Z axis comes out of the screen towards you. The origin (0,0,0) is at the bottom-left of your screen. X and Y axes go along and up the screen as shown here.

The viewpoint can always be moved, so even if a shape is out of the screen behind you, the viewpoint can always be moved backwards or rotated around in order to see the shape.

The units are usually regarded as being metres.



5.4.2.2.6 VRML Tutorials

VRML Tutorial 1 : my first VRML file

This tutorial will introduce you to the basics of creating simple VRML files.

You will learn how to:

- *Use Primitive Shapes*

  Box, Cone, Cylinder and Sphere shapes can all be created without having the geometry of each shape written out, hence they are known as primitive shapes and are often the building blocks for more complex shapes.

- *Apply Textures*

  Textures are bitmap images that are wrapped around 3D shapes.

- *Set the Size, Rotation and Position of objects*

- *Create Text*

Anything more advanced is beyond the scope of this course.

VRML is CaSe SeNsItIvE, so be careful when copying the VRML code.

You will be saving several files, so you should make your own 'VRML' folder to store them in.

Start your text editor and type in the following:

```
#VRML V2.0 utf8

NavigationInfo { type "EXAMINE" }
```

```
shape {
geometry Box {
}
}
```

Save this as 'tut1.wrl'

Open this file in your web browser (or other VRML viewing method).

You should see a white box (cube) that you can rotate by clicking on the screen.

How this works

Line 1 - Tells the VRML application the version used.

Line 2 - Sets the navigation mode to 'EXAMINE' only (no walk or fly for example). Remove this line to allow user choice or set it to "FLY" or "WALK".

Line 3 - Creates a shape (the name for objects in VRML).

Line 4 - Makes the geometry of the shape a 'box'. As its size and location are not given, **defaults** are used, i.e. 1 unit square, **centred on the origin** (0, 0, 0), with **no material** settings (that is why it is white and we cannot see the edges of the box between each face).

Repeat this three time, replacing the word 'Box' with: 'Cone', 'Cylinder' and 'Sphere'. Saving each as tut1a.wrl, tut1b.wrl and tut1c.wrl

Note the use of curly brackets {} - the contents of each node are inside a pair of these.

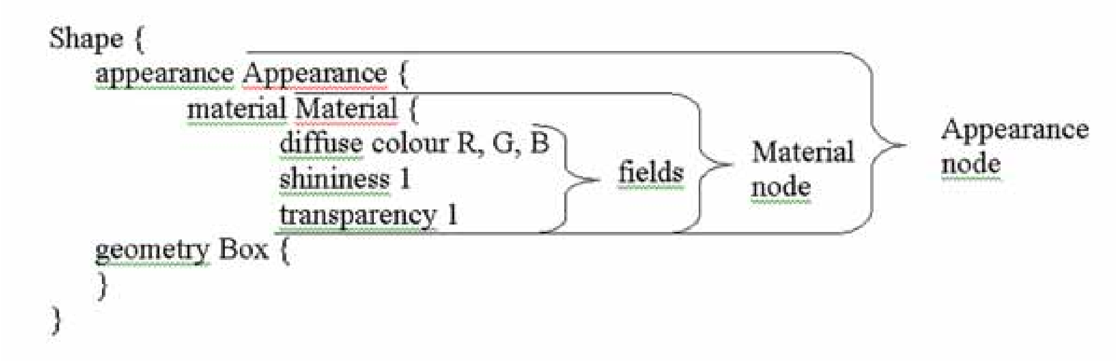
Note the formatting - formatting makes the code readable. The contents of each node are indented.

Comments can be added by placing a '#' sign at the start of the line. (The first line only is an exception - it is not ignored or regarded as a comment!)

VRML Tutorial 2 : adding colour

Everything about how the surface of the shape appears is set in the 'appearance' node.

To give the shape a colour, make it shiny or make it transparent, we use a material node (inside the appearance node).



Start your text editor and type in the following:

```
#VRML V2.0 utf8

NavigationInfo { type "EXAMINE" }

shape {
appearance Appearance {
material Material {
diffuseColor 0.8 0.8 0.8
shininess 0.4
transparency 0.0
}
}
geometry Box {
}
}
```

Save this as 'tut2.wrl' and open it in your browser.

You should see a grey box that can be rotated on the screen.

How this works

diffuseColor - sets the colour of the shape using RGB values between 0 and 1 (e.g. 0 0 0 is black, 1 0 0 is red, 1 1 1 is white and 1 0.5 0 is orange)

shininess - is how much light is reflected off the shape

transparency - is how much we can see other shapes through this shape.

Try creating these shapes, (saving each as tut2a.wrl, tut2b.wrl... etc):

- A shiny blue cylinder
- A dull purple sphere
- A nearly transparent yellow cone

Note: if you do not want to set a field, e.g. transparency, just leave it out for the default value.

VRML Tutorial 3 : controlling the size of shapes

Each type of shape has a different way of setting its size (after all a cylinder has a radius, but a box does not!)

*The methods of setting the size of each type of shape is done by setting values of **fields** within the Box, Cylinder, Sphere or Cone **nodes**. (Note: technically speaking 'geometry' is not a node, simply a holder for the Box, or other, node).*

Box

```
geometry Box {
size X Y Z
]
```

where X Y and Z are the sizes of the box in each direction.

example:

```
geometry Box {
size 0.5 1 2
]
```

would create a box half a metre wide, 1m tall and 2 metres long.

Note: Large, flat boxes are often used where a smooth flat floor or wall is needed in a scene.

Cone

```
geometry Cone {
bottomRadius 0.25
height 0.6
]
```

would create a cone with a radius at its foot of 25cm and an overall height of 60cm. (Tip: a negative height creates an upside-down cone)

Cylinder

```
geometry Cylinder {
height 3
radius 1
]
```

would create a cylinder 3m tall with a radius of 1m.

Sphere

```
geometry Sphere {
Radius 4
]
```

would create a sphere with a radius of 4m. (i.e. 8m across)

Create scenes the following shapes (saving each as tut3a.wrl, tut3b.wrl... etc)

- A green floor, 10m long, 6m wide and 5cm thick
- A pale yellow wall, 2cm wide, 3m tall and 5m long
- A brown cone, bottom diameter 10cm, height 20cm
- A light grey cylinder, 4m tall with a radius of 10cm
- A pink sphere measuring 7cm across

VRML Tutorial 4 : transformations

Transformations are used to move shapes (or place shapes where you wish, called a translation), rotate Shapes (or set the orientation of shapes).

```
Transform {
        translation 1 2 3
        rotation 0 1 0 3.14
        children [
                Shape {
                      ↓
                }
        ]
}
```

Shape node is inside the children [ ] group of the Transform node.

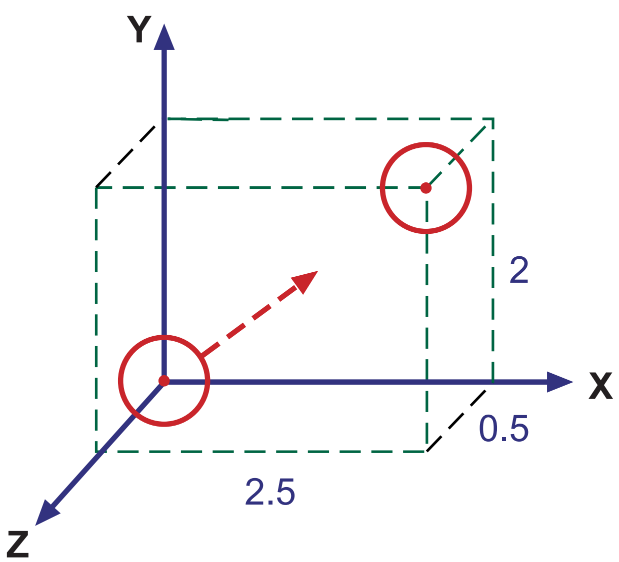
The transformation will affect all shapes defined inside the children group.

Negative values can be used for all arguments in translation or rotation commands.

Translating

The translation command has three arguments: translation X Y Z , where X Y and Z determine the distance that the object has to be moved in each direction.

For Example, applying a translation of 2.5 2 0.5 to a Sphere would produce:

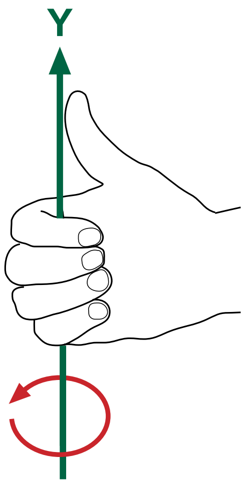


Rotating

Angles in VRML are measured in Radians, not degrees. π (pi) radians is 180176;.

To convert degrees to radians, divide by 180 and multiply by π.

Rotation is using the right hand rule. With your right thumb pointing in the positive direction of the axis, the curl of your fingers show the direction of a positive rotation about that axis.

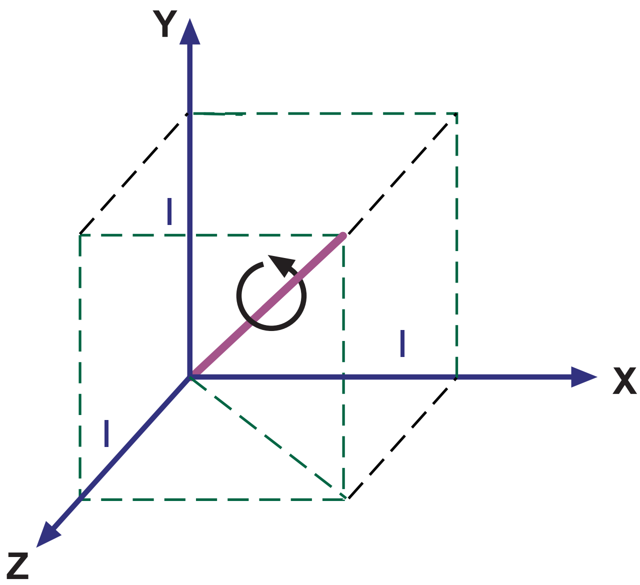


The rotation field in a Transform node has 4 arguments: rotation X Y Z angle

X Y Z determine the axis of rotation. For example, 0 1 0 is the same as 0 23 0 (they are both the same direction, that is, around the Y axis).

The axis of rotation can be any direction.

