Multimedia Data

Information management has traditionally been restricted to data which is entirely numerical (including dates, currency, etc.) or unstructured (and small) text strings.

There then arose a demand to extend data management to other kinds of data.

In particular:
- Structured and Formatted Text
- Graphics
- Audio
- Video
- Animation
- Virtual Reality (3D Graphics)
- Haptic Input

In general, there are two techniques for representing these kinds of data:

- as a model of the **underlying information** – e.g. this scene contains a chair which is made up of legs, etc.;
- as storage of **perceptual features** – e.g. storing pictures of the chair

"Digital Multimedia" by Nigel Chapman and Jenny Chapman, 2ed, Wiley, ISBN 0-470-85890-7

Formatted Text

Text is made up out of **characters** and this presents two questions -

- which characters can the computer store?
  - each distinct entity (letter, ideogram, punctuation) requires a separate computer representation
  - this is called a **character set**
- what should they look like?
  - the appearance of each character must be distinct
  - a (complete) set of visualisations of the set of characters is called a **font**
  - for any given character set there can be many different fonts depending on the use to which they are being put
  - fonts are different in two ways:
    - how **readable** they are
    - their **emotional impact** – see link on web site

Character Sets

A **character set** associates each character with a different bit pattern (c.f. numbers) and conventionally this mapping is represented as a mapping from an integer to a character

e.g. in ASCII, 'A' is the 7-bit bitstring 1000001 or 65 in decimal

There are many possible character sets depending on the context:

- English requires 52 (upper and lower case) letters, 10 digits and some punctuation
- other European languages add accents to the same alphabet (ê, õ) and some extra punctuation (¿)
- Cyrillic and Greek have different letters
- Japanese Kanji has from 1945 to 6000 ideograms
- there are also mathematical symbols, etc.

Desirable Features of Character Sets

Some desirable features of a character set are:

- using any **conventional order** of members of the character set
  - e.g. the alphabet – 'A' is followed by 'B', etc. so if A is 65, B is 66
  - or the digits - '0' is 48, '1' is 49 etc. – *in ASCII*
- **densely packing** character subsets
  - the digits are together (48-57) so you can test if character C is a digit by:
    '0'<= C && C<='9'
  - the capital letters are similarly grouped together (65-90) and so are the lower case letters (97-122)
- exploiting the **binary system** so that one bit determines which character subset there is
  - 'A' is 65 = 1000001, while 'a' is 97 which is 1100001, so case is determined by just one bit

Character Set Standards

These standards dominate:

ASCII code is a 7-bit code (128 characters) for English letters, digits and punctuation, and characters representing control codes such as end-of-line

ISO 8859 is a set of 8-bit codes (256 characters), in which the first 128 characters are ASCII and the last 96 codes (32 are left unused) are used for different alphabets:

- ISO8859-1 adds accented letters for Western European languages
- ISO8859-2 adds letters for Eastern Latin alphabets
- eight others add letters for Cyrillic, Modern Greek, Hebrew, etc.

ISO 10646 is a 32 bit code which structures all known characters in a 256 x 256 x 256 x 256 hypercube

UNICODE is a 16-bit code defining 39,000 symbols from most major alphabets and a coalescence of the ideograms of Japan, China and Korea

Fonts

The appearance of a character is called a **glyph**

The association of a character set with a set of glyphs is called a font, which vary in a number of ways:

- they can be **monospaced** (all characters have the same fixed width – e.g. `Courier`) or **proportional** – e.g. Helvetica
- they can be **serifed** (i.e. have little strokes added to improve clarity – e.g. Lucida) or **sans serif** – e.g. Arial
- they can have different **shapes** – e.g. upright, *italic* or ***calligraphic***
- they can varying degrees of **weight** – i.e. boldness
- they can have different **point size** – where a point is supposed to be 1/72 of an inch
- they can be stored as **bitmaps** or **vectors** (i.e. outlines of the characters)

Controlling the Formatting

The appearance of characters in a document is affected by control information in the document

This appears in two places in the document

- a series of **definitions** appear at the **top** of the document
  - mostly of abbreviations used later on
- **control codes** appear in the **middle** of the document indicating a change of formatting from this point on
  - for instance, there will be control codes to start emboldening and to switch it off

This information can be added **explicitly** (e.g. LaTex or HTML) or **implicitly** (using a WYSIWYG interface such as MS Word)

- although the essential nature of the document is no different

Formatting a Whole Text

As well as formatting individual characters, the arrangement of the text on the page can be organised.

