SIGGRAPH2004

## COLOR SCIENCE AND COLOR APPEARANCE MODELS FOR CG, HDTV, AND D-CINEMA

### COURSE NOTES – TABLE OF CONTENTS

**Charles Poynton** is an independent contractor specializing in the physics and electronics of digital color imaging systems, including digital video, HDTV, and digital cinema. While at Sun Microsystems, from 1988 to 1995, he initiated Sun's HDTV research project, and introduced color management technology to Sun. Prior to joining Sun, Mr. Poynton designed and built the digital video equipment used at NASA's Johnson Space Center to convert video from the Space Shuttle into NTSC for recording and distribution. He is a Fellow of the Society of Motion Picture and Television Engineers (SMPTE), and an Honorary Member of the BKSTS. In 1994 he was awarded SMPTE's David Sarnoff Gold Medal for his work integrating video technology with computing and communications. He has organized and presented many popular courses and seminars, including *HDTV Technology* at SIGGRAPH 91, *Concepts of Color, Video and Compression* at ACM Multimedia 93, and courses on color technology at SIGGRAPHs in 1994 and 1996 through 2000. His 1996 book, *A Technical Introduction to Digital Video*, was reprinted five times. In February 2003, his new book *Digital Video and HDTV Algorithms and Interfaces* was the 3,339$^{th}$ most popular item at Amazon.com.

**Garrett M. Johnson** is a Color Scientist at the Munsell Color Science Lab at Rochester Institute of Technology. He was awarded a Ph.D. in the Imaging Science program at RIT, under Mark Fairchild; his dissertation concerned Image Difference, Quality and Appearance. He holds a B.S. in Imaging Science and an M.S. in Color Science, both from RIT. He has co-authored several journal articles with Fairchild and others at RIT, including "Spectral Color Calculations in Realistic Image Synthesis," published in *IEEE Computer Graphics and Applications.*

SIGGRAPH2004

## COLOR SCIENCE AND COLOR APPEARANCE MODELS FOR CG, HDTV, AND D-CINEMA

This course introduces the science behind image digitization, tone reproduction, and color reproduction in computer generated imagery (CGI), HDTV, and digital cinema (D-cinema). We detail how color is represented and processed as images are transferred between these domains. We detail the different forms of nonlinear coding ("gamma") used in CGI, HDTV, and D-cinema. We explain why one system's *RGB* does not necessarily match the *RGB* of another system. We explain color specification systems such as CIE *XYZ*, *L*a*b**, *L*u*v**, *HLS*, *HSB*, and HVC. We describe why the coding of color image data has a different set of constraints than color specification, and we detail color image coding systems such as *RGB*, *R'G'B'*, *CMY*, $Y'C_BC_R$, and DPX/Cineon. We explain color measurement instruments such as densitometers and colorimeters, and we explain monitor calibration. We explain how color management technology works, and how it is currently being used in motion picture film production (both animation and live action).

Reproducing the tristimulus numbers of classical color science only reproduces colors accurately in an identical viewing environment. If the viewing situation changes, color is not completely described by numbers. In applying color science to image reproduction, we wish to reproduce images in environments where angular subtense, background, surround, and ambient illumination may differ from the conditions at image origination. Recent advances in color appearance modelling allow us to quantify the alterations necessary to reproduce color appearance in different conditions. We will introduce the theory and standards of color appearance models. We will then describe the application of color science and color appearance models to commercial motion imaging in computer graphics, video, HDTV, and D-cinema.



Charles Poynton
Organizer/Presenter

tel +1 416 413 1377
poynton@poynton.com
www.poynton.com

Garrett Johnson
Presenter

tel +1 585 475 4923
garrett@cis.rit.edu
www.cis.rit.edu/mcsl

## Lightness terminology

In a grayscale image, each pixel value represents what is loosely called *brightness*. However, brightness is defined formally as *the attribute of a visual sensation according to which an area appears to emit more or less light*. This definition is obviously subjective, so *brightness* is an inappropriate metric for image data.

*Intensity* refers to radiant power in a particular direction; *radiance* is intensity per unit projected area. These terms disregard wavelength composition. However, if color is involved, wavelength matters! Neither of these quantities is a suitable metric for color image data.

The term *luminance* is often used carelessly and incorrectly to refer to *luma;* see below.

In image reproduction, we are usually concerned not with (absolute) luminance, but with *relative luminance*.

*Luminance* is radiance weighted by the spectral sensitivity associated with the brightness sensation of vision. Luminance is proportional to intensity. Imaging systems rarely use pixel values proportional to luminance; usually, we use values nonlinearly related to luminance.