For example: rotation 1 1 1 angle:

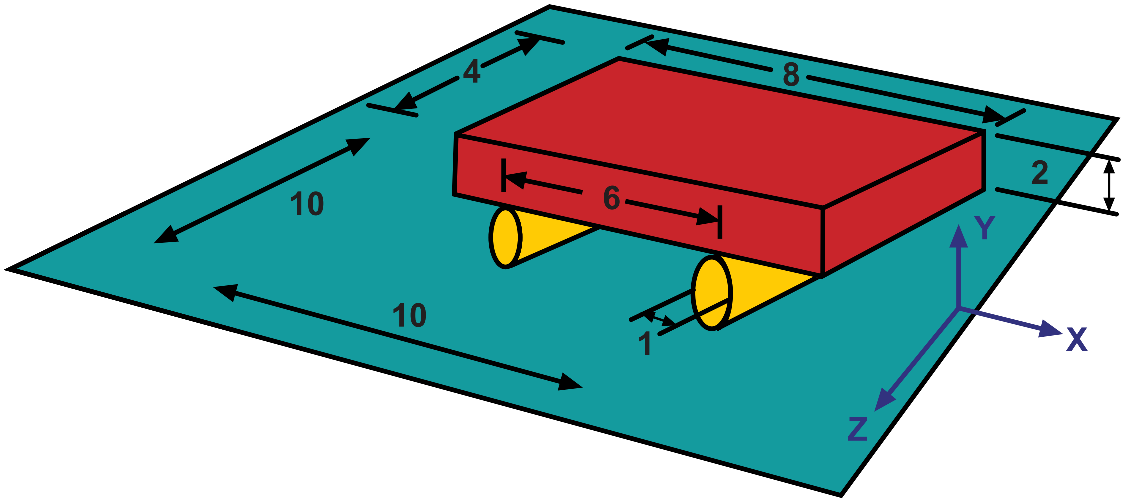


This command would rotate the shape(s) in the children group of the transform node around the X axis by 45 degrees.

```
rotation 1 0 0 0.79
```

VRML Tutorial 5 : Multiple shapes - putting it all together

Many shapes can be added to a scene by typing in other Shape nodes within the file. For example:



Type in or download this example file.

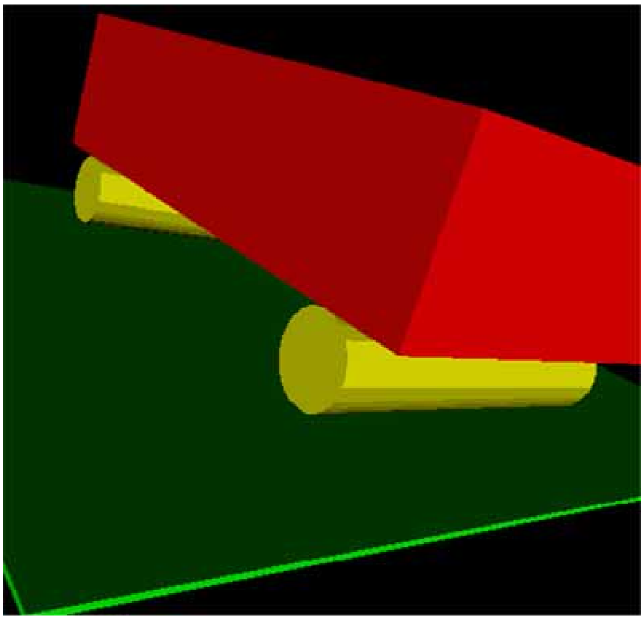
```
#VRML V2.0 utf8

NavigationInfo { type "EXAMINE" }

####################################
#The flat green floor size 10 by 10#
####################################
Shape {
appearance Appearance {
material Material {
diffuseColor 0 1 0
}
}
geometry Box {
size 10 0.05 10
}
}
##########################
#The two yellow cylinders#
##########################
Transform {
translation -3 0.5 0
rotation 1 0 0 1.57
children [
Shape {
appearance Appearance {
material Material {
diffuseColor 1 1 0
}
```

```
}
geometry Cylinder {
radius 0.5
height 4
}
}
]
}
Transform {
translation 3 0.5 0
rotation 1 0 0 1.57
children [
Shape {
appearance Appearance {
material Material {
diffuseColor 1 1 0
}
}
geometry Cylinder {
radius 0.5
height 4
}
}
]
}
#############
#The red box#
#############
Transform {
transformation 0 2 0
children [
Shape {
appearance Appearance {
material Material {
diffuseColor 1 0 0
}
}
geometry Box {
size 8 2 4
}
}
]
}
```

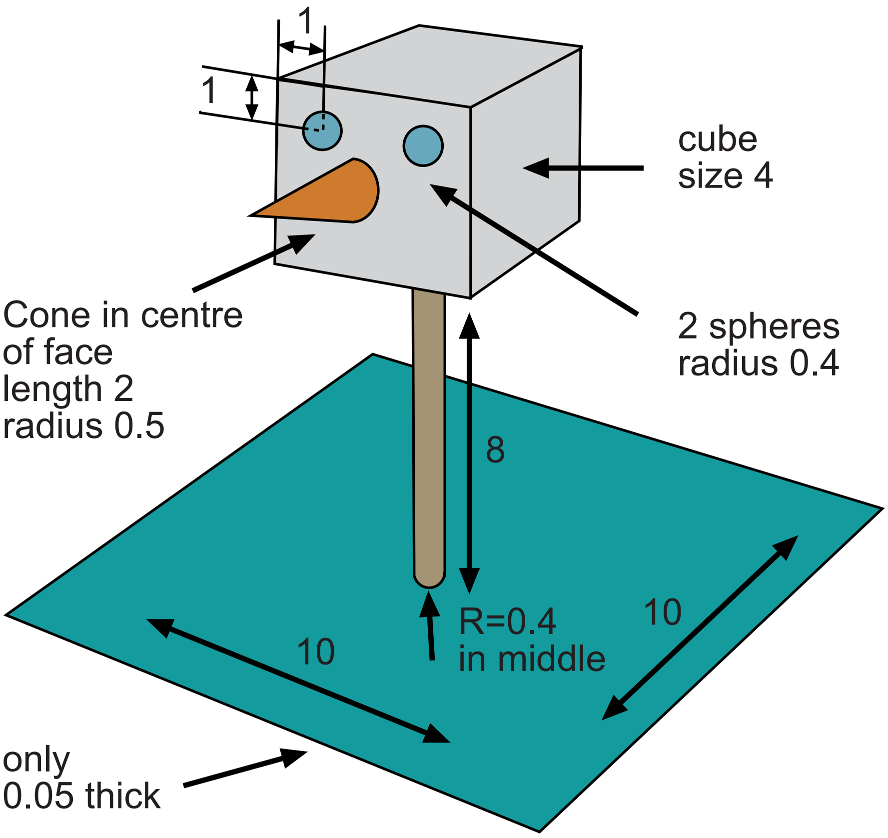
Note: as the file is larger than previous files comments are now being used to structure it.

Save it as 'tut4example.wrl'

Open it in your browser to check it works as expected:



Create this scene, saving it as 'tut5.wrl':



Extension Work

Design - If you have time, make your own design as shown above and try creating it.

Hyperlinks

It is possible to make any shape a hyperlink to any other file (e.g. another VRML file or web page) This is done by creating an 'Anchor' node around the node we wish to make an anchor. The Transform node (if present) is placed inside the Anchor node:

```
Anchor {
        children [
                Transform {
                        Shape {
                                ⁞
                        }
                }
        ]
        description "some text describing where the link takes you"
        url "filename to link to"
}
```

Try creating links between each of your VRML files.

VRML Tutorial 6 : Adding textures

All the scenes created so far look very computer generated. Only bitmap images can give a realistic look. So if we wrap a bitmap image around a 3D object we can make it appear more realistic. Doing this is called texturing the shape.

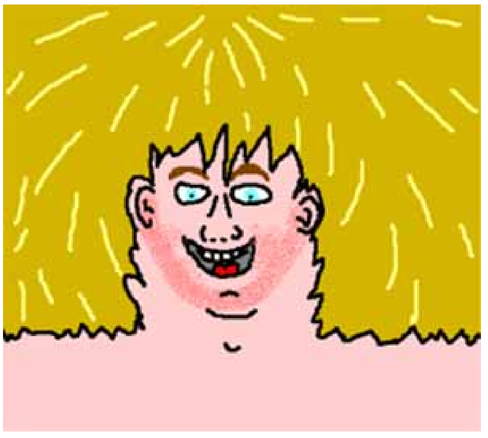
A texture is defined by creating a 'texture' node inside the 'Appearance' node for the shape instead of using a 'material' node. (You could include both material and texture nodes; having a colour underneath the texture often creates a better effect, and shows through when using textures with transparency).

For example, creating a head:

Without texturing

```
#VRML V2.0 utf8

NavigationInfo { type "EXAMINE" }

Shape {
appearance Appearance {
material Material {
diffuseColor 1 0.8 0.8
}
}
geometry Sphere {
radius 1
}
}
```

Image used for texturing



When greater realism is wanted, for example in computer games, photographs can be used as textures. To create a realistic texture of a head, the photograph would have to be edited first, or several photographs joined together.

image saved as 'FACE.JPG' in same folder as the 'wrl' file

Shape with texturing

```
#VRML V2.0 utf8

NavigationInfo { type "EXAMINE" }

Shape {
appearance Appearance {
texture ImageTexture {
url "FACE.JPG"
}
material Material {
diffuseColor 1 0.8 0.8
}
}
geometry Sphere {
radius 1
}
}
```

The file size of bitmap images can be large, especially when compared to the vector data, so using textures can significantly increase the size of the WORLD data - so be careful when choosing image formats and settings for use as textures.

Draw a face of your own design (Bitmap, JPEG, GIF and PNG files all work) and use it to texture a sphere as shown above. Save the file 'tut6.wrl'

VRML Tutorial 7 (optional) - Text

Text is treated just like all the other shapes we have seen. It is created with the handle 'geometry' as before.

Example 1

```
geometry Text {
string "Hello"
length 2
}
```

Creates the text 'Hello' starting at the origin, fitting the text into a length of 2m.

Example 2

```
geometry Text {
string ["Welcome to", "VRML"]
length [3, 3]
}
```

Creates the text 'Welcome to' and 'VRML' over two lines, fitting each line to a width of 3m.

The style of the font is determined by using adding a FontStyle node to the Text node. Here are three of the fields that you can set in FontStyle:

```
FontStyle {
style (choose from BOLD ITALIC or BOLDITALIC)
justify (choose from BEGIN MIDDLE or END)
size (height of the letters, half this number to get the approximate height)
}
```

Example 3

```
geometry Text{
string "Multimedia"
fontStyle FontStyle {
size 2
style "ITALIC"
justify "MIDDLE"
}
}
```

The size field gives the height of the letters. Half this number to get the approximate height in drawing units (meters).

The text here will be about 1m high (bit larger).