- This is a matter of organising the control codes into groups called **styles**

A style controls:

- character appearance
- margins and tabs
- justification
- borders, etc.

Styles and document structure are clearly intimately related but have a different purpose

- Document **mark-up** is used for both

  i.e. placing control codes (usually as **tags** enclosed in angle brackets) into the document to identify fragments

Style sheets can be used to associate formatting styles with structural components – e.g. HTML and Cascading Style Sheets

Structured Text

Text in databases consists typically of small text strings, which is useful for names, addresses and so on, but not for documents

- To capture a complex text, the different components of the structure must be identified
- One way of doing so is to use languages created using a meta-language called **Standard Generalised Markup Language** (SGML)

SGML permits the description of mark-up languages in terms of tags:

- i.e. a text described in a language defined in SGML breaks up the sequence of characters into chunks delimited by start and end tags
- describing a language in SGML consists of defining which tags are available and what they mean
- the most common language implemented in SGML is HTML - described in a later lecture
- a very important simplified version of SGML, used throughout the Internet and for data exchange, is XML – also described later

Compression

Multimedia files are frequently large and regular (e.g. large areas of the same colour), so they often come in a compressed form, one of:

- a form which identifies **repeated patterns** and avoids the repetition – this is particularly useful for text and images
- a form which identifies **redundant perceptual information** – more important for sound and video

Some compression techniques are **lossy** - i.e. you lose some of the information – others are **lossless**

Compression algorithms are compared on a number of measures:

i) how compressed they are terms of the number of bits per symbol
   - e.g. the best English text compressors (the RK algorithm) manage about 1.4 bits per character
   - this is close to optimum – experiments show that there are about 1.3 bits per symbol of information in English text

ii) how fast they are to compress and to decompress

Some Compression Techniques I

Run Length Encoding (RLE)

Sequences of the same values are reduced to a description of the value together with the number of repetitions

e.g. **5, 5, 5, 5, 5, 5, 5, 5 becomes 8, 5**

Huffman Encoding (a form of **entropy** encoding)

Repeated sequences are identified and the most common ones are replaced by short bit strings, the rarer ones by longer bit strings

- e.g. if a text has lots of instances of "the " then this might be assigned 001 and nothing can be assigned 0010 or 0011 since strings starting with 001 have been used up, while the few instances of "Richard " might be assigned 011010011. The result is a long bit stream which exactly replaces the text.
- The associations of the short strings to the long are placed in a dictionary at the start of the file

Lempel-Ziv Welch Compression (LZW)

Repeated patterns of bytes are identified and a dictionary of common sequences is built.

e.g. if 5,6,7,8,9 appears several times it will become a dictionary entry, say the 20th and the sequence will be replaced by the 20.

however, this was patented and so could not be freely used until patent ran out 2006

Some Compression Techniques II

Joint Picture Expert Group (JPEG)

- Pictures can be described into its frequency components

  (often done using a Fourier Transform, but in this case a Discrete Cosine Transform is used)
- This turns an array of pixels into an array of coefficients
- The coefficients can then be compressed
- Higher frequencies are compressed into fewer bits than lower frequencies since they contribute less to the perceived quality
- Many coefficients are zero – thus RLE can be used
- Huffman encoding is used for the rest
- The best feature of JPEG encoding is that the amount of compression (and therefore quality) can be easily controlled

Graphics

Modern applications usually require a graphical component

- Since there are a very large number of graphics file formats, it is difficult to accommodate all of the graphical data that is available

There are basically two forms of graphics files:

