Video Production

Post Production Studios have many VCR's using 1" tapes which have double the standard TV resolution. The reason for using these tapes is that editing will cause quality losses that will bring the finished product close to standard resolution.

Linear Editing

Linear Editing uses sequential storage (Video Tapes) to store video sequences. The important property of linear editing is that scene cannot be accessed out of sequence. There are two types of editing, cuts only and A-B roll. Cuts only takes one input, makes a cut and writes to the target placing a special effect between the cuts. A-B rolls take input from two sources, make cuts, and writes to a target placing special effects between cuts.

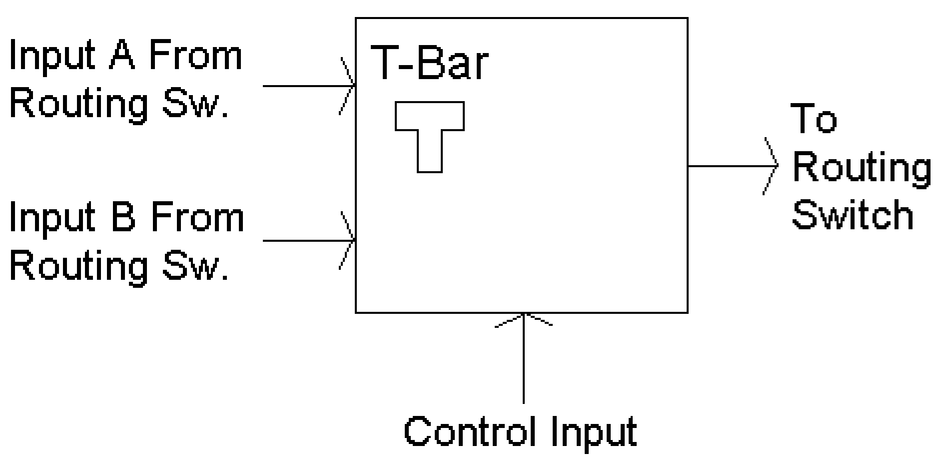
Non-linear Editing

Storage can be accessed non-sequentially. Scenes can be rearranged and accessed at random.

Editing Hardware

Routing Switches are 10x10 switches that can select one of ten inputs and route it to one of ten outputs.

Edit Controller: Edit controllers take input from one or two sources (cuts only or A-B roll), make a cut, and insert a special effect while writing to the output. They have a T-Bar switch that determines the speed of the special effect. The faster the T-Bar is pulled down, the faster the effect.



Types of Special Effects

- Cut
- Fade: A fades out, then B fades in
- Dissolve: A fades out while B fades in
- Wipe: New input slides in horizontally
- Tumble: New input slides in vertically
- Wrapping: Wrap A to a surface, then unwrap B
- Keying: Send a waveform through control input

Analog vs. Digital Editing

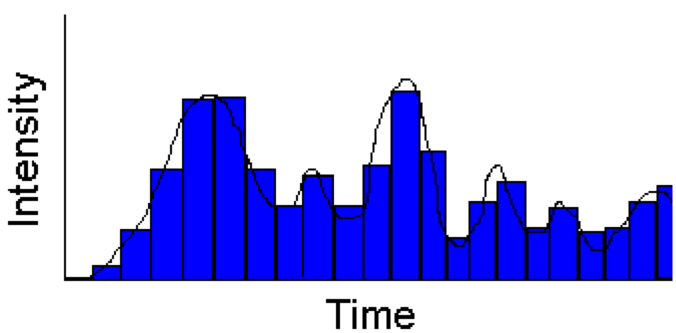
Currently these editing procedures are analog. Converting these operations into digital form is easy if you can operate on uncompress video. Unfortunately, working on uncompressed video is unrealistic due to its large size. Therefore, these operations must be performed on compressed video which is difficult to do.

Digital Media

The following steps must be done in order to convert an analog signal to a digital signal. First, *Sample* the analog input at a given sampling rate. Then, *Quantize* the input, rounding its values to the nearest quantum.

