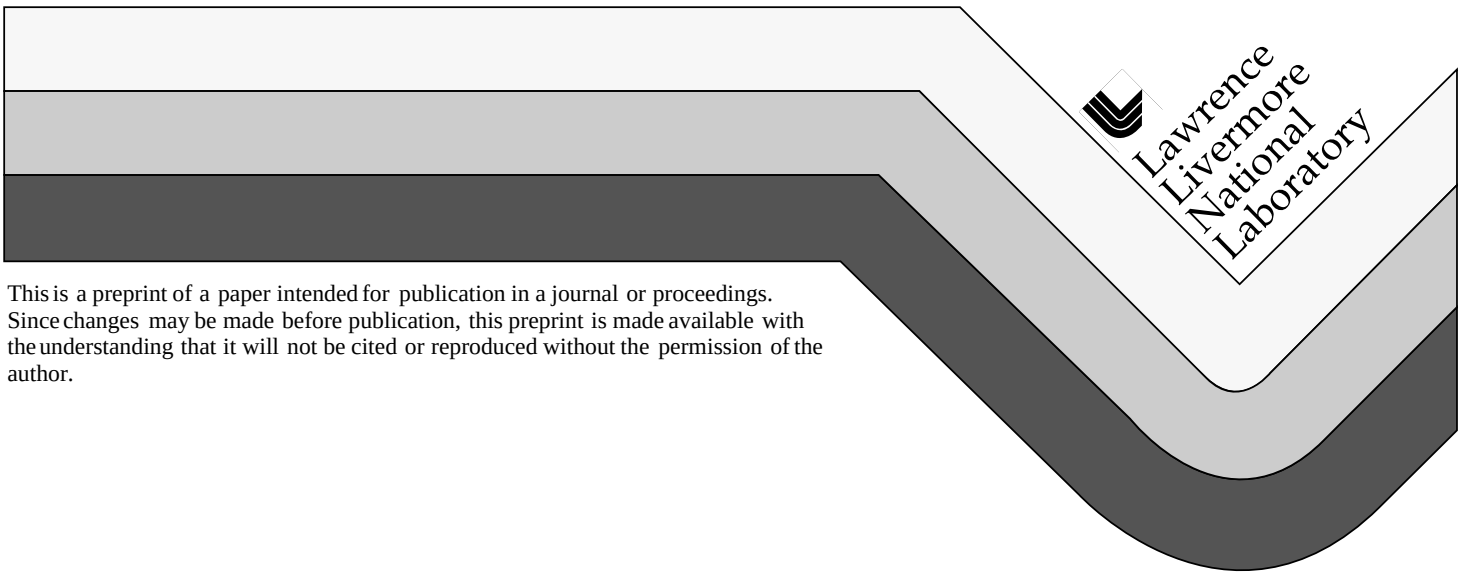
UCRL-JC-127331
PREPRINT

Color Spaces in Digital Video

R. Gaunt

This paper was prepared for submittal to the
Association for Computing Machinery
Special Interest Group on Computer Graphics (SIGGRAPH) '97 Conference
Los Angeles, CA
August 3-8, 1997

May 1997



Color Spaces in Digital Video

Ross Gaunt
Lawrence Livermore National Laboratory

Whether it's photography, computer graphics, publishing, or video; each medium has a defined color space, or gamut, which defines the extent that a given set of RGB colors can be mixed. When converting from one medium to another, an image must go through some form of conversion which maps colors into the destination color space. The conversion process isn't always straight forward, easy, or reversible.

In video, two common analog composite color spaces are Y'UV (used in PAL) and Y'IQ (used in NTSC). These two color spaces have been around since the beginning of color television, and are primarily used in video transmission. Another analog scheme used in broadcast studios is Y', R'-Y', B'-Y' (used in Betacam and MII) which is a component format. Y', R'-Y',B'-Y' maintains the color information of RGB but in less space. From this, the digital component video specification, ITU-Rec. 601-4 (formerly CCIR Rec. 601) was based. The color space for Rec. 601 is symbolized as Y'CbCr. Digital video formats such as DV, D1, Digital–S, etc., use Rec. 601 to define their color gamut. Digital composite video (for D2 tape) is digitized analog Y'UV and is seeing decreased use.

