|  |  |  |  |
| --- | --- | --- | --- |
| For a more simple overview see the [DRI data flow page](http://dri.sourceforge.net/doc/dri_data_flow.html).   |  |  | | --- | --- | | For greater detail see the [DRI control flow page](http://dri.sourceforge.net/doc/dri_control_flow.html).   |  | | --- | | 1. VAAPI 2. **Introduction**   The main motivation for VA-API (Video Acceleration API) is to enable hardware accelerated video decode/encode at various entry-points (VLD, IDCT, Motion Compensation etc.) for the prevailing coding standards today (MPEG-2, MPEG-4 ASP/H.263, MPEG-4 AVC/H.264, and VC-1/VMW3). Extending XvMC was considered, but due to its original design for MPEG-2 MotionComp only, it made more sense to design an interface from scratch that can fully expose the video decode capabilities in today's GPUs.  The current video decode/encode interface is window system independent, so that potentially it can be used with graphics sub-systems other than X. In a nutshell it is basically a scheme to pass various types of data buffers from the application to the GPU for decoding or encoding. Feedback on the API is greatly welcomed, as this is intended to be a community collaborative effort.   1. **Specification**   Version 0.31 of libva, an implementation of VA-API for Linux, is now available via git from the following location (http://cgit.freedesktop.org/libva/):  git clone git://anongit.freedesktop.org/git/libva   1. **Architecture**   Linux_vaAPI.gif   1. **Reference**   http://www.freedesktop.org/wiki/Software/vaapi   1. DRI data flow   http://dri.sourceforge.net/doc/images/dri_data_flow.png   1. Introduction   This overview diagram shows the different paths which the data takes from the program to the graphics hardware.  Starting at the left from the top.  Direct rendering program (3D):  Direct rendering (3D data) -> 3D data -> Graphics Hardware  Direct rendering program (2D):  X Protocol Encode -> Protocol Decode -> DIX -> XAA -> DDX Driver -> Graphics Hardware  3D and 2D commands share the same transport / protocol arrow / stream between the indirect rendering program (client) and the X Server. It's not until the protocol is decoded in the X Server that they diverge.  Indirect rendering program (2D):  X Protocol Encode -> Protocol Decode -> DIX -> XAA -> DDX Driver -> Graphics Hardware  Indirect rendering program (3D):  X Protocol Decode -> GLX -> Mesa (including SW rasterizer) -> DDX Driver -> Graphics Hardware  There is only one arrow at the bottom of the X Server where 2D data flows to the hardware. That's because the server is single threaded and handles one request at a time.  DRI Resource Management commands flow:  This level of detail is not shown in this diagram.   1. From   http://dri.sourceforge.net/doc/dri\_data\_flow.html | |  1. DRI control flow   http://dri.sourceforge.net/doc/images/dri_control_flow.png   1. Introduction   This low level diagram provide details not shown in the overview diagram.  Legend:  RM = Resource Management  or = 1 of these 2 paths are followed  & = both of these 2 paths are followed  2D = 2D commands & data  3D = 3D commands & data  lines in columns indicate individual paths while,  lines not in columns are aggregations of paths.  Starting at the top right.  3D and 2D commands share the same transport / protocol arrow / stream between the indirect rendering program (client) and the X Server. It's not until the protocol is decoded in the X Server that they diverge.  Indirect rendering program (3D):  X Protocol Decode -> GLX -> Mesa (including SW rasterizer) -> DDX Driver -> Graphics Hardware  Indirect rendering program (2D):  X Protocol Encode -> Protocol Decode -> DIX -> XAA -> DDX Driver -> Graphics Hardware  2D only program (2D):  X Protocol Encode -> Protocol Decode -> DIX -> XAA -> DDX Driver -> Graphics Hardware  RM and 2D commands share the same transport / protocol arrow / stream between the indirect rendering program (client) and the X Server. It's not until the protocol is decoded in the X Server that they diverge.  Direct rendering program (2D):  X Protocol Encode -> Protocol Decode -> DIX -> XAA -> DDX Driver -> Graphics Hardware  Direct rendering program (RM):  X Protocol Encode -> Protocol Decode -> DRI Module (-> DDX Driver) -> DRM Lib =>> SAREA & Kernel  Direct rendering program (3D):  Direct rendering (3D data) -> 3D data -> Graphics Hardware  Resource management:  SAREA & Kernel  3DDRP Internals:  Origins of 3D , RM , 2D & RM (x transport)  There is only one arrow at the bottom of the X Server where 2D data flows to the hardware.  That's because the DDX Driver is single threaded and handles one request at a time.   |  | | --- | |  | |

1. From

http://dri.sourceforge.net/doc/dri\_control\_flow.html

[**Digital Video Fundamentals - Frames & Framerates**](http://www.afterdawn.com/guides/archive/digital_video_fundamentals-frames_frame_rates.cfm)

At the heart of film and any kind of video is a series of still images displayed many times per second to give the illusion of motion. At its most basic, a frame is a single still picture that fills the display screen. This terminology is inherited from film to analog and digital video, and has different uses in all three.

The Digital Video Fundamentals Series is designed to give an overview of various video topics. This installment introduces progressive and interlaced frames, as well as framerates and a little about framerate conversions. Along with the other guides in the series, it should help prepare you to work with the digital video format of your choice. Once you understand the basic concepts involved, you can start making sense of whatever reference sources or guides you find, as well as hopefully learning to sort out the maze of misleading or flat out factually incorrect sources that are especially easy to find on the internet.

[**Progressive and Interlaced**](http://www.afterdawn.com/guides/archive/digital_video_fundamentals-frames_frame_rates.cfm)

****There are two basic formats for video, progressive and interlaced. Film is a progressive source because each picture fills the entire frame. That means the framerate is the number of individual pictures. Analog video, on the other hand, uses interlaced, or field based, video. Interlacing uses two different pictures to fill an entire frame. One field (picture) is drawn using every other line on a television screen, followed by a second field that's drawn on the lines skipped by the first. The fields are said to be interleaved together or interlaced. Like with film, each picture is from a different moment in time. Unlike film, the rate of individual pictures is twice the framerate.