Sampling

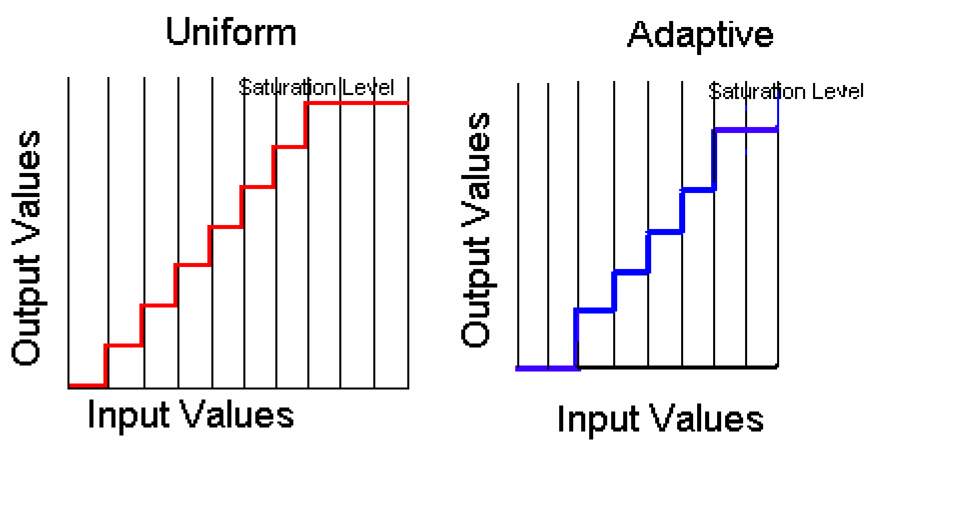
Sampling rates are governed by *Nyquist's Law* which states that to sample an analog signal in a manner which permits you to reconstruct the original signal, you must sample it at or above twice the maximum frequency of the analog signal. Therefore, since the maximum frequency used in human speech is about 4KHz, telephones can sample at 8KHz. Also, since the highest audible frequency for humans is around 22KHz, CD music is sampled at 44.1KHz.



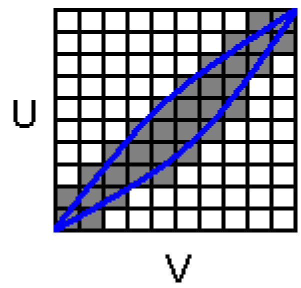
Sampling will take an analog signal (the black line above) and convert into digital representation (the blue bars). Each of these blue bars represents the intensity of the signal at a point in time.

Quantization

In a digital world, you must specify a finite precision (generally in bits) to represent a value. In order to convert an analog signal to a digital one, the intensity values must be rounded to a quantum. This is why a digitized signal loses most of its quality compared to original, analog signal. There are two methods for defining the size of a quantum, Uniform and Adaptive quantization. Uniform quantization breaks the space of possible values into even intervals, so if you wanted to represent a signal with 8 bits, the values would evenly divide the space of possible values by 256 (2^8). Adaptive quantization breaks up the space of possible values in a manner typically defined by properties of the signal. If, for example, you know that human hearing is more sensitive to intensity values near the middle of the range of possible values than it is to very high or very low values, you might want to assign more bits to the middle of the range. Examples of uniform and adaptive quantization are shown below:



When quantizing multi-dimensional values such as pixel values, there are two methods: Scalar Quantization and Vector Quantization. Scalar quantization merely quantizes each of the various dimensions independently, ignoring relationships between them. For example, digitizing a color image based upon separate 8-bit values for a red channel, a blue channel, and a green channel. On the other hand, vector quantization takes advantage of the relation between the dimensions. If, for example, you know that YUV images of a certain type tend to have the same U and V values, you can reduce the number of possible values further. In the following example, if you were to have independent U and V values, you would have 10x10 or 100 possible values (7 bits). If, however, you know that the U and V values tend to fall between the blue lines, you can reduce the number of possible values (grey squares) to 32 (5 bits) . This saves ~30%!



Digital Audio

Audio signals are generally quantized using scalar quantization. Vector quantization can be used if the signal is broken into frequency components and then quantized, but this added complexity generally provides little, if any savings in size. Audio can be encoded in three primary ways:

1. **Pulse Code Modulation (PCM)** audio is encoded by simply storing the intensity values for each sample. E.g. 10, 11, 12, 14, 12
2. **Differential PCM (DPCM)** audio takes advantage of the fact that changes between audio samples are generally small, so it sends an intensity value followed by changes from the previous sample. E.g. 10, +1, +1, +2, -2. In order to facilitate recovery from transmission errors, an intensity value will be sent periodically instead of a difference.
3. **Adaptive DPCM (ADPCM)** further reduces size by storing the sign of the difference and a scaling factor. E.g. 10, (1,+), +, (2,+), -.