Regrettably, many practitioners of computer graphics, and of digital image processing, have a cavalier attitude toward these terms. In the *HSB, HSI, HSL,* and *HSV* systems, *B* allegedly stands for brightness, *I* for intensity, *L* for lightness, and *V* for value. None of these systems computes brightness, intensity, luminance, or value according to any definition that is recognized in color science!

*Lightness* – formally, CIE $L^*$ – is the standard approximation to the perceptual response to luminance. It is computed by subjecting luminance to a nonlinear transfer function that mimics vision. A few grayscale imaging systems code pixel values in proportion to $L^*$.

*Value* refers to measures of lightness apart from CIE $L^*$. Imaging systems rarely, if ever, use *Value* in any sense consistent with accurate color.

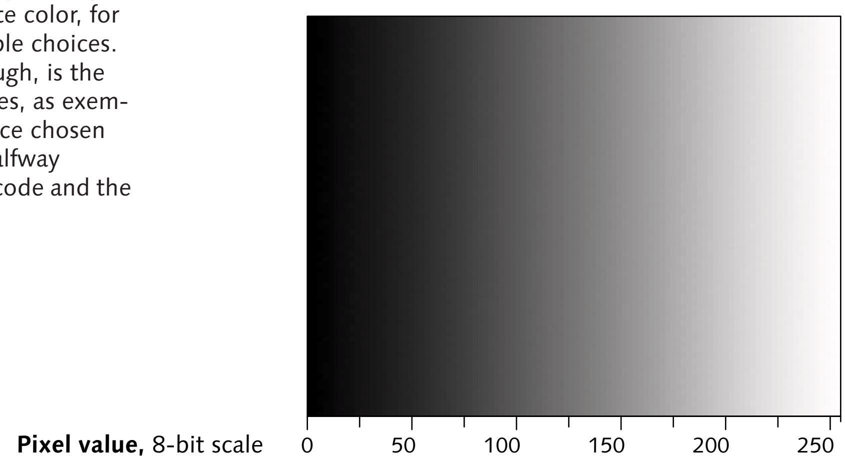
Color images are sensed and reproduced based upon *tristimulus values*, whose amplitude is proportional to intensity, but whose spectral composition is carefully chosen according to the principles of color science. As their name implies, tristimulus values come in sets of 3.

Accurate color imaging starts with values, proportional to radiance, that approximate *RGB* tristimulus values. (I call these values *linear-light*.) However, in most imaging systems, *RGB* tristimulus values are subject to a nonlinear transfer function – *gamma correction* – that mimics the perceptual response. Most imaging systems use *RGB* values that are *not* proportional to intensity. The notation *R'G'B'* denotes the nonlinearity.

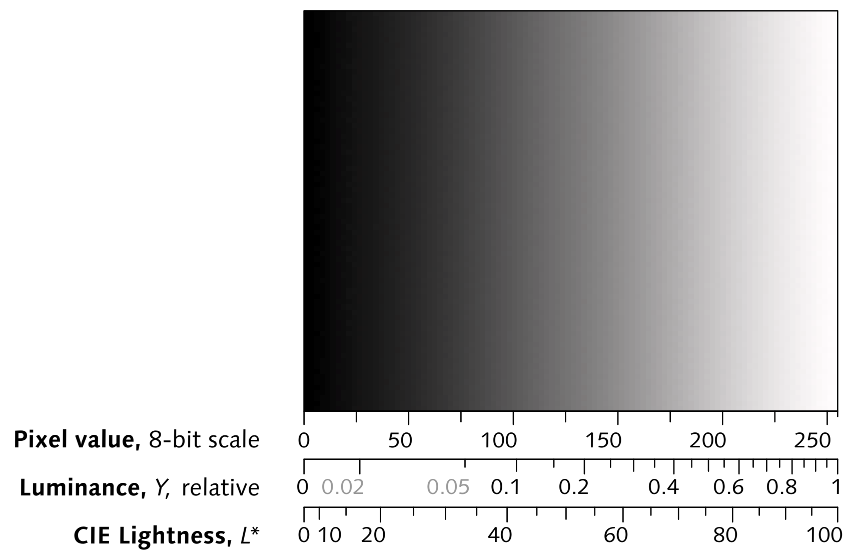
See *YUV and luminance considered harmful*, available at www.poynton.com

*Luma* (*Y'*) is formed as a suitably-weighted sum of *R'G'B'*; it is the basis of luma/color difference coding. Luma is comparable to lightness; it is often carelessly and incorrectly called *luminance* by video engineers.