While film only has progressive frames, and analog video only interlaced fields, digital video may have either or both. In the digital domain displaying progressively is a property separate from the contents of the frame. Although it's possible to encode and store interlaced video as a stream of fields, in reality this is rarely done. Most interlaced video is encoded in pairs in the form of frames. Although this is less efficient than true field based encoding, it's also simpler to design hardware for, making it a useful standard. As a rule, frame based interlaced encoding is less efficient than progressive encoding. This means that files with comparable bitrates will have lower quality if they contain interlaced video.

Field Conventions

When referring to interlaced fields there are different names given to each field in a pair. It's not important which convention you use, but it's a good idea to be consistent to avoid confusion. Sometimes the fields are referred to as even (lines 0,2,4,6,etc,...) and odd (1,3,5,7,etc,...), top (includes top line) and bottom, upper and lower, or even a (top) and b (bottom). I prefer even and odd personally because it conveys the interlaced nature of the fields better, but the most common convention is probably top and bottom.

Spatial and Temporal Order

The pictures that make up interlaced fields have a correct order relative to each other. In fact there is both a spatial order, meaning which should be the top or bottom field, as well as a temporal order, or which one represents an earlier moment in time. If either one or both of these is incorrect it can cause playback problems from jerky motion to blurred edges.

Field Dominance / Field Order

Either the top or bottom field of an interlaced frame will be displayed before the other. The field played first is considered the dominant field. This can be abbreviated to indicate which field (top or bottom) is first temporally. If the top field plays first the video is Top Field First, or TFF. This is also known as field order. Video with a dominant bottom field is considered Bottom Field First or BFF. Since field dominance is normally set by a flag of some kind for every video stream, it can generally be changed with no loss of quality to the video because no re-encoding is required.

Field Polarity

The fields that make up a single interlaced frame also have a spatial relationship to each other. The position of details in one frame is slightly different, either higher or lower, than the details in the other. This position is referred to as polarity, either top or bottom. This is important because in some cases fields can get reversed so that the top field is encoded as the lower field in a frame and the bottom field at the top. In such cases field dominance may or may not also be incorrect. The image on the left shows a portion of a frame with correct parity (ie fields in the correct spatial positions). Notice how much more detail you can make out in the man's face compared to the image on the right with reversed polarity.   


Fixing Field Dominance or Polarity Related Problems

If you have a video file with incorrect field dominance or reversed fields there are sometimes fixes available that won't result in any quality loss. But in some cases there's no way to get around re-encoding. In many cases field order problems are created during analog video capture. As long as you have access to some tool capable of changing video header information for the format you're capturing in you can generally change field order without altering the contents of the fields. For reversing field position you'll need to re-encode the video to fix it.

Deinterlacing

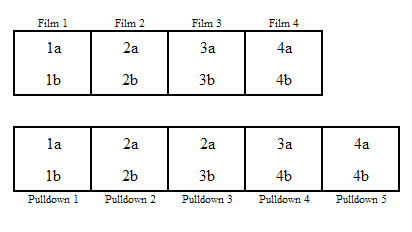
Sometimes it's necessary to deinterlace video for a variety of reasons. In some cases you may be encoding with a codec that doesn't support interlaced video (like MPEG-1). In others you may have a progressive display (like 720p) that you're planning to display your video on. Your options vary, depending on the editing tool you use and what's supported by your destination fomat and display technology. One quality that makes 720p HDTV technology easily adaptable for displaying interlaced content is an available framerate of up to 60fps. Since 720p standards allow twice the framerate of standard interlaced video (29.97i NTSC or 25i PAL) you can use a deinterlacing method called bobbing to avoid any loss of temporal information. Bobbing creates a progressive stream made up of individual fields in display order. Vertical resolution is cut in half because of this, and because of spatial differences objects appear to bob up and down from one frame to the next. At this point, either half the bobbed frames must be shifted down one line and then resized back to full resolution, or the missing lines from opposite fields need to be interpolated. Very few programs are capable of high quality interpolation on this scale, so it's still an area that's improving.

[**Framerate**](http://www.afterdawn.com/guides/archive/digital_video_fundamentals-frames_frame_rates.cfm)

The framerate for film or video is how many full screens are displayed per second. It's measured in fps (Frames per second). Even though analog video isn't progressive, for purposes of comparison it's useful to use the same units. Interlaced video does also have a fieldrate, which will always be twice the framerate. Film has a framerate of 24fps, PAL uses 25fps, and NTSC 30/1.001fps. The strange number for NTSC is because of changes that were made from the original 30fps black and white signal so color could be added. It's commonly noted as 29.97fps, or occasionally 30fps. Sometimes NTSC and PAL signals are identified by a combination of their framerate and type (interlaced or progressive) or fieldrate and type. 25i or 50i refers to PAL while 30i or 60i is generally used for NTSC. This is generally in reference to a digital signal with either NTSC or PAL characteristics. Film can be referred to as 24p.

Film to Video Conversions

In order to display material from film, like movies and many televisions shows, it's necessary to change the framerate to match either PAL or NTSC, depending on the television. For PAL this is generally accomplished by speeding it up approximately 4%. Each field is half of an original film frame. The audio pitch will be raised slightly in the process. Converting film to NTSC is a little more complicated since it's 25% faster than film. The solution here is to repeat fields. Since 24fps is 6 frames (12 fields) less than 30fps, repeating one out of every 4 fields will give you a 30i picture that plays at the same speed as the original. Reducing the speed to 24/1.001 and then duplicate 1 out of every 4 fields, will result in an exact NTSC framerate.

The process of duplicating fields to convert film to analog NTSC is called telecine, referring to the name of a machine that performs the conversion. Digital video can use field duplication for this conversion as well, or it can use a technique called pulldown. Pulldown flags tell the decoder that's reading frames to duplicate fields in a telecine pattern. There are two major advantages to this. File size is reduced because no fields have to be stored twice. Also, due to differences in compressing interlaced and progressive video, what is stored will be closer to the original film frames. This is called 2:3 pulldown, referring to the sequence of 2 fields displayed for one frame followed by 3 (2 original and a duplicate) from the next. Since only the playback framerate is important, when pulldown flags are added the framerate of the file is now 29.97fps.

