Digital Video

We have two kinds:

- composite (D-2) digital video
- component digital video

Composite Digital Video

Composite video is analog video that has been digitized. Digitized the entire analog waveform for the entire time. Based sampling on four times the subcarrier frequency.

How do we make sure the burst runs continually? Keep ringing a crystal.

How do we multiply subcarrier frequency to four times? Phase locked loop. How does PLL multiply the frequency? Start out with VCO running at 14.318181 MHz. Put through a frequency divider and feed that back to the phase detector (and then the filter). Burst goes into the phase detector. Can use a pair of D flip flops, or can use a counter (output a pulse after every 4 pulses). The VCO will be phase locked to 3.579545 MHz.

The VCO is a sloppy oscillator, not precise like a crystal. The VCO is designed to be well-controlled through application of an external voltage, which compensates for the sloppiness of the oscillator.

Take composite digital video, start sampling at 4FSC. Within burst, will be sampling four times per burst cycle. If sampling at sync tip, what kind of digital number would you expect to have? A number on one of the extreme; would be fairly close to zero. We don't have any reserved numbers in D-2 format since there is nothing to reserve. D-2 gets decoded and result gets put out as NTSC analog video. If used highest number in sampling (for the highest IRE value). Bit depth may be 8, 10, or 12. In D-2, 8 is the standard. So, we'd have 11111111 as the highest number. Give some cushion at the high end, meaning don't use 11111111 to represent 100 IRE. Sync tip is at .287 volts. Can't just set this exactly, so might set this at .3. So, could be clipping off sync when digitizing, so need some cushion at the low end as well.

Local community cable TV productions will often show excessive whites as blacks.

Must encode burst also since we need the phase of the burst to properly interpret chrominance.

The 14.3181818 MHz becomes a square wave, and the sampling occurs on the upclock.

D-2 blindly, dumbly encodes/digitizes the NTSC analog signal. D-2 doesn't care what's in the NTSC signal, except that it synchronizes the phase of the sampling frequency to the phase of the burst/subcarrier. The burst/subcarrier stays in the same phase even though burst appears to flip around, but that's just due to the length of time for each line.

Component Digital Video (SDI, D-1, 601, SMPTE 272M)

Components are Y, R-Y, B-Y. Could do RGB also, but is a little different in terms of encoding. These are all analog since that is what the signal originates at; humans are analog, light is analog in the macro level, eyes are analog and ears are analog.

What causes the conversion of RGB to Y, R-Y, B-Y? A matrix of resistors. Y = .30R + .59G + .11B. Why do we take different amounts? It is roughly equivalent to the sensitivity of the human eye to those colors. Could get Y with one tube, but for color, must use multiple tubes all in the same phase.

How do we do addition of these three RGB? Use voltage dividers (resistor networks). If have .7R in one resistor and .3R in the other resistor, we have a voltage divider that outputs .3 of the input voltage. Similarly for G and B when tapping between the two resistors for the output and when the input is applied between the two resistors. How do we get a negative value? Inverting it through an amplifier. Use a common emitter amplifier which inverts the signal (collector signal 180 degrees out of phase with base signal). Adding them via a glob of solder (actually use transistors to isolate them from each other). Green has matrix of .41 on top and .59 on bottom. Blue has matrix of .89 on top and .11 on bottom. Use the transistor in a common emitter configuration to get -Y, then add the R or the B to get R-Y or B-Y.

All the resistors used in the matrix are highly precise, 1% or ½% or better.

Sampling rates. Y sampled at 13.5 MHz (compare with 4FSC at 14.318181 MHz). Sony pushed 13.5 MHz so that they could make just one piece of equipment to cover both NTSC and PAL (just change the power supply: 240V, 120V, or 100V). R-Y is not as important detail-wise as Y, so sample at half bandwidth (6.75 MHz). B-Y even less important for detail, but sample it at same rate as R-Y because it makes the math a lot simpler.

How to add these three components together? Have 10 bits from each component per sample. Go parallel to serial. Pros don't do overclocking; that's a consumer thing. Put the 10 parallel bits, put it into a shift register, take out the 10 bits serially. The rate of the parallel-to-serial conversion must run at 10 times the sampling rate (135 MHz or 67.5 MHz).

On every other clock pulse, sample Y, R-Y, and B-Y, and on the alternate clock pulses, just sample Y. That's how we get the funny dot pattern. Get R-Y, B-Y, Y, Y, R-Y, B-Y, Y, Y.

For a 13.5 + 6.75 + 6.75 = 27.0 MHz total sampling rate. Actual format is [Cb Y Cr] [Y] [Cb Y Cr] [Y]. The [Cb Y Cr] sampling of all three at once is called **co-siting**.

Collect 40 bits every two cycles, which has a period of 148 nanoseconds. So, the bit rate would be 40 bits / (148 * 10 ^ -9) seconds = 270 megabits / second.

Emergency frequencies are 121.5 and 243.0 MHz. Lower is commercial, higher is military. The next harmonic is satellite transponder signals. Must watch out for signal leakage in the station/studio since two harmonics of 13.5 MHz are these emergency frequencies.

Y, R-Y, B-Y came in handy so can do 4:2:2 and save ourselves some bandwidth. If did 4:4:4, our bandwidth would be 13.5 * 3 * 10 = 405 megabits per second. Compare with composite digital at 143 megabits per second.

Digital doesn't get converted back to analog until it reaches the display device (CRT, LCD, etc.).