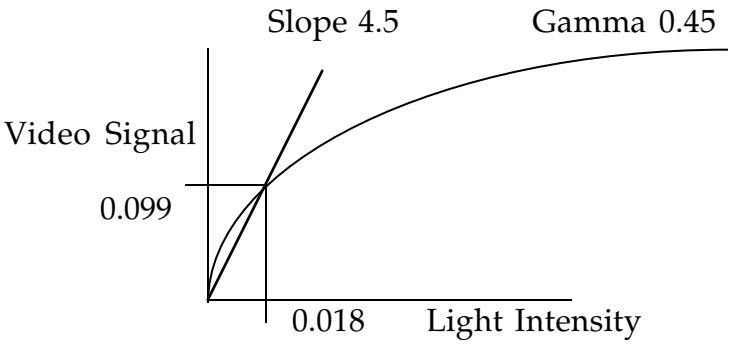
Because so much information is contained in video, segments of any significant length usually require some form of data compression. All of the above mentioned analog video formats are a means of reducing the bandwidth of RGB video. Video bulk storage devices, such as digital disk recorders, usually store frames in Y'CbCr format, even if no other compression method is used. Computer graphics and computer animations originate in RGB format because RGB must be used to calculate lighting and shadows. But storage of long animations in RGB format is usually cost prohibitive and a 30 frame-per-second data rate of uncompressed RGB is beyond most computers.

By taking advantage of certain aspects of the human visual system, true color 24-bit RGB video images can be compressed with minimal loss of visual information. For example, humans 'see' more white-to-black (luminance) detail then red, green, or blue color detail. Also, the eye is most sensitive to green colors. Taking advantage of this, both composite and component video allocates more bandwidth for the luma (Y') signal than the chroma signals. Y'_{601} is composed of 59% green', 30% red', and 11% blue' (prime symbol denotes gamma corrected colors). This luma signal also maintains compatibility with black and white television receivers. Component digital video converts R'G'B' signals (either from a camera or a computer) to a monochromatic brightness signal Y' (referred here as luma to distinguish it from the CIE luminance linear-light quantity), and two color difference signals Cb and Cr. These last two are the blue and red signals with the luma component subtracted out.

GAMMA: As you know, computer graphic images are composed of red, green, and blue elements defined in a linear color space. Color monitors do not display RGB linearly. A linear RGB color space image must be gamma corrected to be displayed properly on a CRT. Gamma correction, which is approximately a 0.45 power function, must also be employed before converting an RGB image to video color space. Gamma correction is defined for video in the international standard: ITU-Rec. BT.709-4. The gamma correction transform is the same for red, green, and blue. The formula for the red signal is shown below. Note, the below formulas and graph are from Poynton.

$$R'_{709} = \begin{cases} 4.5R, & R <= 0.018 \\ 1.099\,R^{0.45} - 0.099, & 0.018 < R \end{cases}$$

In order to minimize noise in the dark regions of a picture, from a camera for example, the slope of the function is limited near black.



High definition television uses a gamma correction transform defined in SMPTE Rec. 240M, which is very similar to Rec. 709. For images, the difference between these two transfer functions is negligible.

$$R'_{240M} = \begin{cases} 4.0R, & R <= 0.0228 \\ 1.1115\,R^{0.45} - 0.1115, & 0.0228 < R \end{cases}$$

Y'CbCr: The color coding standard for component digital video and high definition video symbolizes gamma corrected luma by Y', the blue difference signal by Cb (Cb = B' -Y'), and the red color difference signal by Cr (Cr = R' - Y'). Component analog HDTV uses Y'PbPr. The document "Digital Signal Coding" contains a more complete definition on Rec. 601.

R'G'B' to Y'CbCr: To convert gamma corrected RGB (range 0 to 1), use the below transform:

$$\begin{bmatrix} Y'_{601} \\ C_b \\ C_r \end{bmatrix} = \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} + \begin{bmatrix} 65.481 & 128.553 & 24.966 \\ -37.797 & -74.203 & 112. \\ 112. & -93.786 & -18.214 \end{bmatrix} \bullet \begin{bmatrix} R' \\ G' \\ B' \end{bmatrix}$$

The inverse is:

$$\begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} = \begin{bmatrix} 0.00456621 & 0. & 0.00625893 \\ 0.00456621 & -0.00153632 & -0.00318811 \\ 0.00456621 & 0.00791071 & 0. \end{bmatrix} \bullet \begin{bmatrix} Y'_{601} \\ C_b \\ C_r \end{bmatrix} + \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix}$$

To convert gamma correct RGB (range 0 to 255, 8-bits per color) use:

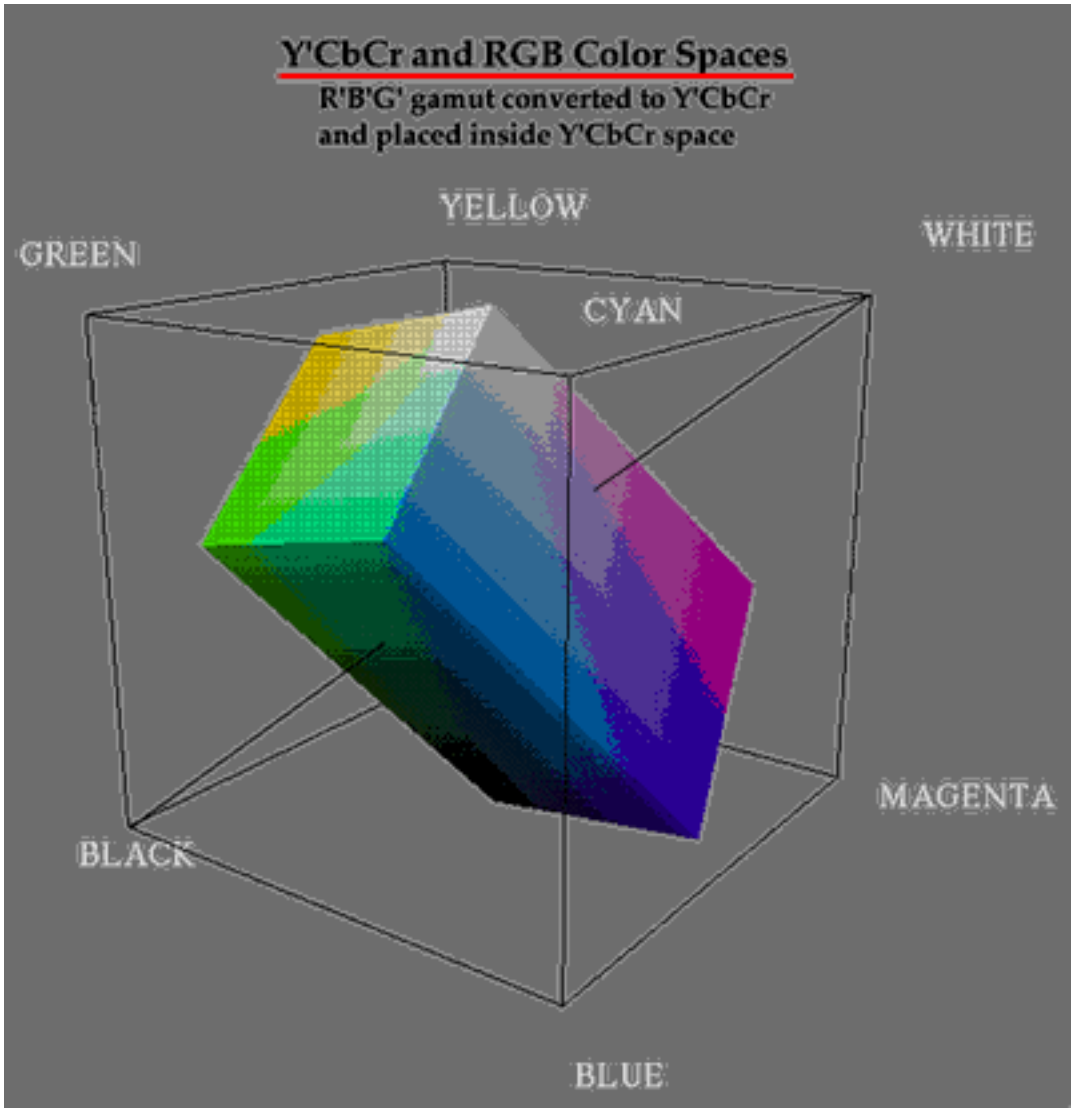
$$\begin{bmatrix} Y'_{601} \\ C_b \\ C_r \end{bmatrix} = \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} + \frac{1}{256} \begin{bmatrix} 65.738 & 129.057 & 25.064 \\ -37.945 & -74.494 & 112.439 \\ 112.439 & -94.154 & -18.285 \end{bmatrix} \bullet \begin{bmatrix} R'_{255} \\ G'_{255} \\ B'_{255} \end{bmatrix}$$

The inverse is:

$$\begin{bmatrix} R'_{255} \\ G'_{255} \\ B'_{255} \end{bmatrix} = \frac{1}{256} \bullet \begin{bmatrix} 298.082 & 0. & 408.583 \\ 298.082 & -100.291 & -208.120 \\ 298.082 & 516.411 & 0. \end{bmatrix} \bullet \left(\begin{bmatrix} y'_{601} \\ C_b \\ C_r \end{bmatrix} - \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} \right)$$

Use 9-bit or larger multipliers.