Other Pulldown Patterns

Pulldown can be used for other things besides film to NTSC conversions. PAL's framerate of 25fps also works well as both a source and destination format for pulldown. Film only needs one more frame per second to reach it, so duplicating a single field out of every 24 (12 frames) would give you a 2:2:2:2:2:2:2:2:2:2:2:3 pulldown pattern and audio identical to the original. Progressive PAL can be converted to NTSC using 2:3:2:2:3 pulldown so that 5 frames become 12 fields. Since pulldown only works with progressive sources, if you want to use it on interlaced PAL you'll have to deinterlace first, but that's a subject for another guide.   
  
[**Digital Video Fundamentals - Resolution and Aspect Ratio**](http://www.afterdawn.com/guides/archive/digital_video_fundamentals-resolution_and_aspect_ratio.cfm)

It would be hard to find two more important aspects of digital video than resolution and aspect ratio. They're equally important for dealing with purely digital content and also understanding conversions from analog to digital and back again.   
  
The Digital Video Fundamentals Series is designed to give an overview of various video topics. This installment introduces the concepts of resolution and aspect ratio, including some comparisons between what the terms mean in both the analog and digital domains. Along with the other guides in the series, it should help prepare you to work with the digital video format of your choice. Once you understand the basic concepts involved, you can start making sense of whatever reference sources or guides you find, as well as hopefully learning to sort out the maze of misleading or flat out factually incorrect sources that are especially easy to find on the internet.

Since resolution means different things with regard to digital and analog video, the most basic thing you need to understand is what we mean by analog or digital video. You may have read or been told at some point that digital has something to do with electronics. In fact, nothing could be further from the truth. In reality, digital refers to something that can only have a limited number of states. For example, when I was growing up we had analog FM tuners. When we were trying to tune in a radio station at a frequency of 94.9 we might actually end up tuning to 94.83452 or 95.00002 because there was essentially an infinite variance. A modern tuner, on the other hand, will normally be digital. That means if you tune upward from 94.7 you'll automatically get 94.9. If you tune downward you'll get 94.5.

Digital Logic

To be more specific, digital video specifically refers to using digital logic. Digital logic involves problem solving using a special type of math known as boolean algebra, which changes all problems into math to represent them as 1s and 0s. Video is represented by a series of points in a grid, each of which is described by a binary (Base2) number of a fixed length. The dimensions of the grid are also fixed, and are what most people think of when they hear the word resolution.

Display Resolution

The width and height of the grid of pixels (points) used for each frame (full picture) of video are called display resolution. This is somewhat misleading, since there's no guarantee that all dots will be visible when the video is displayed. Display resolution is really the amount of detail that's stored (digitally) for each frame. This is purely a digital term, and while the resolution used impacts the detail displayed on your analog television, and is also a determining factor in how much detail can be captured from an analog source.

Visual Resolution

While only digital video has a display resolution, all video has a display resolution when played back. Display resolution is the amount of detail that can be distinguished by the human eye. How it's measured is dependent on the display format.

Line Pairs

The most basic type of analog source is film. Since this guide isn't designed to get into the level of detail someone who works with film transfers needs, I won't go into a great deal of detail. What is useful to understand is the concept of Line Pairs. In order to determine the maximum detail can be represented on film a series of black and white lines is used. The lines are divided into pairs - one black followed by one white - because the edges where each pair meets are where the details are. The visual resolution of film is measured in Line Pairs per millimeter (LP/mm), meaning the maximum number of Line Pairs that can be individually distinguished from a single millimeter of a frame.

Analog Video

Unlike film, in analog video (ie video in analog television format) there are discrete lines. This means that a television is made up of a specific number of (horizontal) lines that can be counted. The number of lines depends on whether the video is destined for a PAL or NTSC television, with 483 lines for NTSC or 576 for PAL. These are known as active lines, because a television signal also includes many more lines that aren't visible. Each active line is measured in the amount of time it takes to draw it from beginning to end, known as the active period.

Analog Waves

While digital pictures are made up of points, all analog signals are essentially waves. Waves have a period (how much time it takes to get from the beginning to the end) and frequency (how many waves there are over a period of time). Frequency is measured in Hz (Hertz) and 1Hz is equivalent to 1 complete wave per second. Period is the inverse of frequency, meaning a 100Hz wave has a period of .01s (.01 seconds).

TV Lines

In the case of television, each line is a series of waves, each of which can represent a pair of lines - one white and one black. This means two TV Lines are equivalent to one Line Pair. The number of TV Lines displayed on an analog television for a given signal is dependent primarily on the frequency of the analog source, which could theoretically be as high as 13.5MHz, although when considering digital sources you're generally limited by the limits of the television. You may find the number of TV Lines listed in your television's owner's manual. Remember to cut that number in half to determine the actual horizontal visual resolution.

Kell Factor

One affect of using discrete lines for video is that details may either fall between the lines or across multiple lines. Since each point (on a TV Line) can only represent a single color, details smaller than a line may also disappear completely regardless of position. Consider what would happen if you looked through horizontal blinds at a ladder leaning up against a wall. Depending on the dimensions of the ladder and how far away it was you might be able to see every rung, no rungs, or any amount in between. Scan lines have the same issue. This is represented by a number called the Kell Factor. The Kell Factor for analog television (on a CRT display) is assumed to be 0.7, meaning you can find the vertical visual resolution by multiplying the number of active lines by that number. The Kell factor is a combination of the underlying system (scan lines) and the limitations of display equipment. It actually varies above and below 0.7, but that's the generally accepted average for modern televisions. This may also be different for newer display technologies.