This generates the word "Multimedia" in italic writing, centred on the origin.

Edit each wrl file you have created and add in some short, descriptive text. You can place the 'Shape { geometry Text { }}' node inside a 'transform' node in order to place the text at some point other that the origin in your scene.

5.4.3 Synthesised Sound Data (MIDI)

MIDI is a standard that was created to allow musical instruments to be able to communicate with each other, for example to synchronise a drum machine with a keyboard. Using this same interface standard it is possible to use a PC to control musical instruments or to communicate with them.

MIDI has also become a standard file type for musical files that contain data that enable musical instruments to recreate music. MIDI files do not contain any recordings of sound (there is no digitised sound), they just contain the data that is required by a musical instrument to synthesise the required sounds. It is basically a file containing all the commands that could be sent to another instrument over a MIDI interface.

MIDI files store the data in a binary format but free software is available that can convert it to text, where you can edit it using any text editor and then convert it back. We can compare a MIDI file to a piece of sheet music; Just as a musician reads the notes to play off a piece of music, so a computer 'reads' the notes to play from the MIDI file. The musician may use the piano to make the music. The computer uses the soundcard to make the music, and just as different pianos will sound different, so will different soundcards. A musician generally only plays one instrument at a time, but a computer soundcard is expected to play up to 16 different instruments at a time.

5.4.3.1 Hardware / Software

MIDI music files can be created using MIDI software, by using the mouse and the normal computer WIMP interface.

The layout shown in Figure 5.2 is typical of MIDI editing software. MIDI allows up to 16 track to be used, in this example 7 tracks are being used. This means that, although the instrument for any track can be changed during the song, that at any one time there will never be more than 7 instruments playing.

Figure 5.2: Interface of typical MIDI editing software

You can see the instrument names assigned to each track on the screenshot. There are two main ways that music can be created or edited on screen:

1. Using the Score View (Figure 5.3) - The score view lets you edit the notes for each track as they appear on sheet music (called the score).

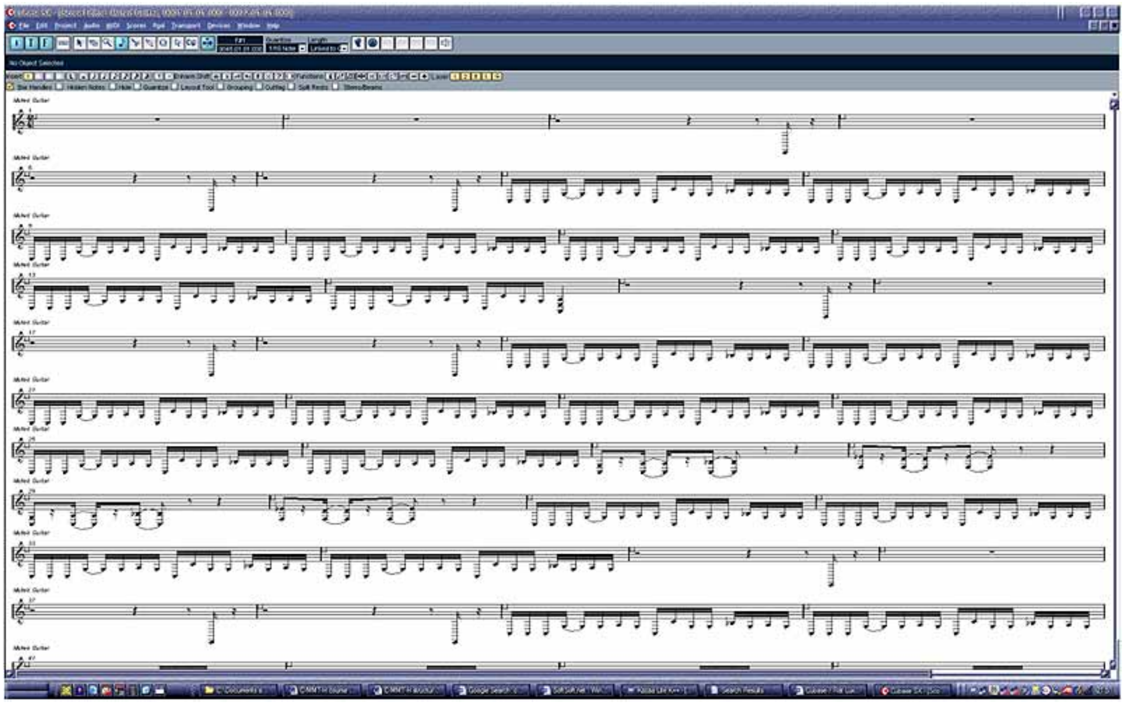


Figure 5.3: Score view

2. Using the Piano Roll (Figure 5.4) - This view shows how the notes for each track relate to the notes as they are found on a piano. This view allows the notes to be 'drawn' in using a pencil tool.

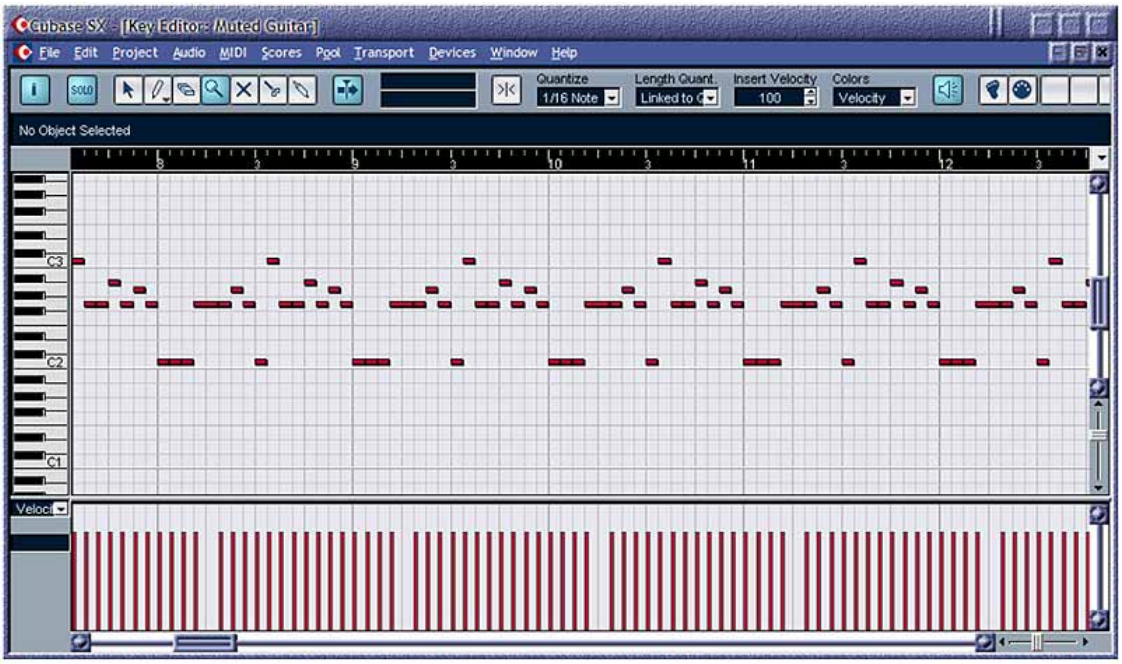


Figure 5.4: Piano roll

This particular MIDI software allows you to view and edit a list of all the events that happen for a particular channel (Figure 5.5).

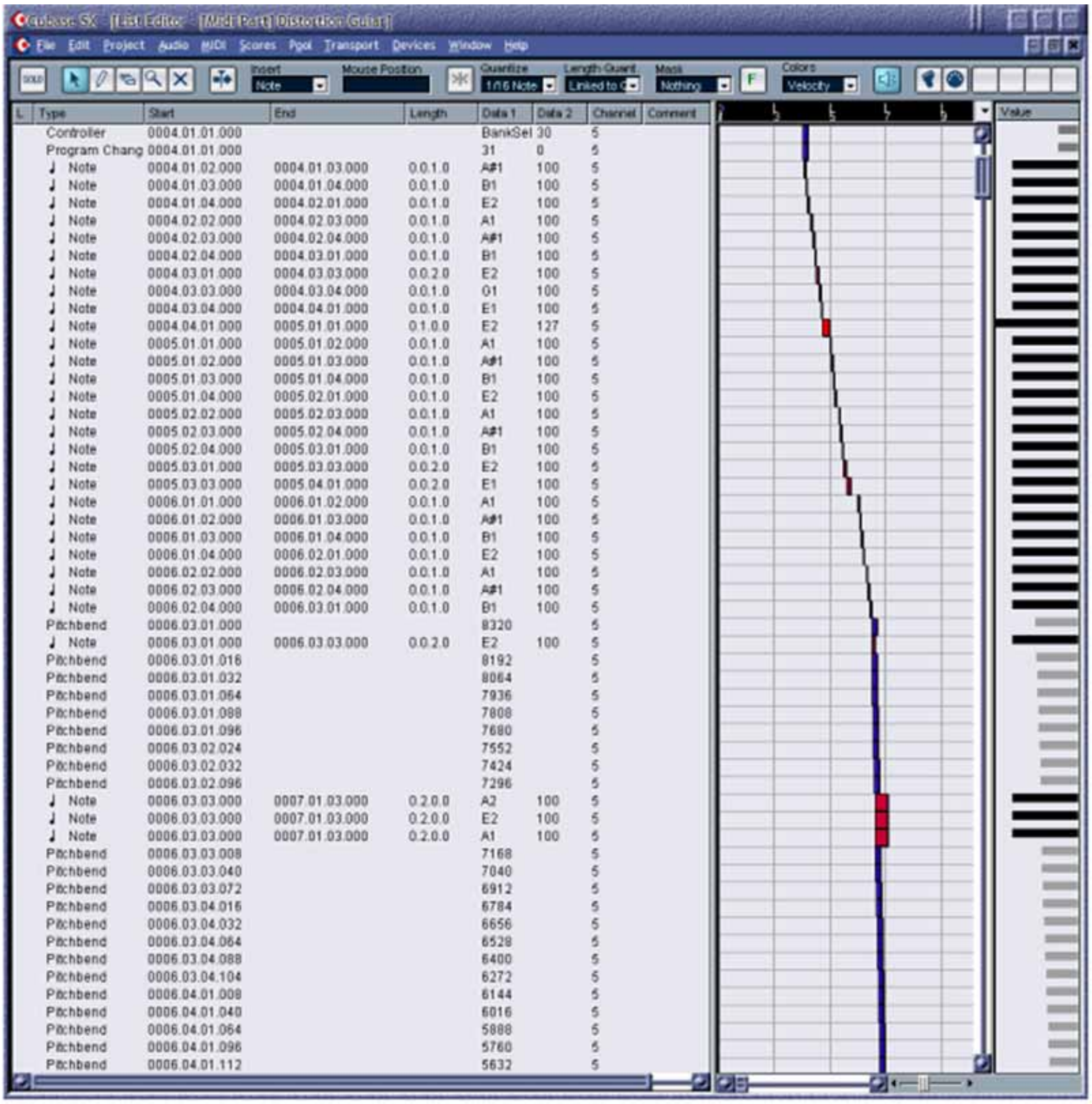


Figure 5.5

Having all this technology does not turn you into a musician. In order to create or edit MIDI tracks, you must have some musical abilities and knowledge.