Problems with converting Y′CbCr: The figure below shows the R′G′B′ color cube converted to Y′CbCr space. The inner cube is the converted maximum range of R′G′B′ (0-255) and the outer wire frame is the legal gamut of Y′CbCr space (16-240). The values above and below this range provide headroom for analog filter overshoot.

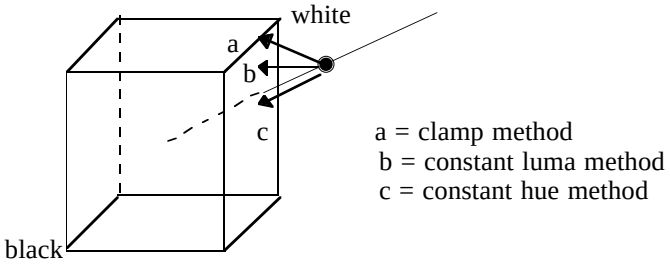


Most noticeably, the R'G'B' cube is rotated and scaled inside the Y'CbCr space. The black to white axis of the R'G'B' cube is rotated from the black to white axes of the Y'CbCr cube. Also, the R'G'B' cube is completely contained inside the outer cube. This leaves a large region between the boundaries of the inner cube and the wireframe. So, while every value of R'G'B', when converted to Y'CbCr space, is represented, not every value in Y'CbCr space can be directly converted to R'G'B' space.

So how do Y'CbCr values outside the R'G'B' cube occur? After all, R'G'B' form a camera or computer image will not produced out of range values. But adding or multiplying the values of an image can produce out of range values. Changes in hue or luma can be produced by image filtering or digital effects processing.

Image processing and conversion adds some amount of error, and with each generation the degree of error increases. After several generations, values may go out of range. An RGB image written to a video disk will be stored in Y'CbCr format. If the image is read back into the computer it will most likely be converted back to RGB, and it will be two generations away from the original image.

Converting from R'G'B' to Y'CbCr will not produce out of range values, but when a conversion is performed from Y'CbCr to R'G'B', out of range colors must be somehow mapped back to colors that are representable. Predicting the results is difficult, due to the rotation and scaling operations. The most common mapping method is to clamp each value to the available range in the destination space. This is the method hardware converters use. Two other methods are the constant hue and constant luma algorithms, which can achieve more accurate results.



In these three approaches, the values are projected back to the destination color space, and are given colors at the boundary along the projection. Of these two schemes, a more accurate image results when the constant hue method is used. Constant luma produces color shifts, but constant hue allows luma to vary. Refer to chapter 5 of Roy Hall's book for detailed information on these algorithms.

RGB is 4:4:4, i.e. there is an 8-bit red, green, and blue value for every pixel. Y'CbCr usually is 4:2:2, meaning there is a luma value for each pixel but only every alternate pixel has a chroma value. Converting RGB to 4:2:2 Y'CbCr and the back to RGB produces soft edges. The problem is how to reconstruct the missing chroma values. Since it is unknown what the original value was, you could guess the value by interpolating between the previous and the next known chroma values. Clearly this produces more problems than it solves, but it is impossible to exactly reconstruct a 4:4:4 image from a 4:2:2 image.

To reduce conversion errors, clip in R'G'B', not in Y'CbCr space. View video on a video monitor, computer monitor phosphors are wrong. Use a large word size (double precision) to avoid warp around, then round the results to values between 0 and 255. And finally, recall that multiplying two 8-bit numbers results in a 16-bit number, so values need to be clipped to 8-bits.

References:

Hall, Roy, Illumination and Color in Computer Generated Imagery, Springer-Verlag.

Jack, Keith, Video Demystified, Brooktree Corp., Hightext Publishers, Solana Beach Calif., 1993.

National Association of Broadcasters Conference 1996 panel, “CCIR 601: How it Should Apply to Computer Platforms?”.

Poynton, Charles A., A Technical Introduction to Digital Video, John Wiley & Sons, Inc., New York, 1996.

www.hooked.net/users/rld/color_faq.html.

This work was performed under the auspices of the U.S. Dept. of Energy at LLNL under contract no. W-7405-Eng-48.

Technical Information Department • **Lawrence Livermore National Laboratory**
University of California • Livermore, California 94551