Sampling

when analog video is converted to digital, the process used is sampling. Sampling is the process of selecting a regular interval of time to determine the value of an analog wave. The samples are then saved, and can be used to reconstruct the waves when it must be returned to analog for display. In this case, the analog waves being sampled are Line Pairs (pairs of TV Lines). The basic requirements for sampling are laid out in the Nyquist-Shannon Sampling Theorem, including what's often referred to as the Nyquist frequency. The Nyquist frequency is twice the frequency of the wave being sampled. This is because at least two samples are required to accurately determine the shape of a wave. Unfortunately the actual technology used for digitizing video isn't accurate enough to perfectly duplicate the original wave, so a better target frequency for sampling would be as many as you can get.

ITU-R BT.601

The International Telecommunications Union has established the guidelines used as a starting point for sampling analog video. This standard, part of a document called ITU-R BT.601, specifies a samplerate of 13.5MHz for both PAL and NTSC. This document also specifies a display resolution of 720 pixels in width for both systems. As we'll discover later this doesn't necessarily mean what you think it means. The major difference is in the number of vertical pixels. PAL's 576 active lines translate to 576 vertical pixels in a digital frame. However, since NTSC's 483 active lines doesn't divide evenly into any multiple of 2 (this is very important in digital compression) it's reduced to 480 vertical pixels. Therefore PAL's full resolution is 720x576 and NTSC's is 720x480.

By applying the information from the previous section, you can make some comparisons of different formats. For analog formats, information relevant to the visual resolution will be listed. For digital formats the display resolution will be given. Remember, when watching digital video on an analog television you need to consider the number of TV Lines it can reproduce the hard limit on visual resolution. The numbers listed for minimum capture resolution are based on the Nyquist frequency. The actual resolution you should capture at depends on your hardware.

Digital to Analog Calculations

Maximum visual resolution for a given analog signal can be calculated based on its frequency compared to the amount of time each active line takes to draw on the screen. This is also known as the active period. For NTSC the active period is 52.6555µs (52 microseconds or .0000526555 seconds). For PAL it's 52µs. The ITU recommendation assumes a frequency of 53.333µs which is longer than either one. This means that not all of the 720 pixels in the ITU standard will be used for a picture. PAL, with an active period of 52µs uses the center 702 pixels (97.5% of 720). NTSC, with an active period of 52.6555µs uses 710 pixels (98.7% of 720). As a general rule, this calculation should be valid for digital sources designed to work with both PAL and NTSC video. Those designed specifically for one or the other, like DV, will generally have an active area that exactly matches either PAL or NTSC, meaning all pixels can be displayed in the active period of your television.

Broadcast PAL

**Frequency:** 5MHz   
**Max. TV Lines:** 520   
**Scan Lines:** 576

Broadcast NTSC

**Frequency:** 4.2MHz   
**Max. TV Lines:** 442   
**Scan Lines:** 483

VHS

**Frequency:** 3MHz   
**Max. PAL TV Lines:** 312   
**Max. NTSC TV Lines:** 316

S-VHS

**Frequency:** 5MHz   
**Max. PAL TV Lines:** 520   
**Max. NTSC TV Lines:** 526

VCD or DVD (SIF)

**PAL Resolution:** 352x288   
**NTSC Resolution:** 352x240   
**Active Period:** 52.148µs   
**Max. PAL TV Lines:** 350   
**Max. NTSC TV Lines:** 352

DVD

**PAL Resolution:** 352x576   
**NTSC Resolution:** 352x480   
**Active Period:** 52.148µs   
**Max. PAL TV Lines:** 350   
**Max. NTSC TV Lines:** 352

SVCD

**PAL Resolution:** 480x576   
**NTSC Resolution:** 480x480   
**Active Period:** 53.333µs   
**Max. PAL TV Lines:** 468   
**Max. NTSC TV Lines:** 474

DVD (Broadcast D1)

**PAL Resolution:** 704x576   
**NTSC Resolution:** 704x480   
**Active Period:** 52.148µs   
**Max. PAL TV Lines:** 702   
**Max. NTSC TV Lines:** 704

DVD (Full D1) or DV

**PAL Resolution:** 720x576   
**NTSC Resolution:** 720x480   
**Active Period:** 53.333µs   
**Max. PAL TV Lines:** 702   
**Max. NTSC TV Lines:** 710

The ratio of width to height of a displayed picture is called the aspect ratio or AR. AR can be listed as a ratio of integers like 4:3 (standard television) or 16:9 (widescreen television). More commonly they're listed as a ratio of something to 1 like 1.33:1 (4:3) or 1.78:1 (16:9). They can even be written with an implied 1 like 1.33 or 1.78. The last two notations are generally the most useful since they allow you to easily see the difference between multiple ratios.   
  
In the early days of film a standard aspect ratio of 1.37 was established. This was the standard AR for all films until television started becoming popular. In order to combat the perceived threat of the new technology, with an AR of 1.33, movie studios figured out how to change the AR of a film frame without requiring new projectors to display it. This new technology, anamorphic lenses, only required that a special lense be used that squeezed a wide image into the space normally occupied by a standard 1.33 image. This looks wrong on the film itself, but when projected through an anamorphic lense that matches the one used for filming it returns to the wide AR of the original.   
  
A standard television screen has an AR of 1.33, nearly the same as film's original 1.37 AR. These aspect ratios are considered fullscreen. Anything wider is considered widescreen. Other common aspect ratios include 1.66, 1.85, 2.21, and 2.35. Some movies, particularly epic movies released during the peak of widescreen experimentation in the 1950s, have aspect ratios as wide as 2.77.   
Since different sources with different aspect ratios may need to be displayed on the same screen (which will obviously keep the same AR all the time), it's necessary to change the video's AR or add borders to two sides to avoid distortion. Some conversions can be done by simply stretching or squeezing it a small amount to match a particular display. This is how fullscreen film is converted to fullscreen television (1.37 to 1.33) and usually how 1.85 widescreen is converted for viewing on a 1.78 widescreen television. Video with a significantly wider aspect ratio than the display they're intended for is generally letterboxed. This adds borders to the top and bottom.