As you can imagine, entering an entire song by individually click each note for each instrument, adjusting its volume and timing and placing it on the score is very laborious and produces very 'artificial' music without any spontaneity.

The usual way to create MIDI music data is to enter the notes using a MIDI keyboard. A MIDI keyboard often looks just like a standard synthesiser keyboard, except these often cannot generate any sounds of their own. The musician plays the notes while the computer software records the notes played, duration of each note and the volume of each note (by sensing the pressure on the key of the keyboard - a feature called after-touch).

The musical notes recorded can then be viewed or edited in MIDI editing software like the one shown here.

5.4.3.2 MIDI file format

A MIDI file, although stored in binary format, could easily be a text file. There is readily-available software to convert MIDI files into text files, where the text MIDI file can then be edited in a favourite word processor, before being converted back to a MIDI file.

A MIDI file is simply a file containing a header and a number of MIDI messages (often called MIDI events).

A MIDI file consists of Channels. There are up to 16 channels. By examining the MIDI files here in text form, you should be able to get an idea for the type of data that is contained within a MIDI file.

MIDI files contain up to 16 channels, each playing one instrument at a time, although the instrument assigned to a channel can be changed any number of times during the song by a control event.

The MIDI file header contains some important information, for example the tempo (i.e. speed to play it at in Beats Per Minute - BPM). Each channel consists of a sequence of messages, where each message may contain the following information:

- start of a note
- channel to use
- pitch of the note
- volume to play it at
- end of a note

Control messages can also be sent and may contain the following information:

- channel to apply the control message
- controller to change (e.g. use effects like echo or chorus or change instrument)
- value of the controller (e.g. the amount of the effect to use or instrument to change to)

5.4.3.3 Advantages and Disadvantages of MIDI

MIDI music never sounds as realistic as recorded (digitised) sound because the soundcard can never be as accurate as the real instrument at making the sounds, although a very high quality (and expensive) soundcard may come close when synthesising particular instruments.

Advantages of MIDI:

- Smaller file size
- All aspects of the music can be edited
- Effects can be applied to individual instruments

- There is no interference or background noise from the recording

Disadvantages of MIDI:

- Dependent on soundcard for quality of sound
- Does not contain vocals
- Fewer effects can be applied to the sound

5.4.3.4 A sample MIDI file

This sample MIDI file has been converted to text format.

Here are the contents of this MIDI file:

```
MThd | Format=1 | # of Tracks=4 | Division=480

Track #0 ******************************************
    Time       Event
   1: 1:  0 |Track Name  | len=12    |
              0x52 0x4F 0x43 0x4B 0x32 0x2E 0x4D 0x49 0x44 0x20 0x23
              <Example.MID #> 0x33 <3>
               |SMPTE  | hour=96 | min=0 | sec=0 | frame=0 | subs=0
               |Time Sig  |  4/4      | MIDI-clocks\click=24 | 32nds\quarter=8
   2: 1:  0 |Tempo   | BPM=115   | micros\quarter=521739
   3: 1:  0 |End of track|
Track #1 ******************************************
    Time       Event
   1: 1:  0 |Instrument  | len=4     |
      0x58 0x35 0x44 0x52        <X5DR>
      |Track Name  | len=5     |
      0x44 0x72 0x75 0x6D 0x73       <Drums>
       |Controller  | chan=10    | contr=Effects      | value= 79
       |Controller  | chan=10    | contr=Volume H | value=100
       |Program  | chan=10      | pgm #=  1 Grand Piano
       |Controller  | chan=10     | contr=Chorus     | value=  0
       |Pitch Wheel | chan=10   | bend=0
       |Controller  | chan=10      | contr=Hold Ped | value= 0
   2: 1:  0         |On Note  | chan=10   | pitch=F#1  | vol=70
        |On Note  | chan=10   | pitch=C 1   | vol=120
   34 |Off Note  | chan=10   | pitch=f#1   | vol=70
   65 |Off Note  | chan=10   | pitch=c 1   | vol=64
 110 |On Note  | chan=10   | pitch=C 1   | vol=94
 115 |On Note | chan=10    | pitch=C#1  | vol=54
        |On Note   | chan=10   | pitch=F#1  | vol=45
 148 |Off Note  | chan=10   | pitch=f#1   | vol=45
 175 |Off Note  | chan=10   | pitch=c#1  | vol=54
 221 |Off Note  | chan=10   | pitch=c 1   | vol=64
 238 |On Note  | chan=10   | pitch=F#1  | vol=66
```

```
 271 |Off Note  | chan=10   | pitch=f#1   | vol=66
 353 |On Note  | chan=10   | pitch=F#1  | vol=50
 389 |Off Note  | chan=10   | pitch=f#1   | vol=50
 478 |On Note  | chan=10   | pitch=F#1  | vol=66
        |On Note  | chan=10   | pitch=C#1  | vol=127
   3: 1:  0 |End of track|


Track #2 ******************************************
    Time       Event
   1: 1:  0 |Instrument  | len=4     |
            0x58 0x35 0x44 0x52        <X5DR>
            |Track Name  | len=4     |
            0x42 0x61 0x73 0x73        <Bass>
            |Program  | chan= 2   | pgm #= 39 Synth Bass 1
            |Controller  | chan= 2   | contr=Mod H | value=  0
            |Controller  | chan= 2   | contr=Effects | value= 29
            |Controller  | chan= 2   | contr=Chorus | value=  0
            |Controller  | chan= 2   | contr=Volume H | value=127
            |Controller  | chan= 2   | contr=Pan H | value= 64
            |Poly Press  | chan= 2   | press=  0
            |Controller  | chan= 2   | contr=Hold Ped | value= 0
              |Pitch Wheel | chan= 2   | bend=0
   2: 1:  0 |On Note  | chan= 2   | pitch=G#0  | vol=120
          89 |Off Note  | chan= 2   | pitch=g#0  | vol=64
         121 |On Note  | chan= 2   | pitch=G#0  | vol=110
         211 |Off Note  | chan= 2   | pitch=g#0  | vol=64
         232 |On Note  | chan= 2   | pitch=G#0  | vol=91
         319 |Off Note  | chan= 2   | pitch=g#0  | vol=64
         351 |On Note  | chan= 2   | pitch=G#0  | vol=110
         449 |Off Note  | chan= 2   | pitch=g#0  | vol=64
         467 |On Note  | chan= 2   | pitch=G#0  | vol=104
   3: 1:  0 |End of track|
Track #3 ******************************************
    Time       Event
   1: 1:  0 |Instrument  | len=4     |
              0x58 0x35 0x44 0x52        <X5DR>
              |Track Name  | len=10    |
     0x4D 0x75 0x74 0x65 0x64 0x20 0x47 0x75 0x69 0x74    <Muted Guit>
             |Program   | chan= 4   | pgm #= 29 Muted Guitar
             |Controller  | chan= 4   | contr=Effects | value=115
             |Controller  | chan= 4   | contr=Chorus | value= 81
             |Controller  | chan= 4   | contr=Volume H | value=105
             |Controller  | chan= 4   | contr=Pan H | value= 74
             |Controller  | chan= 4   | contr=Mod H | value=  0
             |Controller  | chan= 4   | contr=Hold Ped | value= 0
             |Pitch Wheel | chan= 4   | bend=0
              |Poly Press  | chan= 4   | press=  0
   2: 1:  0 |On Note  | chan= 4     | pitch=B 3  | vol=99
```

```
        130 |Off Note  | chan= 4      | pitch=b 3  | vol=64
              |On Note  | chan= 4      | pitch=A#3  | vol=75
        249 |On Note  | chan= 4      | pitch=G#3  | vol=99
        261 |Off Note  | chan= 4      | pitch=a#3  | vol=64
        349 |Off Note  | chan= 4      | pitch=g#3  | vol=64
        365 |On Note  | chan= 4      | pitch=C#4  | vol=75
   3: 1:  0 |End of track|
```

5.4.4 Output hardware

Professional soundcards as used by musicians and the music industry are very expensive and even then not all the instruments sound as realistic as the real thing, but effects can be applied and synthesisers are respected instruments in their own rights.

A lot of effort has gone into creating realistic piano and percussion sounds, but others, like guitars, still sound synthetic.

The sounds are generated by using short recordings of the real instruments (samples). These samples are stored in memory of the sound card (called a wave-table). For the soundcard to synthesise an instrument it calls on a sample from the wave-table and manipulates it to produce different notes.

For a realistic synthesis, several samples may be used to produce the sound for a single instrument, for example, a piano sound may be synthesised using samples taken from the lower notes, middle notes and upper pitches, as they all have different sound characteristics.

Soundcards can also apply effects - in real-time - to sound being produced. The effects selected by the MIDI events are applied by the sound processor in the sound card - effects such as echo and reverb.

5.5 Practical task

Using knowledge you have learned from this section on object oriented data, prepare at least one object oriented file (MIDI, SVG or WRL) and insert it into one of the multimedia applications you created in Topic 1.

5.6 Questions

Q1: Which of these describes an advantage of vector graphics?

a) The image can be edited at pixel level
b) Effects such as emboss, negative or edge detect can be applied
c) The images are often very realistic
d) The file sizes are usually smaller than other types of image file

Q2: Describe three other advantages of vector image files.

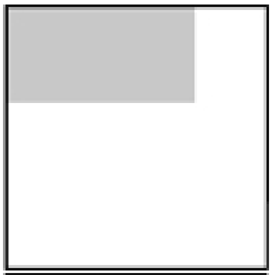
Q3: Which of these is NOT an attribute you would expect of a shape in a 2D drawing file?

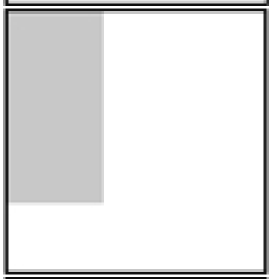
a) Layer
b) Location
c) Rotation
d) Pitch

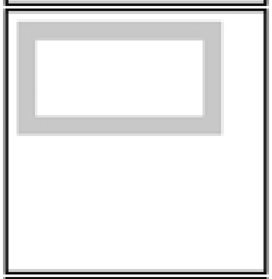
Q4: Choose one type of 2D shape and list all the attributes that you think the software uses or may use to describe its appearance and behaviour.

