ROSE BRUFORD COLLEGE
SOUND & IMAGE DESIGN YEAR 2

Digital Video Unit

Video compression for CD-ROM and the web

By Alex Etherington-Smith

9th October 2000 Semester 1

For (Tutor) Matt Ottewill

Contents

Introduction

Whether delivering video on CD-ROM or the Web, compression is a very important consideration. The need for compression arises when the data transfer rate of an architecture (e.g. an internet server) falls short of the required transfer rate. In other words, if a movie requires its data to be transferred at a rate of 800Kbps (kilobytes per second), a 4x CD drive (600kbps) will have trouble playing back the movie. The size of the movie needs to be reduced for the CD drive to cope with playback. Compression is commonly applied to all types of digital data (images, sound etc) but is required more often with motion video, due to its excessively large data rates.

For instance, a single frame of full-motion video (as defined by NTSC, North America's broadcast standard) has a frame size of 720x486 pixels. This takes up around 900 kB of data. NTSC operates at 30 fps (frames per second), therefore if 900 is multiplied by 30, we get a figure of around 27 (MB). This means that broadcast-quality video requires a data rate of 27MB per second. This is far too high a rate for a desktop computer to cope with, therefore full-motion video needs to be heavily compressed to reach a wider audience via CD-ROM or the Web.

Even at quarter screen size of 320x240 (common in multimedia titles) the data rate exceeds 5MB/s. If this were uncompressed a CD-ROM could only hold around 2 minutes of video. However, after being compressed to 100 KBps (eg with the Sorenson codec) the CD could hold over 100 minutes. To get the same 100 minutes onto a CD uncompressed, the frame size would have to be a tiny 40x30 pixels

A movie can always be compressed, yet, depending on the type of compression used, there will often be a loss of quality. Reducing the size of a movie is a very subjective process because a developer has to decide which data to throw away. The most obvious way is simply to reduce the frame size of the movie , yet this will lessen its impact upon the viewer. Alernatively, the frame rate could be reduced, but this would make the movie jitter. Regardless of the delivery method of a movie, the most important thing is to understand fundamentally how compression works. This will help a video developer decide which apects of a movie can most easily be reduced in file size with a minimum loss of quality.

In this report (which accompanies a class seminar presented by myself) I will first look at some basic compression terminology and compression techniques (known as algorithms). I will then discuss the need to consider compression at all times during the process of making a movie, from shooting to editing. Finally I will evaluate some of the most popular codecs and some of their features and uses.

Compression terminology

It is helpful to be familiar with some of terms most commonly referred to when discussing compression.

Delta:

The part of an image which changes from image to image is known as the *delta*.

Compression ratio:

The compression ratio represents the size of the original image divided by the size of the compressed image. This tells us how much the data are actually compressed.

Image quality:

Compression is described as either *lossy* or *lossless*. Lossy schemes ignore picture information which the viewer may not miss while lossless schemes faithfully preserve the original data.A lossy scheme permanently removes data which cannot be retrieved after decompression. The image quality decreases as more information is discarded. Lossy schemes reduce the data size more effectively but only work with video data with smooth transitions.

Compression coding

There are various lossless compression techniques which can be applied to video or any other type of digital data. Listed below are some of the most common:

Run length encoding

Run length encoding is a technique which replaces consecutive occurrences of symbols with the symbol followed by the number of types the symbol occurs consecutively. For instance the string

111110000003355

would be expressed as

15063252

In this example a string of 15 symbols has been reduced to only eight symbols. This technique is most useful where data are very similar and long lines of consecutive numbers occur. This would apply to pictures with similar areas of colour where pixels share the same value.

Huffman Coding

Huffman coding is a technique which assigns variable length codes to symbols so that the most frequently occurring symbols have the shortest codes. When decompressed, the symbols are reassigned their original fixed length codes. In the case of text, variable length codes are used instead of ASCII codes. The most common characters, normally space, e and t are assigned the shortest codes. Huffman coding is most effective where the data are made up of a only a small number of symbols.

Lempel-Ziv Coding

Lempel-Ziv coding uses a dictionary of symbol sequences. This means that whenever a particular sequence occurs it is represented by its position in the dictionary. The algorithm finds the most commonly occurring sequences and expresses them as a dictionary word. For example a hypothetical dictionary table might look like this

0	HE IS A BIG
1	(space)
2	GER
3	MAN
4	THAN
5	NOR

Here the text string of ‘HE IS A BIG’ is represented by the symbol ‘1’ etc.

This would mean that the sentence ‘HE IS A BIGGER GERMAN THAN NORMAN ‘ would be reduced to the following

0 2 1 2 3 1 4 1 5 3

Dictionary compression works best with structured data with repeating patterns.