Anamorphic Encoding

Possibly the biggest misunderstanding in digital video is how anamorphic encoding works. Since video is often encoded for viewing on both fullscreen and widescreen displays it's necessary to try to optimize for widescreen while maintaining the highest quality possible for fullscreen playback. This can be done using anamorphic encoding. Like anamorphic lenses for film, anamorphic video is stored squeezed into a narrower aspect ratio than the original image. The image can then be stretched to the correct AR when it's displayed on a widescreen television. If, on the other hand, it's being viewed on a fullscreen display, it can be squeezed vertically instead. Since the human eye is more sensitive to vertical resolution than horizontal, anamorphic widescreen will retain more quality (visual resolution) than simple letterboxing.

Open Matte

Another technique that can help prepare a source for display in more than one aspect ratio is filming with an open matte. A matte is simply a border used to change the AR of film. It's an alternative to using an anamorphic lense. It requires a wider filmstock to capture the same area as an anamorphic lense. Using an open matte means that the entire film frame is exposed, with the matte applied when a print is made on a standard fillm frame. The benefit of an open matte is that prints with different aspect ratios can be made, and in many cases they even allow fullscreen transfers by applying no matte at all. This gives a different picture than the original film since it likely includes elements that the director may not have even been aware of (since they weren't intended for the film) but it's better than the alternative of Pan and Scan.

Pan and Scan

There are a few different names used to refer to pan and scan transfers. Most of them are less than flattering. The problem with pan and scan is that it only shows part of an original film frame in order to keep the AR correct to fill the entire (1.33) television screen. This causes major problems in frames where important action occurs at opposite ends of the screen. In many cases a pan and scan transfer also includes some tricks that remove or distort the middle of the frame in order to get the sides both into view at the same time. The general rule of thumb is that a letterbox or open matte transfer is better than pan and scan, and anamorphic is better than either.

[**Digital Video Fundamentals - Color Formats**](http://www.afterdawn.com/guides/archive/digital_video_fundamentals_-_color_formats.cfm)

Color seems like a fairly straight forward concept. You probably started learning about color as a child, and may even have some understanding of different video connections like Composite, S-Video, Component cables. When you're working with digital video there are a lot more options available. Understanding those options and what they're used for may help you find your way through the maze of standards and formats.

The Digital Video Fundamentals Series is designed to give an overview of various video topics. This guide will cover the basics of what color is, how the human eye perceives it, and how it's represented in video; especially digital video. Along with the other guides in the series, it should help prepare you to work with the digital video format of your choice. Once you understand the basic concepts involved, you can start making sense of whatever reference sources or guides you find, as well as hopefully learning to sort out the maze of misleading or flat out factually incorrect sources that are especially easy to find on the internet.

Understanding how color is represented, or encoded, in digital video requires at least a rudimentary understanding of human perception and digital sampling. I'm going to gloss over the more technical information which is important if you're writing a program to understand how things really work, but is irrelevant for someone who just needs to understand what that program does. In some cases technically correct terms are used, and in others terms that are technically incorrect are used to reflect common useage in hobbyist community. In general these will be terms used interchangeably for analog and digital video, but which are really only correct for one or the other.

Color is essentially how the brain interprets different frequencies (wavelengths) of light that reach the eye. That light can either come directly from a light source, such as the sun or a light bulb, or as a reflection off of some other object. Light that contains equal amounts of every visible wavelength is white. When white light reaches an object, whether it's a painting, a car, or your skin, some frequencies will be absorbed and others will be reflected. The color of the object is equivalent to white light minus the absorbed frequencies. If all wavelengths of visible light are absorbed, no light is reflected and the object appears black. This is called subtractive color, and if you're like me you learned about this as a small child. Reflected light can be divided into three primary colors (colors that can be combined to make any other color) - Red, Yellow, and Blue.

When dealing with light coming directly from the source, or reflected from a white screen (meaning no visible wavelengths are absorbed) color isn't determined by subtracting, but rather by adding different frequencies together. Not surprisingly, this is called additive color. Like subtractive color, it can be divided into three primary colors, but instead of Red, Yellow, and Blue the primary additive colors are Red, Green, and Blue (RGB). This is how televisions, computer monitors, and film projectors all work.

Representing Color Digitally

Video that's stored in the standard uncompressed digital format must have a value of each primary additive color for each pixel. That is, each pixel is defined by three numbers; one representing Red, one representing Green, and another representing Blue. Like all representations of analog data in the digital domain, each one of these numbers is a sample. This means that there are a number of possible values for each one based on the size of number used to represent it, also known as bit depth (bits per sample). When applied to colors it's called color depth. Technically this means it's not uncompressed as a nearly infinite number of colors can be created from variations on these three colors, but given the limitations of the human eye it's possible to narrow them down to those the human eye and brain can distinguish from one another.

Bitrate vs. Bandwidth

If you know a little bit about digital video you should recognize the term bitrate, meaning how many bits are read per second. This is the correct way to measure a digital signal, but is completely meaningless in the analog domain. This doesn't mean there isn't a way to measure the "size" of an analog signal, but instead of a stream of bits it's a range of frequencies. This is the bandwidth of the signal. Every wave, whether it's made up of light, sound, electricity, or radio has a certain frequency (how many times a complete wave repeats per second) and one cycle per second equates to 1Hz (Hertz). Just as lowering the bitrate of digital bitstream would allow frames to pass more rapidly on a wire, reducing the bandwidth of analog video allows more signals to be included on the same cable or in the same satellite transmission.

Gamma Correction

Analog video uses a linear scale to represent Red, Green, and Blue values. This means that the difference between a value of 1 and 2 is the same as a difference between 128 and 129, which is also the same as the difference between 254 and 255. This is fine if you're simply measuring the light instead of representing a picture with it. Unfortunately, since the human eye can distinguish between smaller variations in wavelength at some frequencies than others, this representation either gives extra information that isn't perceptible by humans, or eliminates perceptible information to avoid the wasted storage or excessive transmission bandwidth. When this system was designed it wasn't much of an issue because any bandwidth concerns were overshadowed by the quality of technology available to reproduce the signal. In the digital world that extra bandwidth translates to bits, meaning we can either choose to waste bits for details we can't see or avoid wasted bits by sacrificing detail in the most important frequencies. In reality, digital video uses a third option called Gamma Correction.