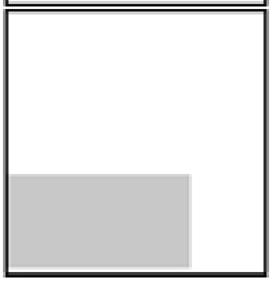
Q5: Which of these shapes would be drawn by this command?

<rect x="0" y="0" width="20" height="40" fill="rgb(200 200 200)"/>

a) 

b) 

c) 

d) 

Q6: Write down commands that will create shapes similar to the other answers.

Q7: Which of these rotations would produce this? Looking face-on from default view

a) rotation 1 0 0 0.79
b) rotation 0 0 1 0.79
c) rotation 1 0 0 45
d) rotation 0 0 1 45

Q8: Draw how the same text would appear from the same viewpoint if the other rotations in this answer were applied. Check your answers by trying them out.

Q9: Which of these is a disadvantage of storing music in MIDI format over wave format?

a) The file size is greater
b) The music cannot be played in web pages
c) The music cannot be easily edited
d) The music is not as realistic

Q10: Describe another disadvantage of storing music in MIDI format

Topic 6

Trends and contemporary technologies

Contents

Prerequisite knowledge

Before studying this topic you should have:

- *Familiarity using a web browser to navigate web sites and to copy text and images.*

All topics require:

- *familiarity in working with the computer filing system - i.e. the ability to create, view the properties of, save and open files and folders;*
- *the ability to locate and run the necessary software for each topic;*
- *the ability to switch between multiple applications on the computer system;*
- *knowledge of units of storage (bits, Bytes, KB, MB);*
- *knowledge of the basic components of computer systems (mostly just input, output and backing store devices).*

Learning Objectives

By the end of this topic you should be able to discuss the effects/implications of the increasing capabilities of contemporary:

- *Communications technologies;*
- *Storage technologies;*
- *Processing technologies;*
- *Display technologies;*
- *Examples of convergence;*

on multimedia systems.

This section, by its very nature, will be outdated very quickly. This material should be supplemented with current material from the Internet and/or computing publications.

Computing is a fast changing world. Never before have computers been so integrated into our lives. This section looks at the reasons for this, and hopefully provides a glimpse into the imminent future and perhaps stimulates thoughts of what might be possible and how we could best utilise our current technology.

6.1 Communications

Communications technology determines how quickly data can be sent between devices. Faster connections allow greater amounts of multimedia data to be transferred and wireless capabilities make this more accessible.

6.1.1 Wireless Communications

WiFi and Bluetooth are two wireless technologies that are enabling computers, not only to communicate more readily with one another, but with a host of other devices as well.

Investigating wireless communications

Search the Internet or look up a journal to find devices that are either wireless or Bluetooth enabled - these devices should not be desktop or laptop computers!

In what way(s) should the wireless capabilities of these devices be of use?

What are the fastest data transfer rates you can find for WiFi and Bluetooth?

In your answers, state what versions of these technologies are being described.

6.1.2 Buses

Buses are physical connections in and out of the computer. Bus speeds are increasing all the time as new buses are developed. This helps us transfer greater amounts of multimedia data to and from the computer.

Two of the fastest buses are currently: USB 2 and Firewire.

Investigating buses

Search the Internet or look up a journal to find devices that connect to computers using these devices. Try to find devices that are not regarded as 'standard peripherals' such as printers or keyboards.

Explain for each device why a fast connection to the computer would be advantageous.

What are the fastest data rates you can find in relation to USB and Firewire buses?

In your answers, state what versions of these technologies are being described.

6.2 Storage

Multimedia data requires a large storage capacity. Computer storage devices have come a long way from the days when a hard disc was 40cm across, cost hundreds of pounds and required both hands to lift it.

There is a desire to have maximum amount of storage in a minimum amount of space, for as little money as possible and that will work extremely fast! This is the trend.

Solid state, or silicon, technologies are prevalent and being increasingly used in many devices. Some common formats for this are Compact Flash, SmartMedia and xD cards. These are being used in many non-computing devices. The miniaturisation of silicon technology is producing larger capacity, small devices such as these. It is likely that this trend will continue for some time, but there are limits to just how much data can be physically fit onto a piece if silicon. Right now there is much research around the World, looking for alternative ways to store even greater amounts of data in small spaces, some even more esoteric than might be seen in a science fiction film.

6.2.1 Magnetic Technologies

The first backing stores were based on magnetic technology. Magnetic tapes, floppy disc drives and hard disc drives all use magnets to store and read data from the devices.

There is speculation that the limits of this technology are being reached, but currently the trend still holds with cheaper, faster, smaller and higher-capacity devices being produced all the time. Some hard disc drives are only a couple of cm in size and - at the time of writing - up to 4Gb capacity. This technology is struggling to stay ahead to silicon flash memory storage as currently the largest capacity flash memory card is also 4Gb.

6.2.2 Optical Technologies

CDROMS and their variants are currently the most common backing store for distributing data and software. DVDs are beginning to replace CDs as they offer a greater capacity. Current DVDs squeeze just about as much data as is possible onto a side of a disc using current red lasers but new optical technologies are being developed that offer even greater speeds and storage capacities. One of these uses blue lasers, rather than the red as the data can be placed much closer together. Blue laser devices went on sale to the public in 2003, offering five times the storage capacity of current DVDs.

6.2.3 Holographic Technologies

Currently only in research, maybe in the shops by the time you read this! Holographic storage is based on lasers, but the medium is a 3-dimensional array. It promises greater data densities than the largest current hard discs along with greater transfer speeds. A 1cm cube of holographic recording medium could store the equivalent of 100 DVDs and read 100 times faster than DVDs. Current research has not yet achieved this.

Investigating storage devices

Look up a catalogue (paper or on-line) and find out:

- What is the largest capacity storage device sold?
- What is the smallest sized storage device sold - what is its capacity?

Describe uses for each of these devices.

Find out about holographic storage. Have any devices been developed? What capacities are offered? Why is there a need for such devices - what applications are envisaged?

6.3 Processing

A processor is a computer! Any device with a processor is a computer and processors are becoming smaller, cheaper and faster. Although super-fast processors are being developed (even processors running on light instead of electrons now exist) it is not these super-fast processors that cause convergence, instead it is the proliferation of cheap, low power, semi-powerful, miniature processors - or computers on a chip that invade our every-day lives.

At the fast end of processing technology, our computers are now multi-processor machines with processors for individual operations: processing and compressing video, processing video to be displayed and processing sound. Sometimes these processors will be more powerful than the main system processor in the computer. This powerful processing allows more impressive multimedia to be handled.

Investigating processing

Think of, or research for, 5 devices that would not normally be classes as computers, but include a processor or computing module. What is the purpose of the processor in each case?

6.4 Display

There are some exciting and revolutionary developments, currently on the horizon, regarding display technology. Current LCD displays are becoming cheaper and more commonplace - allowing the viewing of multimedia material in more and more places. At the other end of the spectrum, high quality displays are being developed for immersive virtual reality (for example in 'cave' setups - where the image is projected all around in an enclosed room).

6.4.1 Flexible displays

These offer many possibilities, for example: roll up your computer display like a newspaper.

6.4.2 Virtual 3D

These systems simulate 3D by providing a separate image for each eye. Some monitors do this without any headwear required (termed ’autostereoscopic’), others need special glasses - to filter out one of two images to each eye, or to provide a separate image to each eye using a small display for each.

These systems are readily available, some as virtual reality helmets, others as ’shutter’ glasses provided with a graphics card and others as specialised 3D monitors - each with advantages and disadvantages.

6.4.3 Real 3D

Real 3D displays project a moving image into space (termed ’volumetric’ displays) - and are currently available, although still a technology in its infancy and not without its problems.

Investigating displays

Research each of these: flexible displays, virtual 3D displays and real 3D displays - what are the intended uses? What are their (current) limitations?

6.5 Questions

Q1: Which of these devices illustrates convergence?

- A standard Video player
- A standard Digital watch
- A standard Smart Phone
- A standard fridge

Q2: Find 4 more devices that illustrate convergence - describing the reasons why they are good examples of convergence.

Q3: A car engine is governed by a computer. During a service the car computer communicates with the service computer, which of these would provide the greatest data transfer rate?

a) USB 2.0
b) Firewire 1394b
c) WiFi 802.11b
d) Bluetooth

Q4: Find out the data transfer rates for these four technologies.

Q5: Which of these storage technologies potentially offers the greatest data densities?

a) Solid state silicon
b) Optical disc
c) Magnetic

d) Holographic

Q6: What is the single largest capacity device available using each of these technologies? (Describe the four devices and give their capacities)

Q7: Which of the following describes the trend that embedded processors are following?

a) Cheaper, more powerful, smaller in size and using less power
b) More expensive, less powerful, larger in size and using more power
c) More expensive, more powerful, larger in size and using less power
d) Cheaper, less powerful, smaller in size and using more power

Q8: Find out how the power of processors is measured.

Q9: Which of these is a current limitation of autostereoscopic virtual 3D display monitors?

a) The flicker rate is to slow, causing headaches after prolonged use.
b) The colour information in the image is lost through the conversion to a red and blue image.
c) The position of the viewer's head must stay within a narrow range to experience the 3D effect.
d) A heavy headset has to be worn.

Q10: State which types of 3D display for which the other answers are draw-backs.

Glossary

ADC

Analogue to Digital Converter: converts analogue signals to a stream of digital data

ADPCM

Adaptive Differential Pulse Code Modulation: a technique for encoding audio data

AVI

Audio Video Interleave: file format consisting of frames of audio interleaved between frames of video

CCD, Charge Coupled Device

a light detecting sensor

CMOS Complementary Metal Oxide Semiconductor

A type of transistor that can be used as a light sensor

DAC

Digital to Analogue Converter: converts from a stream of digital data to an analogue signal

DAC: Digital Audio Converter

Digital Audio Converter

DSP

Digital signal processor: a processor specialised to processing waveform signals

GIF

Graphics Interchange Format: a format for graphics files, possibly subject to patent restrictions

GPU

Graphical processing unit

JPEG

Joint Photographic Experts Group: the name used for an image compression format specified by that committee.

LZW

Lempel-Ziv-Welch, a data compression technique developed in 1977 by J Ziv and A Lempel, and later refined by Terry Welch.