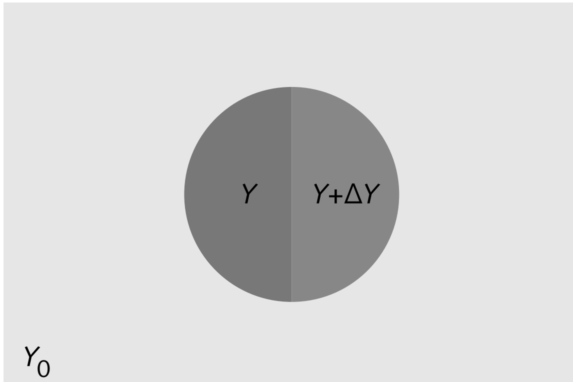
**Grayscale values in digital imaging** are usually represented as nonnegative integer code values, where zero represents black, and some positive value – in 8-bit systems, typically 255 – represents the maximum white. The interpretation of the black code is fairly straightforward. The interpretation of white depends upon the choice of a reference white color, for which there are several sensible choices. Perhaps most important, though, is the mapping of intermediate codes, as exemplified by the relative luminance chosen for the code value that lies halfway between the reference black code and the reference white code.



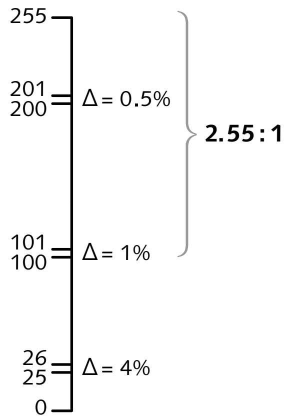

**Grayscale ramp** on a CRT display is generated by writing successive integer values 0 through 255 into the columns of a framebuffer. When processed by a digital-to-analog converter (DAC), and presented to a CRT display, a perceptually uniform sweep of lightness values results. A naive experimenter might conclude – mistakenly! – that code values are proportional to intensity.



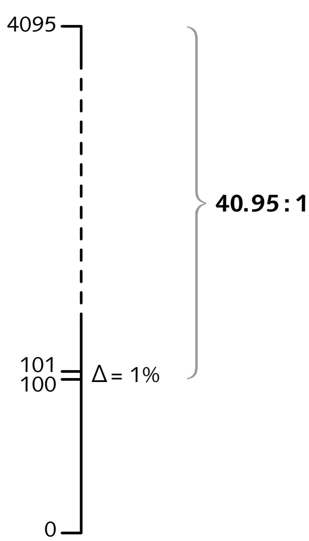

**Grayscale ramp augmented** with CIE relative luminance (*Y*, proportional to intensity, on the middle scale), and CIE lightness (*L**, on the bottom scale). The point midway across the screen has lightness value midway between black and white. There is a near-linear relationship between code value and lightness. However, luminance at the midway point is only about 20 percent of white! Luminance produced by a CRT is approximately proportional to the 2.5-power of code value. Lightness is roughly proportional to the 0.4-power of luminance. Amazingly, these relationships are near inverses. Their near-perfect cancellation has led many workers in computer graphics to misinterpret the term *intensity*, and to underestimate the importance of nonlinear transfer functions.




**Contrast sensitivity test pattern** is presented to an observer in an experiment to determine the contrast sensitivity of human vision. The experimenter adjusts $\Delta Y$, and the observer is asked to report when he or she detects a difference in lightness between the two halves of the patch. The experiment reveals that the observer cannot detect a difference between luminances when the ratio between them is less than about one percent. Lightness is roughly proportional to the logarithm of luminance. Over a wider range of luminance levels, strict adherence to logarithmic coding is not justified for perceptual reasons. In addition, the discrimination capability of vision degrades for very dark shades of gray, below several percent of peak white.