Gamma correction fixes the problem by using a non-linear scale for each value. By devoting more bits to describe frequencies that the eye can perceive the most detail in, equal quality in all frequencies of visible light can be achieved without wasting precious storage space. In general, when you see what looks like an analog notation, such as RGB, with a single quote (or apostrophe) added afterward, it refers to gamma corrected digital video. For example, gamma corrected RGB uses the notation R'G'B'. This is normally only an important distinction if you work with both analog and digital signals, but is also important to understand if you're trying to understand a technically accurate text. For the purposes of this guide, unless analog video is specifically mentioned, you should assume that all values refer to gamma corrected digital video. Most of the information available on the internet intended for hobbyists uses analog analog notation to refer to digital video.

RGB

For our purposes we'll consider 24 bit RGB (R'G'B') to be uncompressed color. The number 24 refers to the total number of bpp (bits per pixel) used to describe color. A 24 bit number has a range of 0 (no color or black) to 16,777,215 for a total of 16,777,216 different colors. Since Red, Green, and Blue are each described by 8 bits (24/3), which gives a range of 0 (no Red, no Green, or No Blue) to 255 for a total of 256 possible variations of each primary color. From this you can start to get an idea of why using RGB color presents storage challenges. If you were to use the lowest resolution allowed for an NTSC DVD frame (352x240) with RGB24 color (the common name for 24 bit RGB) you'd end up with 352x240x24 = 2,027,520 bits or 253,440 Bytes for each frame. And that's without any of the additional information required for correct playback. At that bitrate, a single second of video is over 7 Megabytes, a minute is nearly 2/3 of a CD, and a full hour is over 25 Gigabytes - more than five DVD-9 discs. At the highest DVD resolution, which most DVDs are encoded at, it would be over 100GB per hour.

Color Space

The term color space refers to a method of describing color. RGB is one color space, but not the only one. Just as the human eye is more sensitive to certain wavelengths of light than others, it's also more sensitive to changes in luminosity (how bright something is), than chromaticity (changes in hue and saturation of each primary color). Therefore it makes sense to use a color space that takes that handles luminosity and chromaticity separately. The analog color space that does this is called YUV or YPbPr. Since both are technically analog terms, you may also see this referred to with the correct terminology of Y'U'V' or Y'CbCr. As the notation suggests, The difference is that YUV values are calculated from linear scaled analog RGB, while Y'U'V' values are figured from gamma corrected R'G'B'.   
  
**Note:** Even though most digital compression formats use YUV, that doesn't mean YUV itself is compressed. It's simply a more sophisticated model of human perception than RGB.

How YUV Works

The most important difference between RGB and YUV is separation of luminosity and chromaticity characteristics, commonly referred to as luminance and chrominance in the analog (ie linear scaled) domain. For digital video they're actually given the names luma and chroma to denote the use of Y', Cb, and Cr calculations vs. Y, Pb, and Pr. Since most luma is actually perceived in green light waves, and conversely most information the human eye perceives from the color green is luminance, green can be omitted from the chroma information completely. That leaves us with chroma components of only Red and Blue. Each of the three components may also be referred to as a channel, as in the Y (Luma) Channel, U (Blue) Channel, and V (Red) Channel.   
<H3COLORIMETRY< h3>

Regardless of what color space is used to store video, it always starts out and is displayed in RGB format. In order to make sure the RGB values used for display are as close as possible to the ones oroginally encoded to YUV, both the encoder and decoder need to use the same calculations. Those calculations are called colorimetry. Fortunately there are specific standards that cover this. For the most part, the only two standards you need to know about are Rec. 601 (aka ITU.601, BT.601, or SMPTE 170M) and Rec. 709 (aka ITU.709 or BT.709). Rec. 601 is used for MPEG-1, MPEG-4 ASP (DiVX, XViD, and the like), and DV. MPEG-2 may use Rec. 601, Rec. 709, or SMPTE 240M (almost the same as Rec. 709). HDTV and DVD video are always supposed to use Rec. 709. Just as with nearly everything related to digital video this isn't always as simple as it should be. Some HDTV signals and DVDs are encoded with Rec. 601, and sometimes the colorimetry even changes in the middle of a video stream. In the case of MPEG-2, although the colorimetry used is stored in the file (since it supports multiple standards) sometimes it's missing so Rec. 709 is assumed. Since there's no way to confirm this from the actual frame data this may or may not be a correct assumption.

Chroma Subsampling

The human eye is far less sensitive to changes in chrominance than to changes in luminance. Since YUV already has a separate luma channel the next logical step is to lower the resolution of the chroma samples. In other words, a single chroma sample can be used for multiple pixels. In fact, some YUV implementations sample chroma at 1/4 the resolution of luma. Since the bpp for chroma is 2/3 of the total bpp this can save a lot of space. DVD, for example, uses YV12, which has only one U and one V value per block of four pixels. This cuts the bitrate in half - from 24bpp to 12bpp. Each variation on chroma sampling is considered a separate color space, but they can all be described as being in the YUV color space.

Each YUV color space can be described with a series of three numbers, representing the relationship between the number of Y samples and the number of U and V samples. The primary YUV color spaces you're likely to see are 4:2:0, 4:2:2, and 4:1:1. While I've seen explanations for a supposed system that can be used to understand what these numbers mean, I can't verify the accuracy of these explanations, and in the end it's easier to simply memorize what each standard set of numbers means.

4:2:2

In 4:2:2 YUV each pair of chroma samples is shared across a pair of pixels horizontally adjacent to each other. The vertical luma and chroma resolution are identical. MPEG-2 and MPEG-4 both technically support 4:2:2 YUV, but I'm not aware of any encoder or standalone player format that takes advantage of this.

4:2:0

YUV with a chroma subsampling of 4:2:0 shares chroma samples across both horizontally adjacent and vertically adjacent pixels. Each 2x2 block of pixels contains only a single U/V chroma sample.

4:1:1

Much like 4:2:2, 4:1:1 YUV uses the same vertical resolution for chroma as luma. Unlike 4:2:2, the horizontal resolution is only 1/4 of the luma, meaning each group of four horizontally adjacent pixels shares a single piece of U/V information.

