

Higher Computing Science

Media Types

Summary notes

Compression

- Appropriate use of compression can reduce file sizes.
- Small file sizes can be transmitted and processed more quickly.

For example, a whole webpage can be compressed, reducing the amount of data which has to be sent to the client computer and therefore reducing the load time. The browser will decompress the page before displaying it.

Individual media elements (audio, graphics, video) within the webpage could also be compressed in order to reduce the size further.

- **Lossless compression** reduces file sizes without removing any data.
- **Lossy compression** reduces file sizes by removing some data.

Audio formats

MP3 (.mp3)	Lossy compression	<ul style="list-style-type: none"> ▪ Uses perceptual coding compression. ▪ Removes the very high or low frequencies that humans cannot hear (i.e. <20 Hz and >20,000 Hz). ▪ When two similar sounds (but different volume) occur at the same time the quieter one is removed. ▪ Lower frequency (bass) sounds are converted to mono. ▪ User can choose the level of compression. ▪ A full compression MP3 compresses data to around 10% of the original size, with relatively little loss of quality.
FLAC (.flac)	Lossless compression	<ul style="list-style-type: none"> ▪ An open source lossless compression algorithm for audio. ▪ Using a technique called rice encoding to try to balance difference between what is predicted and what was sampled (linear prediction). ▪ Using a form of run length encoding to store any sequences of identical samples. ▪ A typical FLAC compresses data up to 50% of the original size, with no loss of quality.

Audio file size calculations

To calculate the file size (in bits) of an uncompressed audio file the basic formula is:

sample frequency (Hz) x sample depth (bits) x length (seconds) x number of channels

- **Sample frequency** is the number of times per second that a sample of the original sound was taken.
1 Hz = once every second. 1 KHz = 1000 times every second.
- **Sample depth** is the number of bits that were used to store each sample.
- **Length of sound** should always be converted into seconds if necessary.

- **Number of channels** is typically 1 (mono) or 2 (stereo), although it could be more for a surround sound file (e.g. 6 channels for 5.1, 8 channels for 7.1 etc.)

Example

Calculate the file size of a mono voice recording captured at 24KHz with a sample depth of 8 bits, lasting for 15 seconds.

$$24000 \times 8 \times 15 \times 1 = 2880000 \text{ bits}$$

$$2880000 / 8 = 360000 \text{ bytes}$$

$$360000 / 1024 = 351.56 \text{ KB}$$

Graphic formats

JPEG (.jpg)	Lossy compression	<ul style="list-style-type: none"> • 24 bit colour. • Allows the user to select an appropriate level of compression. • JPEG is a good compression format for high-quality and detailed images and images where quality is unimportant (e.g. thumbnails). • Does not support transparency or animation.
PNG (.png)	Lossless compression	<ul style="list-style-type: none"> • Intended to improve upon GIF format. • 24 bit colour. • Allows user to set alpha level for improved transparency. • Does not support animation.
GIF (.gif)	Lossless compression	<ul style="list-style-type: none"> • Restricted to 8 bit colour. • Uses LZW algorithm to compress data. • Several images can be stored in one GIF file, allowing for animation. • Allows a specific colour in an image to be made transparent. • GIF is a good compression format for images with areas of flat colour, images that have few colours, animated images or images that require transparency.

DCT (Discrete Cosine Transform) encoding is a lossy compression algorithm.

- It works by converting an RGB (red, green, blue) image into three components known as YCbCr, which splits the pixels by colour and brightness (luminance).
- The image is then split the into 8x8 pixel blocks.
- Both a cosine and a huffman function are used to eliminate frequencies used to represent colours that do not appear often in the 8x8 block.
- Used for JPEG compression.

It is often the case that the data that has been removed will go unnoticed by users unless they apply a very high compression ratio at the time of DCT encoding.

RLE (Run Length Encoding) is a lossless compression algorithm.

- It works by taking runs of the same colour and storing them as two numbers: *how many pixels* and *colour code of this run of pixels*.
- RLE works best for images with large blocks of the same colour.
- If an image has many different colours (e.g. a photo) RLE could potentially end up increasing the file size!

LZW (Lempel Ziv Welch) is a lossless compression method which is more effective than RLE.

- It works by looking for repeated patterns of bits within data, stores them in a dictionary, then replaces them with references to the dictionary.
- Used for GIF compression.

Indexed colour involves storing the actual RGB codes used by the image in a separate colour palette, in order to reduce file size. This can be used as a compression technique where the bit depth of the graphic is less than 24.