Interframe and intraframe compression

Interframe compression is compression applied to a sequence of frames whereas intraframe compression deals with only a single frame/image at a time.

Interframe

There is often relatively little change between frames in video, especially when the picture does not contain much motion. The similarity between consecutive frames is known as *temporal redundancy* and is exploited by the compression to reduce the volume of data. Interframe compression describes a sequence of frames by reusing parts of preexisting frames to construct the new frames. Most interframe techniques work in this way and following are two of the most common.

1. Sub-sampling

Sub-sampling only transmits a certain percentage of the frames. Sub-sampled video might for instance contain only every second frame (50%). This means that either the decoder or the viewer's brain would have to interpolate(guess) the missing frames on reception. The Sorenson codec uses sub-sampling in its feature *temporal scalability* which allows a slower computer to skip through a percentage of the frames.

2. Difference coding

Difference coding is a simple technique which compares each frame with its predecessor and only updates pixels that have changed.

There are also more complicated techniques such as *block based difference coding* and *block based motion compensation* which divide similar areas of pixels into blocks whose positions in an image are then described by a vector, known as the motion vector.

Intraframe

Intraframe compression also employs sub-sampling to reduce the data in a single image. With both inter- and intraframe compression, sub-sampling requires the viewer to make up the missing(discarded) pixels inbetween adjacent pixels. This 'guesswork' is known as *interpolation*, and is applied by a viewer's subconscious brain when focusing on an image. It is also used by some codecs and can make a sub-sampled image appear to have a higher resolution than it actually has.

Intraframe compression also uses a technique known as *quanization* to describe pixels. Instead of reducing the number of pixels as does sub-sampling, it instead reduces the number of bits used to describe each pixel. Quantization where the size of the range is small is known as *coarse quantization.* The three images below have different numbers of colours. The first has thousands of different coloured pixels while the middle image has 64 and the rightmost only 8.

Vector quantization

Vector quantization is a more complex form of quantization which first dides the incoming data stream into a series of blocks. A set of patterns for the blocks already exists in a pre-defined table and each block is coded using the most similar pattern from the table. If the blocks are very small (few levels of quantization), the compression will be lossy.

Reducing frame size, frame rate and image quality

There are three ways to reduce the size of individual video frames and ultimately achieve a smaller file size for a movie. You can lower the frame size, frame rate or reduce the quality of the separate images. There is also a fourth technique called keyframing which compares and compresses multiple frames over a period of time.

Reducing the frame size alone will not affect the quality of the images displayed but will of course lower the visual impact of the movie. A frame size of 720x486 (NTSC standard size) is very large and the average movie of this size might take a desktop computer the best part of a day to download. A similar computer would also have difficulty playing such a movie file smoothly. This is why most movies found on the web have a considerably smaller frame size.

A reduction in the frame rate will also not affect the quality of the images but will add jitter to the motion. A smooth animation will often stutter and there will be a visible flicker . This effect in a movie is often less appealing than a poorer quality picture.

Considering compression when shooting video

Video images are made up of thousands of tiny, coloured individual pixels which constantly change over time to create the illusion that the images are in fact in motion. Thus motion video can only ever be conisdered a representation of the scenes captured on it.

The appearance of motion video images is dependent upon various factors including range and spread of colour and camera motion. Compression codecs do not respond to all movies in the same way and there are certain video images which they find difficult to compress.

Complex textures

Complex textures like leaves, grass or running water which all contain fine detail are challening to spacial compression. Furthermore, the tiny detail change between consecutive frames adversely affects temporal compression. This is an important consideration when choosing shooting locations and surrounding backgrounds.

Zooms and pans

Zooms are pans should be used sparingly when shooting because they cause every pixel in every frame to change. However, codecs such as MPEg and Sorenson are able to minimize the problem using a technique called ‘motion compensation’.

Moving objects

Moving objects have the same effect as zooms and pans in that they change every pixel in every frame and create more keyframes than stationary objects. A scene containing relatively little motion, such as a talking head in front of a still background will give a codec less trouble than a scene featuring for example an energetic dance scene. Movie files featuring rapid movement will experience jitter and blurring problems the more they are compressed.

Considering compression when editing

When editing your aim should be to avoid making the video difficult to compress. Following are some points to consider when choosing editing techniques:

MTV-style music videos which contain rapid scene changes force many more keyframes

Fade and dissolve transitions are the hardest on codecs as every pixel in every frame is changing over time. These types of transitions should be used both sparingly and briefly.

Hard cuts are the opposite of fades and dissolves as the codec is given the minimum work to do when calculating the pixels to draw from one scene to the next.