Sample Locations

Although it would be possible to simply repeat the same chroma information for each pixel that shares a pair of chroma samples, the quality is much better if the chroma is assigned to a specific point and then values between any two points (samples) is interpolated. Interpolation means simply taking two points and mathematically determining the value of the points in between. Depending on the algorithm involved, more points can be considered for greater accuracy. For example, consider three points in a line. If the first point has a red value of 0, the last has a red value of 255, and the middle a value of 127, you can be fairly certain that colors in between them are even graduations. In reality, the logic required to interpolate the additional pixels is much more complicated than that, but the basic idea is the same.

Packed vs. Planar

There are two ways for a YUV encoded picture to be stored in a file. The most common is using a packed format. This simply means that the luma and chroma samples are stored next to each other in the file. For example, a 4:2:2 encoded frame could have the folowing order:   
|U|Y|V|Y|U|Y|V|Y|U|Y|V|Y|U|Y|V|Y|   
Each chroma sample is located in the exact same coordinate of the frame as the corresponding Y sample. Each group of pixels that share a single chroma sample is called a macro pixel.

The alternative, planar format, changes the order so that all the Y information for a given frame is followed by all the U information for that frame, which is followed by all the V information for the frame. Think of it as starting with a surface that's backlit with a single white lamp for each pixel in a frame. Then a sheet of translucent blue material with the chroma samples and interpolated points in between is put over the top. Finally a translucent red sheet, with chroma samples and interpolated points, is put on top of the blue. When all three layers are in place you have your final picture.

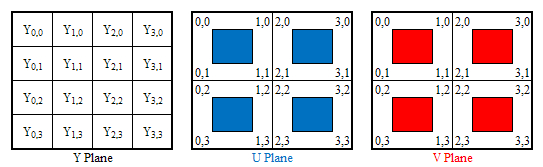
So why bother with planar format? It has one major advantage over packed formats. Since luma is never subsampled, each luma point is always at an exact pixel location. Chroma however doesn't necessarily need to be lined up perfectly with a single pixel. With a planar format pixels can be placed on a grid that centers them between pixels. For example, a 4:2:0 encoded frame could make use of this by putting the chroma samples in the center, where all four pixels a sample applies to meet.

Once you understand the basics, it's easiest to use more graphic representations of each one to understand them. I've also included a table at the end that details which color space is used for some standard video formats.

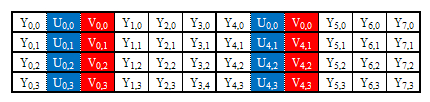
YUY2

YUY2 is a packed color format that uses 4:2:2 chroma subsampling. Each pixel has either a U or V sample, but not both.   


YV12

YV12 is a planar color format that uses 4:2:0 chroma subsampling. Each chroma sample is located at the nexus of 4 pixels. In the diagram below the chroma layers are shown relative to pixel positions.   


DV (NTSC)

NTSC DV doesn't have a name for its color format, but always uses the same packed 4:1:1 chroma subsampling. Some PAL DV formats (but not consumer grade DV equipment) also uses this subsampling. Notice that it uses the same number of chroma samples as YV12, but only subsamples horizontally, and locates both the U and V sample at the left most pixel in the group instead of centering like YV12 or locating the samples at different pixels like YUY2.   


DV (PAL)

PAL DV, like its NTSC cousin, uses a packed color format that doesn't have its own name. Unlike NTSC DV, the chroma is subsampled at 4:2:0, but since it's a packed format the chroma samples are aligned to a specific pixel.   


[**Digital Video Fundamentals - Lossy Video Compression**](http://www.afterdawn.com/guides/archive/digital_video_fundamentals_-_lossy_compression.cfm)

Due to the size required to store even compressed medium to high quality video, most formats a consumer will come into contact with use some form of lossy compression. MPEG-1, MPEG-2, MPEG-4 ASP, AVC, VC-1, and many other compressed formats are used for delivery to the consumer, and are also used by consumers to create original content or transcode existing content.

Possibly the biggest obstacle to making use of even the simplest guide is a clear understanding on the meaning of particular settings or instructions. Armed with just a little bit of knowledge about the purpose for what you're doing, you should be able to get better use from a wider array of reference sources.

The Digital Video Fundamentals Series is designed to give an overview of various video topics. This guide explores some basic concepts of lossy video encoding. It's recommended that you be familiar with the basics of digital video terms and concepts, as they'll be used here with little explanation. The rest of our Digital Video Fundamentals series of guides would be a good place to start:

General Concepts

Before trying to understand what's involved in lossy compression, we should start with a better description of what it is and how we decide when to use it, what kind of compression fits our needs, and what settings to use.

What Is It?

Whenever you see the word lossy you should assume that a transformation is taking place, and information will be lost in the process. Compression obviously means making a smaller file. While you can achieve a significant decrease in size using lossless compression techniques, a two hour movie would still take up around 35 DVDs unless some detail is approximated. Unless quality or editing requirements outweigh size restrictions it's almost always necessary to use lossy compression to drastically reduce size. With newer HD video it's more important than ever.

When To Use It

Obviously the purpose of lossy compression isn't to lower quality (which is what happens since it's lossy) so it must be compression. So the most basic answer is use it when you need to have a smaller file size. For example, if we consider 24 bit RGB to be uncompressed that would mean over 100GB for 1 hour of video (with no accompanying audio). At 4.37GB each, DVDs wouldn't store much. The other side effect of leaving video uncompressed is ridiculous transmission requirements. At over 200Mbps it couldn't be streamed over most home (or corporate) networks. It can also cause problems for storage hardware, which isn't designed to be read and written to as quickly as uncompressed video would require.

The primary reason for most people is the most obvious. Use it when your destination formats requires it. If you're making a DVD-Video disc you'll need to encode to MPEG-2. If you're recording with a miniDV camcorder you'll be encoding with DV or with a newer HDAVC camcorder you'll automatically get HDAVC.

What Kind Of Compression?

Once you've determined you need to compress video you need to decide what kind of compression is appropriate. The first consideration will obviously be destination format requirements. If you're encoding for a DVD you'll be using MPEG-2 because that's what a DVD player can play.