MPEG2

Acronym stands for Motion Picture Experts Group. JPEG stands for Joint Photographers Experts Group.

The best lossless compression algorithms can only compress by a factor of 3 on average. Key word here is *lossless*. To do better, we have to have loss. We have to exploit irrelevancy in order to obtain the further 15-fold compression. Irrelevant because it cannot be seen by the human eye under certain reasonable viewing conditions.

The I Frame

I pronounced eye. We don't talk in frames really. We talk in terms of pictures here. This name should be the I picture. It means Intra Pictures. The I-frames are intra coded, they are coded and can be reconstructed without reference to other frames.

Reduction of spatial but not temporal redundancy. Spatial relates to a single picture, within the same picture. Temporal relates to among multiple pictures (like motion pictures). Not temporal here since don't want to reference other frames. The blue screen wall has little spatial information. The curtains have a lot of spatial information due to the folds.

The I frame **provides access points to where decoding can begin**. This is important. Sometimes called the **anchor picture** or **anchor frame**.

This is the most important frame. Other things are supplied by this frame to fill in what they are missing.

The P-Frames

Only three of these.

Predictive frames or pictures.

Can use previous I or P frame for motion compensation.

May be used as a reference for further prediction.

Each block can be predicted or **intra-coded**.

The block is the basic unit in MPEG. It is a block of pixels (two-dimensional block).

In the receiver, the P-frames are forward predicted from the last I-frame or P-frame. It is impossible to reconstruct P-frames without the data of another frame (I or P). This is temporal compression. P frames may compress temporally and spatially. By reducing spatial and temporal redundancy, we have increased compression over I pictures alone.

Coding of P-pictures

Motion-compensated prediction from an encoded I-picture or P-picture.

Half-pixel accuracy of motion compensation. This is true in standard definition or high definition.

One displacement vector per macroblock. We will learn about this later.

Differential coding of displacement vectors.

The B-Frames

Highest compression of all the three types of frames. Can look like a ghostlike skeleton on a black background.

Bidirectionally predictive pictures. Has so little information so that it has to look forward and backward. Requires that the future (subsequent) frames be in memory before can recreate these frames. Changing channels can take a few seconds because TV has to load all those pictures first, due to the way MPEG is structured.

Each block can be forward, backward or bidirectionally predicted or **intracoded** from the last/next I-frame or P-frame. Two other frames necessary to reconstruct a B frame. Must look forward and backward. Have to reduce 1.5 Gbits/second to 19 megabits per second; this is one way we do it.

P-frames and B-frames are referred to as **intercoded frames**.

These frames become important as things change (like during dissolve transitions). Football games are hard on MPEG. Camera panning across a crowd while following the game.

Coding of B-pictures

Motion-compensation prediction from two consecutive P- or I-pictures One of:

- only forward prediction (1 vector/macroblock)
- only backward prediction (1 vector/macroblock)
- average of forward and backward prediction = interpolation (2 vectors/macroblock)

Half-pixel accuracy of motion compensation. This is why we can do high definition because of this good level of motion compensation.

Backward Prediction from a Future Frame?

The encoder must reorder the pictures. Any picture the B frame references must be sent first. This means the future pictures must arrive before the B frame. This is why it takes so long to display pictures after changing channels.

(In heavy action pictures, get I frames much more often than once per second.)

Introduces a reordering delay related to the number of consecutive B pictures. Could send 5 or 8 B pictures in a row. The future picture used by the B pictures must be sent before any of those B pictures.

Watch an analog and digital versions of the same programming at the same time. The digital picture is always delayed by a few seconds. This is true whether the content is live or prerecorded.

Display Order (input to the encoder): I1 B2 B3 B4 P5 B6 B7 B8 P9

Bitstream Order (the order in which the frames are sent out of the encoder): I1 P5 B2 B3 B4 P9 B6 B7 B8

For the decoder, the input is the bitstream order and the output is the display order.

Where do the Predictions Come From?

Depends on the kind of picture.

Predictions for P picture

If it's a **frame picture** they may come from the previous I or P frame. If it's a **field picture** they may come from the two I or P fields coded most recently. So, must be concerned if interlaced or progressive scan.

Predictions for B picture

They may come from the previous (in display order) I or P frame as from the next (in display order) I or P frame and there may be an **interpolation** between predictions coming from both directions.

Group of Pictures (GOP)

We will see this many times. Very important.

As an example the frame sequence transferred as I P B B B P B B B. The only task of the decoder is to reorder the reconstructed frames. To support this an ascending frame number comes with each frame. Diagram 1.

Diagonal stripes have high frequency information in horizontal and vertical directions, producing high values in the bottom right corner of the DCT output matrix. Here we have a group of 4 by 4 pixels. Diagram 2. This is the coding that would come out of an alternating black-white in a 4 by 4 pixel block. DCT is **discrete cosine transform**. Notice the block is the Y channel with black and white blocks, but doesn't correlate to our graph/table (correlation between the pixel table and the number table).

The Discrete Cosine Transform

Many image compression techniques available, but transform coding is the preferred method. Since energy distribution varies with each image, compression in the spatial domain is not easily attained. Transform coding is used to maximize compression of images in the frequency domain. Don't try to understand the frequency domain; it is very complex. Transform from an analog domain to a frequency domain

Homework

Read up in MPEG compression in our text book. Find out all information about I, P, and B frames in the text book. Quiz on Tuesday about the information we find in the text book. Don't get into vectors in depth, just follow through and try to understand what is being discussed.