video file basics for audio folks

by dave linnenbank

18 july 2008

introduction and terminology

Working with digital video files requires a basic understanding of modern multimedia principles. This document aims to provide a rough outline of these fundamental topics and offer some general suggestions for common usage.

To identify the key issues associated with digital video, let's think of a movie as what it really is: a series of still pictures (or **frames**, in the movie context) shown in succession.

One parameter is the **frame size** of the pictures being displayed (sometimes called **image size**). The size is indicated with two numbers, the first representing width and the second height (e.g., *640 x 480*). These values are expressed in **pixels**, which can be thought of as the number of individual samples that form the image.

Another parameter is the speed at which the frames are being played, called the **frame rate**. Frame rate is regularly shown as the number of individual frames displayed in one second of time (e.g., *24 frames per second* [sometimes abbreviated as *fps*]). This expression of time is what differentiates movies from still image files.

A final, crucial parameter – and one that is somewhat less obvious – is what compression scheme is being used to encode the visual component. The choice of video **codec** (compression/decompression scheme) is critical and rarely well understood.

These three essential parameters are the variables that can be adjusted to meet our needs. The balancing act here – particularly in codec selection – involves how much processing power it will take to play back your video file (we'll call this *complexity*), how big your file will be (*file size*), and how good your video will look (*quality*). Knowing the purpose of your file – and, in some cases, the eventual distribution medium – will often help decide these settings for you.

With these general concepts in mind, let's visit the three key parameters in more depth.

frame size

Something to keep in mind about frame size is that it is a two-dimensional count. The ratio between the picture's width and height is known as the file's **aspect ratio**. Our earlier example mentioned a movie with a frame size of *640 x 480* (again, measured in pixels). The aspect ratio of this frame size would be *4:3*. Even if you choose to manipulate the frame size, you will in most cases want to preserve the aspect ratio of your original file, preventing it from distorting the image.

[We are only discussing the *file's* aspect ratio, which is set by manipulating the frame size. Various letterbox-type presentations may make the aspect ratio of the *image* different from that of the file. We will leave manipulation of the image's aspect ratio to the video's creator.]

Frame size is plainly a measure of how much information you are trying to represent within each frame. As such, this setting has a fairly direct relationship with your overall file size (how much disk memory your movie file takes up). The two-dimensional quality of this value is significant and often overlooked. If you halve most audio parameters – such as bit depth or sample rate – you end up halving the amount of data to be represented and reduce your file size by about half. Since frame size is made up of two values, however, you would be dividing both numbers by two and equivalently quartering the data to be represented in your file. Put another way…

640 x 480 = 307,200 total pixels
320 x 240 = 76,800 total pixels

For a presentation, I would try to keep the frame width to at least 480 pixels. You can always reduce the original image size to shrink file size, and then get your video player to stretch the image to a larger size for playback. View your file this way to make sure the picture doesn't appear overly blocky (or **pixelated**). This will always happen some when stretching the image to make it larger, but don't worry too much; the audience won't be sitting as close to the screen as you are to the monitor.

For a reduced bandwidth presentation such as the internet or a live performance where the video will be used as source material, I would at least halve the frame size. As suggested above, many platforms allow you to present your movie file at double the image size. This is sufficient for most scenarios.

For a work print version, I would also suggest halving the frame size. Image quality is not a priority while working on audio, and this will free up your computer's processor for audio-related tasks.

frame rate

Frame rate is a direct representation of how many individual video frames will be played in the span of one second. Frame rate – somewhat like frame size – corresponds with both the amount of information you are encoding and the resulting file size of your movie.

For a presentation, I would generally keep the frame rate at its original setting. Most NTSC video sources operate at *29.97 fps* (frames per second). If the video itself is computer generated, this may be the best frame rate to use. In many cases – especially where the clip is slow moving – reducing the file to *24 fps* will have little noticeable effect. (Video originally shot on film probably started at *25* or *24 fps* anyway.)