Assuming you need to choose between different lossy compression formats you'll want to think about what you're going to be doing with it. Whether it's interlaced, how much space it will take on a hard drive or other media you're fitting it to, whether you'll need to edit it later, and what software will need to read it are all legitimate considerations. You may find that a particular lossy compression format fits most of your needs, and use it almost exclusively or you may decide that each case should be decided independently.

Settings

It's impossible to make across the board generalizations about the best settings for even a single program, let alone all compression software. What you can say is that matching settings for your source when encoding and decoding will always give you the best results.

Encoding

Any time you create a digital representation of an image or series of images you're encoding. This applies to saving uncompressed video from an analog source in a digital format as well as encoding to XviD for an HTPC. Encoding just means it's stored in a code that some software or device will later be able to decode (read and interpret). Encoded video can be compressed, uncompressed, lossy or lossless.

Transcoding

Most lossy encoding is done to alter a file that's already been encoded in some digital format. That's the definition of transcoding. Transcoding starts with digital (ie encoded) video and changes the encoding. The most common examples of this are DVD backup programs that remove bits from video streams, and are commonly referred to as transcoders. However, since most video processing is already digitally encoded already, almost everything you do with it involves transcoding. This includes encoding MPEG-4 video to DVD and compressing DVD-9 backups to DVD+/-R or CD size in various formats.

Codecs

Codec stands for Compresssor/Decompressor and it means software or hardware that can compress (encode) and decompress (decode) a particular format.

Basic Compression

Two of the most basic video comression strategies actually come from still image compression; compressed colorspaces and DCT (Discrete Cosine Transformation). They're important not only because they're used so universally, but also because they work well for video that needs to be edited later.

Colorspace Compression

By taking advantage of the human eye's limited chroma (color hue and saturation) sensitivity compared to luminosity (brightness) we can remove a great deal of chroma information and approximate it when decoding. This requires that we use a YUV colorspace of some kind and then remove U and V information for half or more of the pixels. By eliminating half the chroma this way we reduce the space required to store two pixels from 48bits to 32 bits - a savings of 1/3 - while maintaining decent quality for editing. Reducing the chroma to one pixel out of every four (like DVD and NTSC DV) you'll end up reducing video size by 50%.

YUY2 and YV12

The two most important compressed YUV colorspaces are YUY2 and YV12. YUY2 is a good editing format because it keeps half the chroma detail while greatly reducing file size. It's especially good for interlaced video because the position of chroma pixels never crosses field boundaries. YUY2 is generally preferred for distribution and playback becuase of its 50% size reduction. It's not particularly good for editing, and should generally be avoided for video that may need more than simple editing in the future.

DCT

Discrete Cosine Transform, or DCT, is a mathemetical process I don't pretend to understand, that divides images into 8x8 blocks of pixels which can be compared to each other mathematically. The resulting values are then quantized, another process that's over my head, resulting in values that are approximations of the originals, but include more identical groups of blocls now. Just like the area those pixels represent, the less precise the detail is, and the more similar the 8x8 blocks are to other blocks.

RLE

Once DCT values have been quantized to form regular patterns a final process called Run Length Encoding (RLE) can take place. This finds instances where a sequence of bits is identical, like two identical 8x8 DCT blocks, and replaces redundant data with a shorter series of bits. If the blocks weren't quantized first this wouldn't normally do much because of how much variation can occur betweem blocks.

Three Dimensional Thinking

The biggest distinction between compressing individual images and a series of moving images is the introduction of a third dimension into the equation. Not only do pixels have similarities to those above and below them, but they can also be compared to pixels that occupy the same or adjoining space in preceding and following frames. In other words pixels close in time.

GOP

A Group of Pictures (GOP) is a group of sequential video frames, at least some of which rely on others in the group to be a complete picture. Each GOP starts with a complete image that can be drawn without decoding any other frame. To save space, any information that doesn't change in the next frame can be skipped, and only the changes recorded for the second frame in the GOP. The more a particular frame depends on the contents of other frames, the more potential for inaccurate data there is, but the more compression is achieved as well.

Open GOPs

GOPs may be completely self contained, and in some cases may have to be. They can also contain frames that require information from adjacent GOPs. These GOPs are considered opened. Closed, or self contained, GOPs are required by DVD-Video specs for multiple angles or seamless branching (ILVU).

Bidirectional Frames

Some comnpression formats allow frames to be built from frames both before and afterward. These bidirectional frames have the advantage of more compression, because so much information comes from other frames, without the errors caused by relying on only one frame. By using combinations of frames that only use information from one other frame but take more space with those that rely more on other frames but are more compressed, a compromise between size and accuracy is achieved that may allow us to use fewer full frames, also known as Key frames.

Drawbacks

While using a GOP structure will increase compression, it will also cause problems editing later on. If you want to cut a frame out of the middle of a GOP, frames on either side would be affected unless you cut on a key frame of a file with closed GOPs. This is why formats like the various DV variants, which are intended for editing, don't use GOP compression.

MPEG GOP Structure

Since it's so common, it's worth mentioning that MPEG-1 and MPEG-2 refer to key frames as I (Index) frames, frames that rely on one other frame as P (Predictive), and those that rely on frames before and after as B (Bidirectional).

How Much To Compress

Ultimately the picture quality you're left with after lossy compression depend on which of these techniques you use, and how. Formats for distribution, like DVD, typically use all of them because they're not intended for editing. On the other hand, DV was designed with editing in mind, so every frame is an I frame. This means you can do anything you want with that frame, even remove it completely, and the surrounding frames won't be affected except for later frames having a different time code.

Bitrate Considerations

Other than format requirements for standalone players, the most important consideration for most encoding is bitrate. Bitrate determines that maximum quality you can get from a given format, and may sometimes be the determing factor in what format is appropriate. Final video size will always be constant for a given bitrate, but quality will vary by format and usually by specific encoder software or hardware. The same bitrate that would be acceptable for one format (1000kbps at 720x480 MPEG-4) might look horrible in another (1000kbps at 720x480 MPEG-2).