Codecs

Codec stands for coder/decoder, and it is used to both compress and decompress video data. For a user to play back a compressed video, they require the same codec as the one which created it. There are newer, more efficient types of codec being developed all the time but multimedia developers need to consider which codecs their intended users have at their disposal when choosing how to compress their movies. A movie compressed with the latest and best codec may end up being viewed by relatively few simply because the codec has not yet caught on with many internet and CD-ROM users.

It is also important to know your audience's target bandwidth, browser plug-in or CD-ROM speed, and likely CPU because it will give you more control over their viewing experience. If a CD-ROM or web site is aimed at the general public you may have to cater to the lowest common denominator and thus calculate the minimum requirements for each system (eg 4x CD-ROM drive, 133 mHZ clock speed). However if you are designing an intranet web site which can only be accessed by users on a particular corporate LAN, then it is likely you will be able to choose a codec and compression settings which best suit their CPU's.

Following is a brief evaluation of some of the most common codecs:

Quicktime

Apple's Quicktime is the most popular choice among title developers because it is one of the most versatile applications and includes a wide range of codecs which are regularly added to. It is also free to download from Apple's website. This means that a developer can expect a user to have a copy of it installed on their system. CD-ROMs often interrogate the host computer for Quicktime software and if none or an outdated version is installed, the user will be asked to install the latest version contained on the disc. Quicktime is also cross-platform meaning that it plays on both Windows and Mac OS systems while supporting a wide range of media including: video, audio, stills, text, MIDI and 3D graphics.

Cinepak

Sice it was first introduced about eight years ago, Cinepak has become perhaps the most widely used video codec around today. It is available for almost every platform and architecture due to the fact that it has low CPU requirements. It functions well on as low a specification as a 486 mHZ speed processor and some movies may even be able to play back smoothly on a 386. Cinepak will work with a 2x CD-ROM, a frame size of 320x240, frame rate of 15 fps and a data transfer rate of 180 kBytes/sec (2x speed theoretically delivers data at approximately 300kB /sec).

The disadvantage with Cinepak is the rather poor image quality. Images appear pixelated and colour saturation is often degraded.

Indeo

The 3x version of Indeo plays on low-end systems and

has similar image quality to Cinepak. While it produces more colour shift it can be better for movies containing relatively little movement. Versions 4x and 5x are known as 'Indeo Interactive' and has similar image quality and CPU requirements to MPEG.

Sorenson

Sorenson has low-end system performance similar to Cinepak (both play on a 486) yet features far better image quality. Sorenson works well both on the Web and CD-ROM. Sorenson includes a feature called temporal scalability which allows slower computers to skip through a certain percentage of the frames. Thus a 486 might play a movie of 30 fps at 15 fps. Sorenson can yield a high compression ratio on many video clips. For instance a 'talking head' at 320x240x15fps and compressed with Sorenson normally plays back fine at 20 kB/sec. A CD-ROM would be able to hold over nine hours of movie clips with the above specifications.
Sorenson Video 2.1 provides a feature called variable bit rate which adjusts the data rate according to how much action and how many scene changes there are in a movie. This feature allows for better video at lower bitrates. Using variable bit rate means that each scene is effectively encoded as if it were a separate movie.

Additional delivery considerations

For an end user to view motion video as the developer intended, the end user will be required to have a list of minimum requirements, as defined by the developer. Various issues concerning the end user's system need to be borne in mind by the developer during authoring including:

Available RAM
CPU speed
CD-ROM drive speed
Quicktime version
Installed browser and browser pug-ins
Internet connection speed (ie 14.4, 56k etc. modem)
Sound capabilities

Conclusion

The need to compress of all types of digital data seems to be an issue which multimedia developers will have to consider for some years to come. While data transfer rates are improving all the time, computer sytems will still have to work with relatively large quantities of data. In other words, no matter how fast an internet connection, we will always want a faster one: no matter how large a hard drive, we will always want a larger one. Digital video developers must realise that their movies will only reach a wider audience if compression is applied and file sizes are reduced. The subjective process of choosing the most suitable codec and compression techniques comes with experience.Developers should experiment with as wide a range as possible so that their compressed movies will look as good and function as closely to their uncompressed counterparts as possible.

Bibliography

Books

Vaughan, T. "Multimedia:Making It Work" California, USA: Osborne/McGraw/Hill, 1998.

Internet resources

Braunling, O. "Data Compression" at http:/ www.members.aol.com/breunling/obcompr.htm

Manning, C. "Interframe Compression Techniques" at http:/www.newmediarepublic.com/vide/compression/adv07.html

Baumgaertner, M. "Shooting Video For The Web" at http:/www.digitalchicago.com/Mag/JA99/Webvideo/squeeze.html

Author unknown Sorenson video

Giles, D. "Optimizing Video For CD-ROM" at: http:/ www. terran.com/ Codeccentral/Articles/OptimizeVideoCD.html - 1997