For a reduced bandwidth presentation such as the internet, I would consider exporting my clip at *24 fps* or even *15* or *12 fps* if the result isn't too distracting. Assuming you start with a clip at *30* or *29.97 fps*, going to 24 fps would reduce the amount of information by about 20%, with *15 fps* saving closer to 50%.

For a live performance where the video will be used as source material, I would suggest halving the original frame rate. Whenever video processing is involved, having too much data per second can often cause more harm than good. Processing can also potentially provide additional data on output between the occurrence of new frames.

For a work print version, I would suggest keeping the file at its original frame rate. This is the best practice when sync points may be involved.

codec

The choice of video codec controls how the visual element will be transcribed in your file. Accordingly, it is the single decision that affects your file most.

While current computers can easily handle 'uncompressed' audio files, video files must be compressed for proper playback. The main reason for this is an issue of size; while the CD audio standard (44.1 kHz, 16-bit, stereo) requires about 10 megabytes per minute, an old video standard (*640x480*, *29.97 fps*) requires about 30 megabytes per second when uncompressed. The computer needs to reduce the amount of data so it can play the file back without freezing up or excessively dropping frames.

Different needs require different methods of video compression, and many codecs have been developed over the years to address the issue of data representation. They each do their job in slightly different ways, but as an end user, you will never investigate the inner workings of individual codecs. However, we can speak generally about the two main premises in video data reduction.

All codecs utilize **spatial compression** techniques. This technique attempts to describe larger areas of the image instead of individual pixel values. By identifying patterns or repetitions in a single frame, the amount of data used can be greatly reduced.

Some codecs also use **temporal compression** schemes. Only certain frames are stored using spatial techniques, usually at a user specified interval (i.e., every 4th frame). These frames record the entire image and are called **key frames**. (The presence of a *key frames* setting indicates that you have selected a temporal codec.) For every other frame, the codec identifies what is common from the previous frame and records only the parts of the image that have changed. This significantly reduces the amount of data.

This is where the double-edged sword of compression appears. Just as a processor has problems handling too much data, it is also computationally expensive to playback a video with high complexity. Think of a codec as a way of reducing the video signal to a set of instructions on how to recreate each image. While a very light compression can cause the computer to deal with an enormous file, a highly complex codec creates a small file, requiring an enormous amount of calculations to turn it back into a whole image on playback.

The one setting you tend to get with every codec is amorphously and appropriately labeled *Quality*. Try different settings and realize that this parameter is not necessarily on a linear – or useful – scale.

For a live or reduced bandwidth presentation such as the internet, I would probably use a temporal codec since the computer processor will not be needed for other major tasks and the file size will be smaller in the end.

For a work print version or a live performance where the video will be used as source material, I would definitely use a non-temporal codec that has a medium amount of compression. This usually results in a larger file, but the processing power needed for decompressing the video is kept to a minimum, allowing more cycles for audio processes or further video manipulations.

closing thoughts

The best thing you can do is experiment with different settings. While codecs may exhibit a remarkable difference between *Medium* and *High* quality settings, some practically ignore this parameter altogether. Some changes in frame rate or frame size will seem small but make all the difference in the world between a usable file size – or load time – and one that is abysmal. Exporting your movie file with different settings is the best way to find the right ones.

The smartest thing you can do is test your movie in the context in which it will be presented. While a laptop computer may butcher your file, a desktop machine may handle it with ease. The only way to know is to know ahead of time.

One unmentioned factor that will affect the size of your file is the audio settings you choose. For a lower bandwidth file, audio compression can help file size and be fairly transparent (at least for accompanying video). For a live presentation, uncompressed stereo audio probably won't make a huge dent so err on the side of less compression.

And as always, test your file to ensure settings work on their own and play together nicely. Most programs will tell you the file's **data rate** – the average amount of data used to represent one second of time. This offers a good basis for comparison between files with differing frame rates, sizes, codecs, or even lengths. A number too large can cause poor performance. A number too small may indicate excessive compression (a high level of complexity), either compromising the video quality or overtasking the computer. You will decide and make appropriate adjustments.

Good luck.