- **bitmap**, **raster** or **image** graphics files store a rectangular array of pixels
  - these are the kind of files used by "Paint" programs
  - they are good for storing scanned photos, etc.
  - but are hard to scale
- **vector** graphics files store the objects in the image rather than the pixels
  - these are the kind of files used by "Draw" programs
  - they are good for drawings and diagrams
  - objects are stored as a series of functions rather than a sequence of pixels
  - they are easier to scale
- There are also metafile formats which encompass both of these

Bitmap Image Formats

A **bitmap** is an array of pixels, where:

a **pixel** indicates what is displayed at a single point of the screen

The most important feature of a bitmap is its **depth -** i.e. the number of bits used to describe each pixel. This is usually one of :

1 - the pixel is either on or off - useful only for monochromatic images

8 - the pixel can now distinguish 256 values - usually 256 different colours or greyscale – ok for many purposes

24 - the pixel can now distinguish roughly 16,000 colours - usually broken up into three parts, eight bits for each primary colour – need for print quality

A bitmap image file typically contains the following parts:

- a **header** which contains a description of the structure of the file - its size, its type, compression used, etc.
- a **colour palette** - see colour slides
- the array of pixels
- and sometimes a **footer**

Physical and Logical Pixels

A bitmap image file will specify a certain number of pixels in each direction – the **logical pixels**

A device such as a screen or a printer will have different sizes – the physical pixels

Scaling the logical pixels to fit onto a screen can often cause strange effects, such as jagged lines or striations in a pattern

Be careful to use appropriate techniques to get around these problems:

e.g. **anti-aliasing** displays a straight line using gray levels for each pixel – the further away the pixel is from the true line, the lighter it is shown

Colour

There are four (roughly equivalent) systems for colour:

RGB (Red Green Blue) is an **additive** technique in which each colour is defined in terms of the amount (between 0 and 255) of redness, greenness and blueness it has

CMYK (Cyan, Magenta, Yellow, Black) is a subtractive technique in which the amount of the colours which absorb red (cyan), green (magenta) and blue (yellow) is given together with black to add depth – again each having a value between 0-255 using eight bits

HSV (Hue, Saturation, Value) describes a colour in terms of the colour it is (hue), the amount of white mixed with this colour (saturation) and the brightness (value)

YUV (Luminance, Blue Difference, Red Difference) describes in terms of luminance (Y = 0.2125*R+0.7145*G + 0.0721*B) and B-Y and R-Y, since this separates brightness and colour better

- **$Y'C_BC_R$** is a variety of YUV with different parameters used in digital TV

A Table of Comparisons of Colour Systems

	RGB	CMY	HSV
Black	0, 0, 0	255, 255, 255	?, 0, 0
Red	255, 0, 0	0, 255, 255	0, 240, 120
Yellow	255, 255, 0	0, 0, 255	40, 240, 120
Grey	127, 127, 127	127, 127, 127	?, 0, 120
Green	0, 255, 0	255, 0, 255	80, 240, 120
Light Grey	191, 191, 191	64, 64, 64	?, 0, 180
Cyan	0, 255, 255	255, 0, 0	120, 240, 120
Blue	0, 0, 255	255, 255, 0	160, 240, 120
Magenta	255, 0, 255	0, 255, 0	200, 240, 120
White	255, 255, 255	0, 0, 0	?, 0, 240

Two Kinds of Colour

In fact we use colour in two fundamentally different ways:

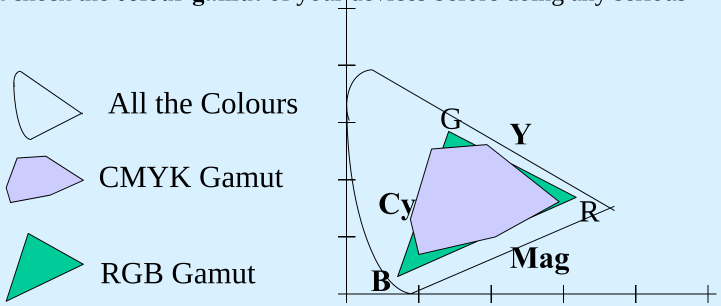
- When the computer displays an image, it **projects** light outwards from the screen as three coincident beams of red, green and blue that jointly can create just about any useful colour
  - In this case, an additive model works well
  - RGB is therefore the most common for computer-displayed files
- A real world object being observed is **reflecting** part of the light and **absorbing** the rest. The chemicals on the surface of the object determine which colours are absorbed. For a printed image, the pigment is constructed with this in mind, so to create a red section, the dye will be used which absorbs cyan. Therefore a subtractive model is appropriate.
  - CMYK is used for publishing. The reason for the black is that the dyes used for absorbing cyan, magenta and yellow when added together don't do a very good job of absorbing everything and creating black.
  - Therefore, extra black needs to be added