MPEG

Moving Pictures Expert Group: a series of standards for compressing audio and video.

PAM

Pulse Amplitude Modulation: sampling mechanism for audio signals

PCM

Pulse Code Modulation: storage and transmission format for audio data

PNG

Portable Network Graphics: a freely available format for graphics files

Rasterisation

process of converting from an object oriented format to a discrete bitmap format

RIFF

Resource Interchange File Format: a multimedia data file format

RLE

Run length encoding: a simple compression mechanism working on repeated sequences of the same character

SVG

Scaleable Vector Graphics file format for vector graphics interchange on WWW

VRAM

Video Random Access Memory

Answers to questions and activities

1 Development process for multimedia applications

Answers from page 14.

Q1: b) A Hollywood block-buster film

Q2: It is not necessary to use a computer to view a film.

Films are not interactive.

[The BCS does state that multimedia requires the information to be presented by a computer system and here the question states that it is an *application*, implying interaction - these two requirements fit with the definition given in the notes]

Q3: c) Deciding the layout of each page of the application

Q4: A- Analysis

B- Evaluation

D- Documentation

Q5: a) It is more flexible, allowing greater control over what happens in the application

Q6: Authoring software is more difficult to learn to use as it has more complex capabilities. Debugging a multimedia application created with Authoring software can be more difficult as the application can be more complex.

Q7: d) The client software for receiving the streamed broadcast

Q8:

- RealPlayer
- Microsoft Media Player

Both of these have the ability to view and play a wide range of media files that are stored on the computer.

Media player can also be used to capture (record) sound from audio CDs.

Q9: d) New codecs are being developed that are better at compressing the data

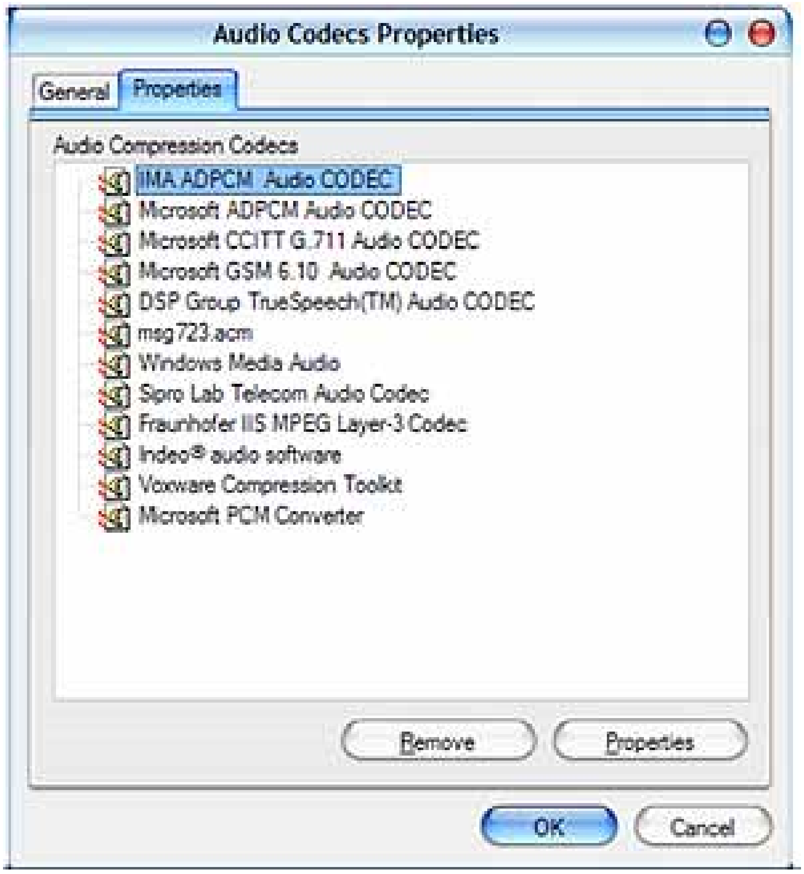
Q10:

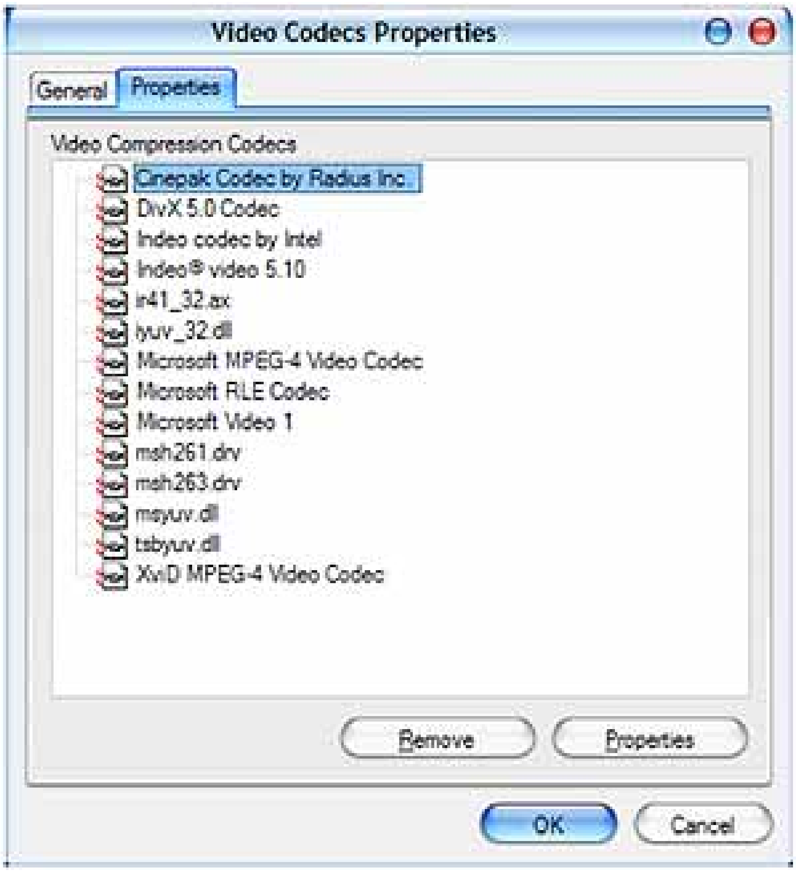
You can view the codecs installed in Windows XP:

Select Windows Start > Control Panel > Sounds and Audio Device Properties > Hardware

choose Video Codecs or Audio Codecs from the list > Properites.

Here is what I found...


Audio Codecs Properties
General
Properties
Audio Compression Codecs
IMA ADPCM Audio CODEC
Microsoft ADPCM Audio CODEC
Microsoft CCITT G.711 Audio CODEC
Microsoft GSM 6.10 Audio CODEC
DSP Group TrueSpeech(TM) Audio CODEC
msg723.acm
Windows Media Audio
Sipro Lab Telecom Audio Codec
Fraunhofer IIS MPEG Layer-3 Codec
Indeo® audio software
Voxware Compression Toolkit
Microsoft PCM Converter
Remove
Properties
OK
Cancel


Video Codecs Properties
General
Properties
Video Compression Codecs
Cinepak Codec by Radius Inc.
DivX 5.0 Codec
Indeo codec by Intel
Indeo® video 5.10
ir41_32.ax
Iyuv_32.dll
Microsoft MPEG-4 Video Codec
Microsoft RLE Codec
Microsoft Video 1
msh261.drv
msh263.drv
msyuv.dll
tsbyuv.dll
XviD MPEG-4 Video Codec
Remove
Properties
OK
Cancel

Summary of 4 codecs for each:

Audio: ADPCM, WMA, MP3, PCM

Video: DivX, MPEG-4, Microsoft Video 1, Cinepak

2 Bit-mapped graphic data

Bitmaps and compression (page 29)

Q1:

2 colour image size 400 x 300 pixels

2 colours needs just 1bit per pixel, so 400 x 300 bits are required

=120000bits

divide by 8 to convert to bytes

=15000Bytes

divide by 1024 to convert to Kilobytes

=14.6KB

16 colour image, size 600 x 400 pixels

16 colours needs 4bits per pixel, so 600x400x4 bits are required

=960000bits

divide by 8 to convert to Bytes

=120000Bytes

divide by 1024 to convert to Kilobytes

=117.2KB 256

colour image, size 600 x 600 pixels

256 colours needs 8 bits, or 1 Byte per pixel, so 600x600Bytes are required

=360000Bytes

divide by 1024 to convert to Kilobytes

=351.6KB

Estimate the file sizes, including the colour tables? (Do all three of these files need a colour table?)

All 3 of these images need a colour table.

Each entry in the colour table needs the colour code plus the 24bit RGB code

File (a)

2 entries in the colour table, each 1bit plus the 24bit colour code

=2 x 1 x 24

=48bits

divide by 8 to convert to bytes

=8Bytes

File (b)

16 entries in the colour table, each with 4 bit colour code and 24 bit RGB code

=16 x 4 x 24

=1536bits

divide by 8 to convert to Bytes

=192Bytes

File (c)

256 entries in the colour table, each with 8bit colour code and 24bit RGB code

=256 x 8 x 24

=49152bits

divide by 8 to convert to Bytes

=6144Bytes

divide by 1024 to convert to Kilobytes

=6KB

Of these three only the size of file (c) is significantly affected by the colour table, increasing its size to approx 357.6KB.

Q2: If using paint - NO

Q3: The image file size matches or exceeds predictions

Q4: If using paint - NO

Q5: The image file size matches or exceeds predictions

Q6: If using paint - NO

Q7: The image file size matches or exceeds predictions

Q8: Each bit of data represents a single pixel of the image, thus the bit pattern maps out the image.

24-bit images (page 30)

Q9:

Each pixel is represented by 3Bytes (24bits), therefore the size of the image file should be 3 times the number of pixels:

File size: 400 x 300 x 3 = 360000B = approx 350KB

File size: 600 x 400 x 3 = 720000B = approx 703KB

File size: 600 x 600 x 3 = 1080000B = approx 1.05MB

Bitmap formats (page 38)

Here I have given each image a rating out of 10 for quality (10=best quality).

Perhaps the photograph could have been better chosen as it does not show as much degradation when reduced to 256 colours, as some images do. Notice how the reduction of colours has little effect on the quality of the logo image (compare the 16 colour bitmaps for each image).

But the exercise still highlights the best image formats for each use: For photographs JPEG gives the best quality at lower file sizes; For logos GIF gives the best quality at lower file sizes.

Notice I have also included an extra row for RLE bitmaps - this illustrates how much this non-lossy compression scheme can reduce file sizes, particularly for images with solid blocks of the same colour.