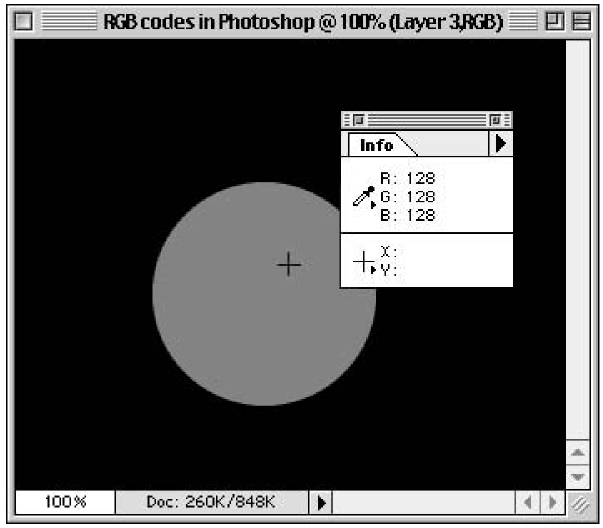



**Linear light coding.** Vision can detect that two luminances differ if their ratio exceeds 1.01 (or so). Consider coding luminance values in 8 bits. With linear light coding, where code zero represents black and code 255 represents white, code value 100 represents a shade of gray that lies near the perceptual threshold. For codes below 100, the ratio of luminances between adjacent code values is exceeds 1.01: At code 25, adjacent codes differ by 4 percent, which is objectionable to most observers. For codes above 100, adjacent codes differ by less than 1 percent: Code 201 is perceptually useless, and could be discarded without being noticed.



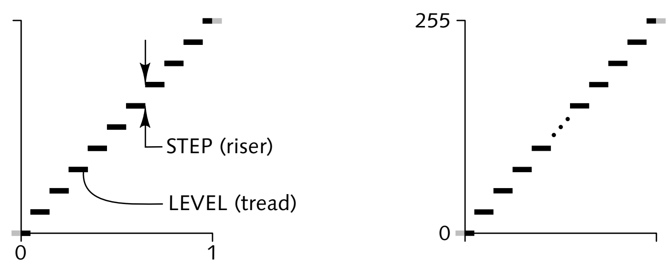

**The "code 100" problem is mitigated** by using more than 8 bits to represent luminance. Here, 12 bits are used, placing the top end of the scale at 4095. Twelve-bit linear coding is potentially capable of delivering images with a contrast ratio of about 40:1 without contouring; however, of the 4096 codes in this scale, only about 100 can be distinguished visually: The coding is inefficient.

**Image coding in computing.** The interior of the disc has *RGB* codes [128, 128, 128] in Photoshop, halfway up the code scale from black to white. However, its luminance reproduced on the screen, or its reflectance on the printed page, is usually not proportional to code value. On a Macintosh, reproduced intensity is proportional to code value raised to the 1.8-power, so the disc will be reproduced at a luminance of about 28% of white.

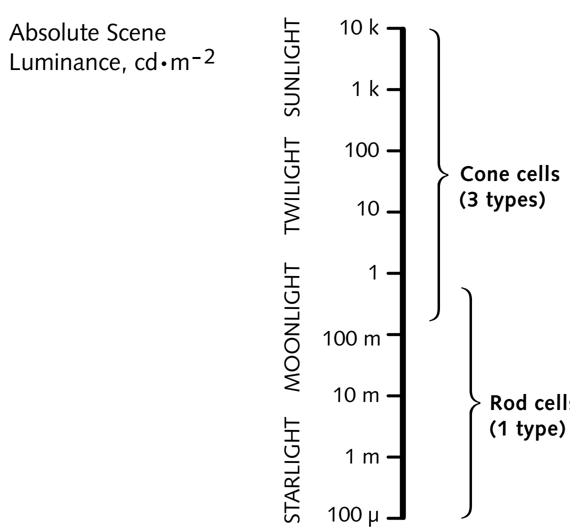


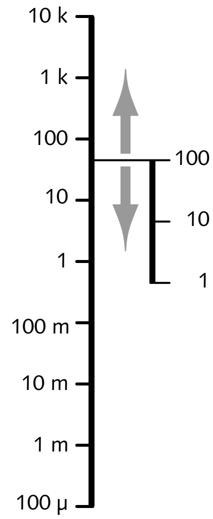

**Uniform quantization** has equal-amplitude steps. Though uniform quantization is sketched here, the signals ordinarily quantized in video have been subjected to a nonlinear transfer function, and so are not proportional to light intensity.

**Quantized range in computing** usually uses 8 bits, with code 0 for reference black, and code 255 for reference white. Ordinarily, *R'G'B'* values are perceptually coded.