Problems Using Colour

Mapping between the two is very tricky – a program like PaintPadPro lets you do it, but the results are often unsatisfactory – this is a very skilled job

This is further complicated by the fact that different monitors will display the same image in different colours

You should check the **colour gamut** of your devices before doing any serious work



Colour and Pixels

To give full RGB colour requires 24 bits which means the files will be quite big, but is often used in practice

Many systems work with fewer than 24 bits (e.g. 8) and to do this they use a **colour map** or **palette**

A palette is essentially a look up table between a pixel value and a 24-bit colour description. Thus you might choose 3-bit pixels and have a palette like:

0	->	0, 0, 0	black
1	->	255, 0, 0	red
2	->	255, 255, 0	yellow
3	->	0, 255, 0	green
4	->	255, 0, 255	magenta
5	->	0, 255, 255	cyan
6	->	0, 0, 255	blue
7	->	255, 255, 255	white

Vector Image Formats

Vector image files have a similar structure to a bitmap data, except that the array of pixels is replaced with a list of graphical objects

The objects describe:

the shape

the colour

the filling pattern

Vector files efficiently represent regular data such as drawings but are not useful for pictures

Extended vector formats are similar but can represent three-dimensional data

Layers

Images are often constructed in the form of **layers**

- these are equivalent to clear sheets of acetate which can be drawn on and then combined
- they can be used to separate different parts of the image construction or to achieve a wide range of effects

Combining the Two Kinds of Graphic Format

Although the two forms are separate, they can be transformed into each other:

- **rasterisation** takes a drawing and makes a bit map of it
- **vectorisation** takes a bitmap and identifies lines and so on, resulting in a vector graphics image

Some Standard Image Formats

There are many formats but the most important are:

Windows Bitmap - stores a single bitmap using RLE and allows up to 32 bits for colours

GIF - Compuserve created format which dominates the web - uses LZW compression, up to 8bit colour,

JFIF - An interchange format which supports up to 24 bit colours, but uses the lossy JPEG compression technique (usually known as .jpg files)

TIFF - A very rich interchange system which supports up to 24 bit colours and several compression techniques. The header of a TIFF file can have a variety of tags defined in it- other formats typically have a fixed set.

PNG - Portable Network Graphics - intended to supersede GIF since it doesn't use LZW

Postscript – a vector image format that is used for printing and page layout

Scaleable Vector Graphics (SVG)

SVG is an XML-based description of three kinds of graphic objects for use over the internet:

vector graphic shapes; **images**; and **text**

SVG documents are made up of a list of shape elements

<svg> is the document type

<rect> defines a rectangle

<circle>, <ellipse>, <line>, <polyline> and <polygon> are others

<path> defines is a line made up of multiple segments

Each of these has a number of attributes:

- **shape attributes** are required for each kind of shape, e.g. a circle has a centre and a radius
- **stroke and fill attributes** determine the look of the shape border and interior
- **transform attributes** allow the shape to be moved, scaled, rotated or skewed

Virtual Reality

There are three main kinds of VR system accessible over the web:

- **Modelled VR** – i.e. provide descriptions of 3D scenes– this is equivalent to the vector graphics files
- **Panoramic Imaging** - e.g. QuickTime VR - integrate a series of photographs into a 360 degree view, either from the inside (e.g. round a room) or the outside (e.g.round an object) – this is equivalent to bitmap graphics
- **Multi-user shared VR** - These are sites which can be concurrently accessed by multiple users. Each user is represented by a figure and can move around and explore a world

Among the languages for modelling 3D are:

- **VRML** (Virtual Reality Modelling Language) describes the structure of the world in marked-up text
- **Java 3D** – A Java API
- **Open GL** – a low level API from Silicon Graphics

Fundamentals of VR - I

The basis must be a three dimensional description of the **3D scene** in terms of Cartesian (x, y, z) or Projective (x, y, z, w) geometry

- usually this is hierarchical – i.e. a leg on a person in a room
- often it is made up of cubes and cylinders, etc.