	Photograph		**Logo**	
Format/Setting to use	**Image quality**	**Size**	**Image quality**	**Size**
Max quality JPEG	10	820	10	114
75% quality JPEG	10	312	10	57
50% quality JPEG	10	208	9	40
25% quality JPEG	10	134	8	31
Min quality JPEG	9	58	7	27
GIF	10	859	10	11
8 bit PNG	10	853	10	12
24 bit PNG	10	1905	10	17
24 bit bitmap	10	3601	10	980
256 colour bitmap	10	1202	10	328
256 colour, RLE bitmap	10	1144	10	26
16 colour bitmap	3	601	10	164
2 colour bitmap	1	151	3	42

Q10:

File sizes:

Transparent GIF: 20KB

Transparent PNG: 22KB

Yes, adding transparency seems to increase the size of the files.

Answers from page 46.

Q11: a) Bitmap

Q12: JPEG - JPEG uses lossy compression, so some image information would be lost, even with minimum compression.

GIF - can only use 256 colours - so colour information would be lost with these 'high colour' images.

Q13: b) 8 bits

Q14:

Colour depth (bits per pixel, bpp)	Number of colours: Calculated using 2^[colour depth] e.g. 2^8 = 2*2*2*2*2*2*2*2 = 256 colours
1	2
2	4
4	16
8	256
16	65536 (~65 thousand)
24	16777216 (~16 million)

Q15: PNG of these formats, only PNG files offer full colour with transparency

Q16: Bitmap: When it is important not to alter the original image in any way and the file size of the file is not important.

JPEG: When the file size of photographic images needs to be reduced.

GIF: When the file size of low-colour images needs to be reduced.

Q17: b) Anti-alias the image

Q18: The staircase effect is reduced and a higher screen resolution is simulated by altering the colour of pixels along the boundaries of contrasting colours, to produce a more blended effect.

Q19: d) 03 03 04

Q20: This signals the end of a row of pixels in the image.

3 Digitised sound data

Answers from page 68.

Q1: a) The number of times each second that a measurement is recorded of the sound

Q2: Sampling depth is the number of bits used to store each sample of the sound. The number of bits in the sampling depth determines the number of levels of the sound that can be represented, hence it is sometimes known as the 'sampling resolution'. This 'number of levels' of the sound is called the dynamic range of the sound, so the sampling depth (or sampling resolution) determines the dynamic range.

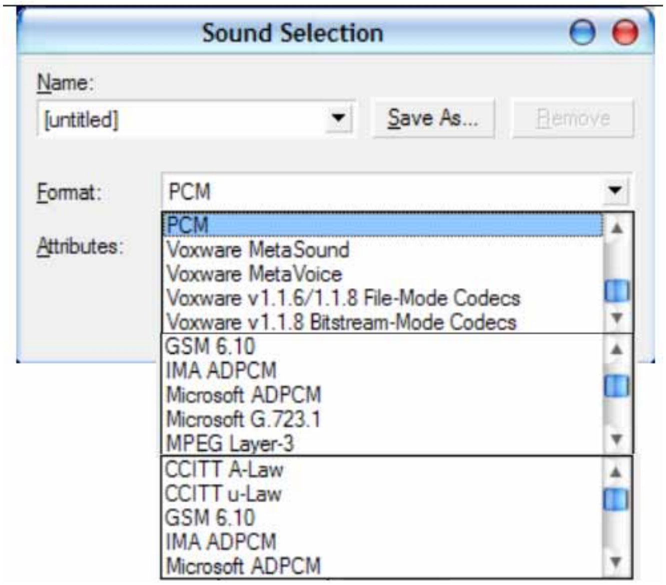
Q3: c) 16 bits

Q4:

Sampling depth (bits) (or Sampling resolution)	Dynamic range: Calculated using 2^[sampling depth] e.g. 2^8 = 2*2*2*2*2*2*2*2 = 256 colours
4	16
8	256
16	65536 (~65 thousand)
24	16777216 (~16 million)

Q5: b) Audio data files are so large that only lossy compression produces reasonable file sizes

Q6: I looked at Windows Sound Recorder. Here is the list of codecs it supports when recording sounds:



To summarise this: PCM, GSM, ADPCM, MP3 and CCITT are among the codecs supported.

Q7: c) Store it in MP3 format (keeping equivalent audio quality)

Q8: Using a stereo song of length of 3minutes (i.e. 180 seconds). Sound in the question: 30.3MB (by calculation)

a) 13.7MB (by calculation)
b) 15.1MB (by calculation)
c) 3MB (reduction can be up to 10% of original size)
d) 15MB (ADPCM reduces file sizes by about half)

You could confirm these sizes by experiment!

Q9: d) Normalise it

Q10: A: The sound would remain the same but the file size would be reduced.

B: The sound would be captured with a greater dynamic range, creating a clearer recording but the file size would be increased.

C: The sound would be distorted, becoming unclear but the file size would not be affected.

4 Video data

Answers from page 81.

Q1: a) The 'video in' on the graphics card does not allow analogue video to be captured

Q2:

Here is an example of what I found:

Video Capture Card 1

Sigma REALmagic DVR PCI

Encodes into MPEG 1 and MPEG 2

Max frame rate 60fps

Audio input on this card

Data transfer rate: 15Mbps

Max capture resolution: 704 x 480 pixels

Live preview of video as it is captured

Places little demands on computer system

Min system spec: 266MHz Pentium and 64MB RAM

Cost £699 plus VAT

Video Capture Card 2

Pinnacle Systems Studio DC10 Plus - Version 8

Encodes into MPEG 1, MPEG , AVI, MJPEG and RealVideo 8

Max frame rate: 30fps

Relies on PC soundcard to capture sound

Data transfer rate: 6Mbps

Max capture resolution: 768 x 576 pixels

Places big demands on computer system, so requires fast system processor

Min system spec: 500MHz Pentium and 128MB RAM

Cost £109 plus VAT

Graphics Card with video capture capability

Sapphire Radeon 9600Pro AllinWonder 128MB D/VIVO/DDR AGP

Encodes into MPEG 2

Max frame rate: 30fps

Requires soundcard to capture sound (I think!)

Max capture resolution: 720 x 480 pixels

System requirements: at least a Celeron CPU and 128MB RAM

This graphics card has additional features, but here I am only examining its video capture capabilities.

Cost: £153 plus VAT

Summary: More money buys a more capable video capture card that places fewer demands on the rest of the computer system. These best capture cards also support a greater bit rate and capture sound input along with the video. The hardware codecs in the specific capture cards are more varied than in the graphics cards. Can you think why capture cards do not offer greater capture resolutions than around 768 x 576, and why this *may* never be necessary?

Q3: d) 440Mbps

Q4: (640 x 480) x 30fps x 24bpp = 211Mbps

Q5: d) Any, information on the codec to use is contained in the header of the container AVI file

Q6: MPEG movies are stored as a key frame, which is a JPEG image, followed by a series of frames - these other frames are not entire images, but just store any changes since the last key frame. They use a lossy compression in which the quality / file size ratio can be adjusted.

Q7: a) 24fps, 600 x 400 pixels, 12 bits

Q8:

Video settings	bit rate	file size (25mins = 1500s)
24fps, 600x400 pixels, 24 bits per pixel	138Mbps	24GB
24fps, 600 x 400 pixels, 12 bits	69Mbps	12GB
6fps, 300 x 200 pixels, 12 bits	4Mbps	772MB
12fps, 600 x 200 pixels, 24 bits	35Mbps	6GB
12fps, 300 x 200 pixels, 24 bits	17Mbps	3GB

Q9: a) A transition was used

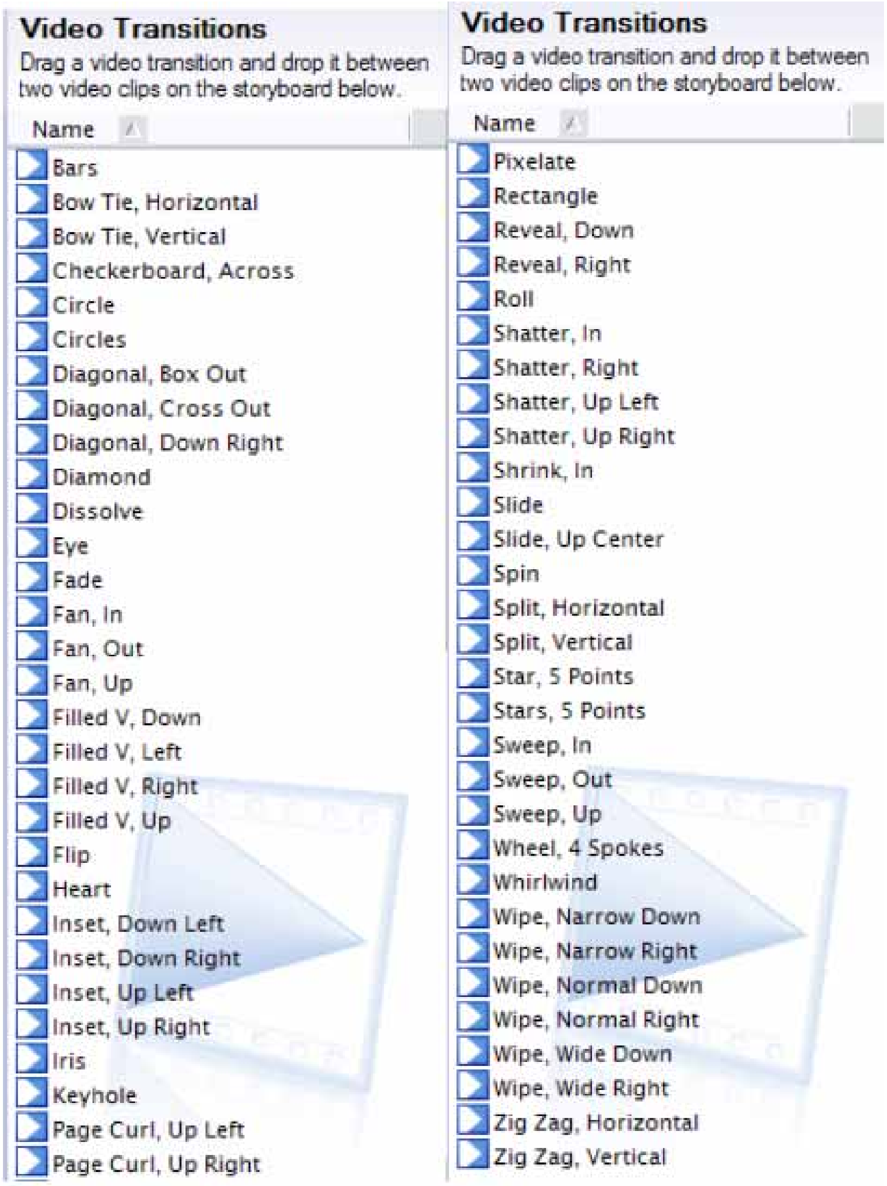
Q10:

An effect alters how the video is displayed on screen (similar to an effect in a painting or photo-editing program), a transition determines how one video clip ends, while another starts. I looked at the effects and transitions in Microsoft Windows Movie Maker v5.1:


Video Effects
Drag a video effect and drop it on a video clip on the storyboard below.
Name
Blur
Brightness, Decrease
Brightness, Increase
Ease In
Ease Out
Fade In, From Black
Fade In, From White
Fade Out, To Black
Fade Out, To White
Film Age, Old
Film Age, Older
Film Age, Oldest
Film Grain
Grayscale
Hue, Cycles Entire Color Spectrum
Mirror, Horizontal
Mirror, Vertical
Pixelate
Posterize
Rotate 90
Rotate 180
Rotate 270
Sepia Tone
Slow Down, Half
Smudge Stick
Speed Up, Double
Threshold
Watercolor

18 effects!