Graphic file size calculations

To calculate the file size (in bits) of a bitmap graphic file the basic formula is:

$$\text{number of pixels in image} \times \text{bit depth}$$

- **Bit depth** (aka colour depth) is the number of bits that were used to store the colour of each pixel.

Example 1

Calculate the file size of a 400 x 400 pixel graphic with a colour depth of 24.

$400 \times 400 = 160000$ pixels in the image

$160000 \times 24 = 3840000$ bits

$3840000 / 8 = 480000$ bytes

$480000 / 1024 = 468.75$ KB

Example 2

Calculate the file size of a 300 x 200 graphic with 256 colours.

256 colours means a colour depth of 8 (because $2^8 = 256$)

$300 \times 200 = 60000$ pixels in the image

$60000 \times 8 = 480000$ bits

$480000 / 8 = 60000$ bytes

$60000 / 1024 = 58.6$ KB

Example 3

Calculate the file size of a 6' x 5' image, scanned at 200dpi, with 16 bit colour.

$6 \times 5 \times 200 \times 200 = 1200000$ pixels in image

$1200000 \times 16 = 19200000$ bits

$19200000 / 8 = 2400000$ bytes

$2400000 / 1024 = 2343.75$ KB

$2343.75 / 1024 = 2.3$ MB

Video formats

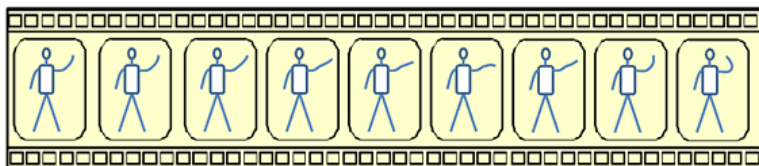
MPEG (.mp4)	Lossy compression	<ul style="list-style-type: none"> The level of compression can be determined by the user. Key frames (individual still images) are compressed and stored as JPEGs – generally only one full frame in every 8 to 24 frames needs to be stored (interframe compression). Any data that stays the same beyond the key frame does not need to be stored. Changes to the image in each subsequent frame are then stored.
AVI (.avi)	No compression built-in	<ul style="list-style-type: none"> Limited to 320 x 240 resolution Limited to 30 frames per second Can take up a large amount of storage space and can be quite RAM intensive.

Video compression techniques

MPEG video files often make use of intraframe and interframe compression.

Intraframe compression

Intraframe compression is simply the process of compressing each individual image (frame) in the video. In the case of MPEG videos, each individual frame is compressed as a JPEG using DCT encoding. These frames are then known as i-frames.



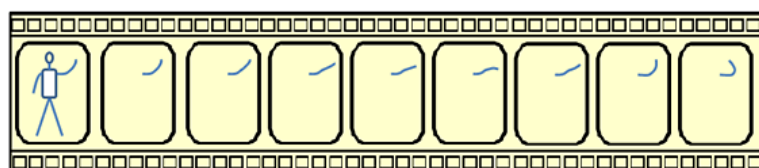
Intraframe compression

Every frame is encoded individually

Interframe compression

Interframe compression is the process of only saving changes between i-frames. This is achieved by using p-frames and b-frames. A p-frame only stores the changes that exist between the previous i-frame and the current i-frame. It does this by disregarding any data that has not changed.

A b-frame attempts to achieve greater compression by storing changes from both the previous and next i-frames, again by disregarding any data that does not change, either from the previous frame or in the next frame.



Interframe compression

Only the differences between frames are encoded
for each group of frames

Advantages of interframe compression

Interframe compression is most effective when data in the video remains the same between frames. For example when a news reporter is being filmed then there may be no or only small changes in the background, so storing only the changes between frames can significantly reduce the data that needs to be stored.

However in the case of footage of a sport, where there is lots of movement and panning of the camera, then intraframe compression may be more efficient as every frame will be different and the benefits for interframe compression are lost.

Video file size calculations

To calculate the file size (in bits) of an uncompressed video file the basic formula is:

number of pixels in frame x bit depth x frame rate (fps) x length of video (in seconds)

A video is simply a series of bitmap images, so this rule simply multiplies the size of one image by the number of images in the video.

Example

Calculate the file size of a 300 x 250 pixel video, using 16 bit colour, 30 frames per second and lasting 30 seconds.

$300 \times 250 = 75000$ pixels

$75000 \times 16 = 1200000$ bits for 1 image

$1200000 \times 30 = 36000000$ bits for 1 second of video

$36000000 \times 30 = 1080000000$ bits for entire video

$/8 = 135000000$ bytes

$/1024 = 131835.94$ KB

$/1024 = 128.75$ MB