Then there must be control of the **point of view**, i.e. where the viewer is in the scene

The view is then **rendered** onto the screen using geometric transformations which

a) place all 3D points into the point on the screen where the eye expects them using (e.g.) **perspective**

b) **hides** any points which are behind other points

Fundamentals of VR - II

The look of the scene can then be made more or less realistic:

- the easiest and fastest representation to use is the **wire frame** model

  i.e. a set of lines which make up the edges of the objects and create a transparent image

- you may then fill in the surfaces and place **textures** on the surfaces or use **colour**
- you may add **lighting effects** to create shadows, etc.
- you may add other effects such as **fogging** or **dimming** objects which are further away

Haptic Interfaces

Haptic devices attempt to use our sense of touch to interact with software

There are two basic mechanisms

- **force-feedback devices** which resist movement in a way which can be programmed
- **textured devices**

They can be used

- for training in sensitive or dangerous tasks
- to support people who are visually impaired, etc.

They are only really effective in a multi-modal setting - i.e. if used in close collaboration with the other senses (e.g. sound or visual feedback)

- They may have their best use in conjunction with VR

Audio

Audio files contain data which can be played back as sound

There are four main parameters of any format:

- **sampling frequency** - the closer together the samples are the bigger the file would be but the better the reproduction
- **bits per sample** - usually either 8 or 16 bits, the latter has much better quality
- **number of channels** - usually one (mono) or two (stereo)
- compression **lossiness**

There are two main kinds of format (c.f. graphics)

- those which just record the sound
- those which store a description of the sound (e.g. MIDI)

The Main Sound Formats

RIFF WAVE (.wav extension) - The Microsoft format which is common on the web.

- **Sun µLaw** (.au extenion) - The other most common format on the web.

AIFF – Apple and SGI format again very common

MP3 – the standard for music clips, compresses 10:1 while maintaining quality

- uses perceptual features of the sound to determine the compression

MIDI – Musical Instrument Digital Interface System - used for synthesisers as well as computers:

- permits a much higher level description – e.g. in terms of the instrument to be used
- messages are sent to the digital instruments, e.g.
  - *program change* to instrument (numbered 1-128)
  - *note on* note (numbered 0-127), attack
  - *note off* after touch, pitch blend

Video I

Video started out as the capture of analogue colour TV signals which has three dominant standards, each with its own features (such as number of lines per frame):

- NTSC – USA, Japan, Taiwan, etc;
- PAL – most of Western Europe, Australia and New Zealand
- SECAM – France and Eastern Europe

Digital video is dominated by the **CCIR 601** standard which unifies the above by allowing non-square pixels

- uses $Y'C_BC_R$ colour
- each analogue line is turned into 720 luminance and 360 of each of the colour difference values
- NTSC becomes 720 x 480 pixels; PAL becomes 720 x 576 pixels

Video II

On top of that, two main standards are emerging based on CCIR with further **chrominance sub-sampling**

- sampling luminance more than colour differences

They are:

- **DV** – used for the home and semi-professional market
- **MPEG-1 and -2** – used by professionals
  - **MPEG-4** is more extensive and supports other multimedia

Compression of video can take two forms:

- **spatial compression** compresses data within a frame (as for an image)
- **temporal compression** compresses data between frames using **key frames** (others are described as differences from the nearest key frame)

The mechanisms for compressing and decompressing video signals are called **codecs**

- they may be installed into hardware or be software components

Animation Tools

One simple way of providing animation is to make a multi-image graphics file and to have the browser flick through the images -