Transitions:



34 transitions!

When listing effects and transitions, variations of the same effect can be omitted, e.g. just 'wipe' transition, rather than listing all six variations of the wipe transition.

5 Object Oriented Data

Answers from page 115.

Q1: d) The file sizes are usually smaller than other types of image file

Q2: Individual elements (objects) in the image can be edited without affecting any other objects.

The quality of the image can often be improved by using different hardware or software to display it.

Q3: d) Pitch

Q4: For example: LINE

Start Point

End Point

Line thickness

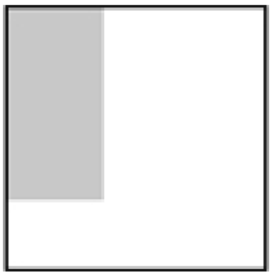
Line colour

Line pattern or style

Layer

(perhaps you can think of some other attributes for a LINE?)

Q5: b)



Q6: A: <rect x="0" y="0" width="40" height="20" fill="rgb(200 200 200)"/>

C: <rect x="0" y="0" width="40" height="20" stroke="rgb(200 200 200)" stroke-width="5"/>

D: <rect x="0" y="40" width="40" height="20" fill="rgb(200 200 200)"/>

Q7: b) rotation 0 0 1 0.79

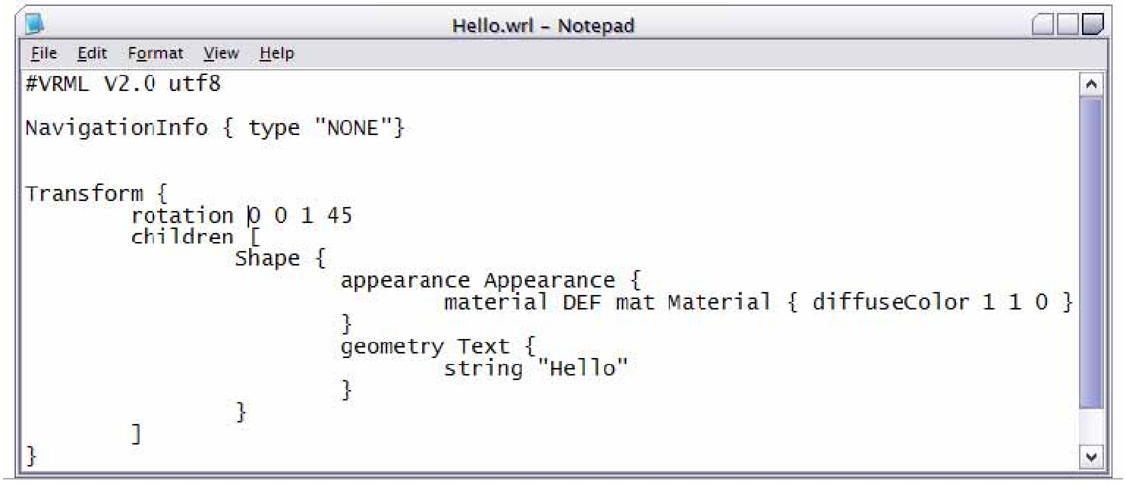
Q8:

A: Tilted towards the viewer (around the X axis) by 0.79Rad (45 degrees)

C: Tilted towards the viewer by 45Rad (2578 degrees! - 7 full turns plus 58 degrees)

D: Rotated around the Z Axis, seven times plus 58 degrees.

Here is the VRML code used for D:




```
#VRML V2.0 utf8

NavigationInfo { type "NONE"}


Transform {
        rotation 0 0 1 45
        children [
                Shape {
                        appearance Appearance {
                                material DEF mat Material { diffuseColor 1 1 0 }
                        }
                        geometry Text {
                                string "Hello"
                        }
                }
        ]
}
```

Q9: d) The music is not as realistic

Q10: The quality of the sound when played back cannot be guaranteed as it is dependent on the soundcard used to play it back.

Vocals cannot be included in a MIDI file.

6 Trends and contemporary technologies

Answers from page 124.

Q1:

A standard Smart Phone

Smartphones run mainstream operating systems (e.g. Linux) this turns them from a telephone into a full computing platform, with built in sound functions, networking, digital camera and support for powerful software applications such as playing videos or 3D java gaming.

Q2:

Digital cameras:

Many digital cameras offer more than just the ability to take and store photographs - with video capability, sound recording, MP3 playback and use as a web-cam.

Media centre computers:

Computer systems are beginning to enter living rooms, with media centre computer systems being styled more like DVD players, having remote controls, connectivity for televisions and the capability to control all the homes entertainment and broadcast it around the house using a wireless network.

Internet fridge:

these fridges with Internet connectivity act as media centres for your kitchen, allowing viewing of TV, checking of email, playing of music or any other task that could be performed by a computer system. E.g. see http://www.lginternetfamily.co.uk/fridgedemo.asp

Wristwatch PDA:

The shrinking in size of devices is a major force behind convergence. Wristwatches are prime targets for all sorts of convergence ideas. Here is a wrist-worn PDA from Toshiba that also tells the time! http://www.pdalive.com/showarticle.php?threadid=5809

Q3: b) Firewire 1394b

Q4:

The following data transfer rates are for 'burst modes', i.e. short transfers of data, the rate drops for longer, sustained, transfers of data; or are for good wireless connections, a poorer connection (e.g. over a greater distance) will also lower the data transfer rate.

- USB 2.0: 480Mbps
- Firewire 1394a: 400Mbps
- Firewire 1394b: 800Mbps
- WiFi 802.11b: 11Mbps
- Bluetooth v1.2: 721Kbps
- Bluetooth EDR: 2.1Mbps (Enhanced Data Rate)

Q5: d) Holographic

Q6:

Solid state silicon: Non-volatile, solid state memory is sold as various technologies, e.g. flash memory, compact flash, pen drives and other formats of removable card. This technology is currently fairly advanced with limits predicted by the size of the molecules on the silicon (although this could be extended through the quantum [i.e. sub-atomic] technologies or the incorporation of magnetic technologies [see magnetic RAM - MRAM]). The technology involved with investigating these devices is called 'nanotechnology'. Devices are available at the time of writing in capacities of up to 2GB with greater on the horizon.

Optical disc: It could be argued that holographic storage is an optical technology, as lasers are used to read and write data to the medium. A further distinction is then: surface optical technologies and volumetric optical technologies (for simplicity - the definitions shall currently be left as they are!)

In order to pack data closer together on optical discs light with a small wavelength is used, thus the emergence of DVD drives using blue lasers (Blue ray, Blu-Ray or Blue Laser). These devices currently hold 23GB of data on a single layer. Sony has announced a dual-layer drive - providing 50GB of storage on the single side of a disc. See http://www.theregister.co.uk/2004/03/26/sony_preps_50gb_nextgen_bluray/

Magnetic: Hard drives are (at the time of writing) available in capacities up to 400GB (Terabyte drives cannot be far off!) and magnetic tapes can provide capacities of up to 500GB, but the greatest data densities go to the tiny hard disc drives, called microdrives, that fit in Compact Flash slots with capacities of 4GB and soon, 8GB.

Holographic: Not yet at production stage, but research is indicating capacities of around 1TB (1000GB) in a 3.5inch (hard disc drive) size. This diagram from a paper written in 1999 (http://www.research.ibm.com/journal/rd/443/ashley.html)

Now that this technology is nearing production, initial devices will be below this capacity, for example a drive predicted for 2005 is expected to have a capacity of 'only' 200 to 300GB on a 12cm disc. see http://addict3d.org/index.php?page=viewarticle&type=news&ID=1818 NB In 1970 predictions were made to the ultimate limit of recording on magnetic disc of 2Mb per in2 up to 130Mb per in today's best drives operate at nearly 100 times the latter limit! So who knows what the predictions for holographic technologies hold?

Q7: a) Cheaper, more powerful, smaller in size and using less power

Q8:

There are many ways to measure or report the speed of a processor, for example:

- Clock Speed (measured in Hertz, Hz)
- The number of instructions or operations carried out each second (measured in MIPS - Millions of Instructions Per Second - or FLOPS [look it up!])

NOTE: These are not useful taken out of context as they do not take into consideration, for example, the architecture of the processor, or if considering the operation of the system as a whole, the other components involved.

As many processing tasks are shared by processors scattered around computer systems (not just the central processing unit) then the speed at which a system performs tasks (processes data) is dependant on the entire system.

A common (although not entirely un-controversial) way to compare overall system performance is to measure how quickly each system can perform the same set of tasks, called a BENCHMARK.

Q9: c) The position of the viewer's head must stay within a narrow range to experience the 3D effect.

Q10:

General references:

http://www.neotek.com/3dtheory.htm and http://www.3dcgi.com/cooltech/displays/displays.htm (1)

link (1) above shows a currently available autostereoscopic 3D display - from Dimension Technologies Inc.

Option A is a drawback of shutter glasses or any display technology that requires switching between images quick enough to try and fool the eyes that they are both seeing different images at the same time

See http://www.stereo3d.com/shutter.htm

Option B is a drawback of Anaglyph technology - i.e. red and blue glasses

Option D is a drawback of head mounted displays (HMDs) This may be addressed in newer headsets due to improving 2D display technologies like flexible OLED displays.

Extra notes on Flexible Displays: Flexible displays are possible due to a few technologies, i.e. Organic LED (OLED) displays or electronic inks (E-inks). These allows the plastic light emitting components (e.g. PolyLEDs or inks) to be incorporated into thin plastic film.