- **animated GIFs** were the first common source of animation of the web

- The standard for producing animations of reasonable quality is **Macromedia Director**
- **Flash** is the more popular alternative to Director (also from Macromedia) for producing quick and simple animations
  - the file format for Flash is SWF (Shockwave Flash) which is compressed for internet delivery – Director can also produce this
- Writing an animation program - e.g. **Java applets** - is another popular method

Animation Basics

The basis is a rectangular grid called the **score**

- left to right defines the time line
- the time line is broken down into key frames which show the important points in the animation
- each line describes the way in which one cast member moves in the movie
- there is also another frame called the **stage** for creating characters

An animation is built up as a set of layers called **cels**

- each will describe the behaviour of one aspect of the animation – perhaps one character or cast member
  - these are called **sprites**

A sprite is described by a set of images (called **faces**) which show the character in the main ways in which it is to be seen

- e.g. a walk cycle

Animation Techniques

Motion Tweening

- The frames in between key frames are created by interpolation

Shape Tweening or Morphing

- One shape changes into another gradually

Co-ordinating Multimedia Displays - SMIL

Synchronised Multimedia Integration Language (SMIL pronounced smile) is an XML language which allows you to:

- place multimedia elements wherever you want on the screen
- synchronize those elements
- support user-preferences such as language, bit-rate, etc.

A SMIL document has the following parts:

- a **layout** section which identifies regions of the page and how components fill that space
- a series of **media tags** for audio, video, etc.
- attributes to control the **time** when a component appears and disappears
- tags to specify appearances in **parallel** or in **sequence**
- attributes to display the page **differently** depending on the language, bit-rate and so on

SMIL Layout

The layout section specifies regions of the screen

- **absolutely**:
  ```
  <region id="Region1" left="30" top="30" width="40" height="40" />
  ```
- **relatively**:
  ```
  <region id="Region2" left="20%" top="20%" width="40" height="40" />
  ```
- how **overlap** works:
  ```
  <region id="Region3" left=... z-index="1" />
  ```
  - 1 means on top, 2 would be next to the top, etc.
- how the data **fits** with the region - the fit attribute can be:
  fill - distort the data to fill the whole region
  meet - fill without distortion so that one dimension fills, the other is shorter
  slice - fill without distortion so that one dimension fills, the other is longer
  scroll - add scrollbars

SMIL and Synchronisation

Multiple objects can be synchronised as:

- a **sequence** using a *<seq>* tag:
  ```
  <seq>
     <img src="myPic.jpg" alt="My Picture" region="Region1" dur="6s" />
     <text src="myText.txt" region="Region2" dur ="10s" begin = "4s"/>
  </seq>
  ```
  - The picture appears for 6 seconds, there is a gap of 4 seconds and then the text appears for 10 seconds
- **in parallel** using the *<par>* tag:
  ```
  <par>
     <img src="myPic.jpg" alt="My Picture" region="Region1" dur="6s" />
     <text src="myText.txt" region="Region2" dur ="10s" begin = "4s"/>
  </par>
  ```
  - The picture appears for 6 seconds, the text appears after 4 seconds from the start and lasts for 10 seconds

Streaming

The previous slides mainly describes files, which may be accessed, downloaded and then "played“ on your home machine

Another useful technique avoids the file system and plays the file as it is downloaded

- this has the great advantage that the server maintains ownership of the clip

RealPlayer are the main purveyors of this facility

- WinAmp, Quick Time, Windows Media Player are others

With a Real Player player you can:

- download and play video or sound clips
- listen to radio stations all over the world, etc.

Each streaming system has

- renderers including plug-ins for the main browsers
- producers for creating streamed forms of sound and video

Streaming Protocols

HTTP is the main protocol for sending data over the internet

- It is secure and universally available
- However it’s reliability is at the cost of unacceptable speed for streamingf

RTSP (Real Time Streaming Protocol) has been designed to be more appropriate

- It uses UTF-8 characters rather than ASCII
- The client finds the recourse and its presentation description and then connects to it
- Then it can send PLAY (including how long) and PAUSE requests
- and ultimately a TEARDOWN request to terminate it