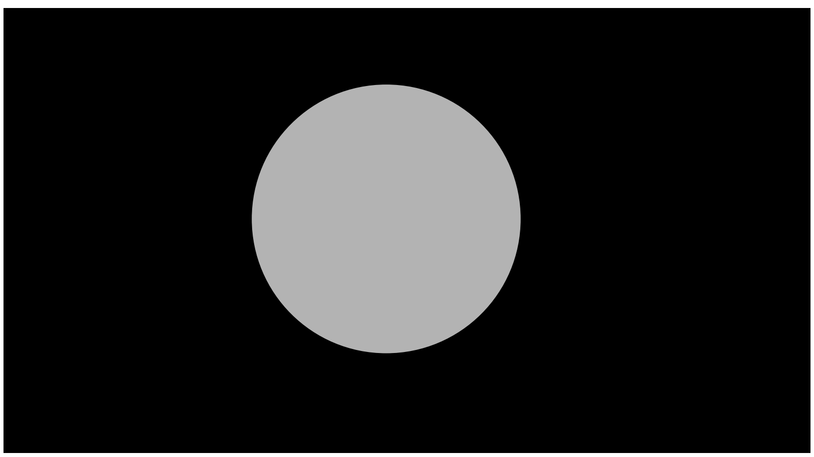



**Intensity range of vision** encompasses about seven decades of dynamic range. For the top four or five decades of the intensity range, photoreceptor cells called cones are active. There are three kinds of cones, sensitive to longwave, mediumwave, and shortwave light – roughly, light in the red, green, and blue portions of the spectrum. The cones are responsible for color vision. For three or four decades at the bottom end of the intensity range, the retinal photoreceptor cells called *rods* are employed. (Since there is only one type of rod cell, what is loosely called *night vision* cannot discern colors.)

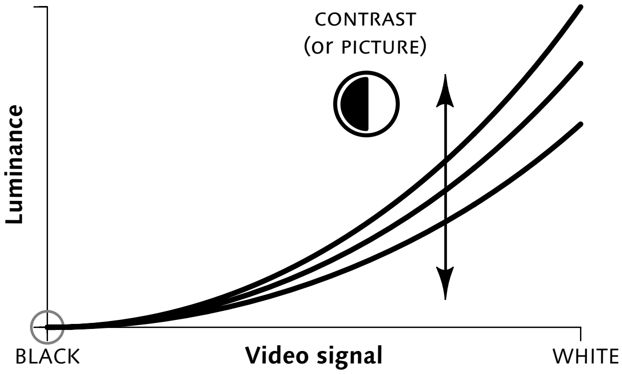




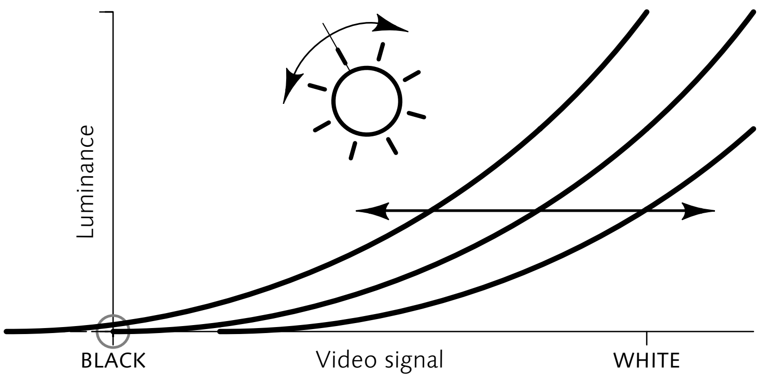

**Adaptation.** Across the seven decade intensity range of vision, about one decade of adaptation is effected by the iris; the remainder is due to a photochemical process involving the *visual pigment* substance contained in photoreceptor cells. At any particular adaptation level, vision makes use of about a 100:1 range of intensities. In image reproduction, luminance levels less than about 1 percent of white cannot be distinguished.



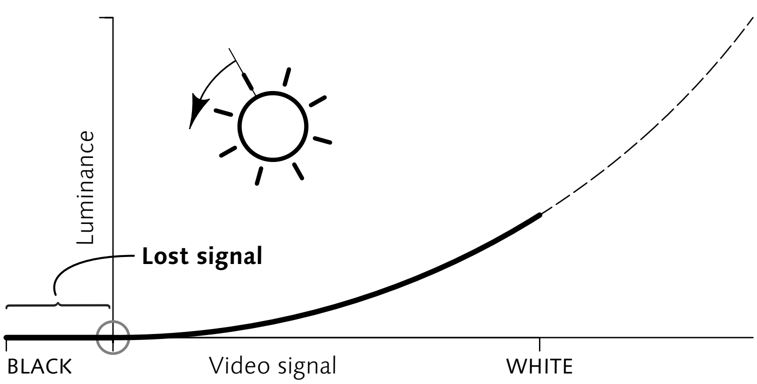
**Adaptation to white.** The viewer's notion of white depends upon viewing conditions. When this image is projected in a dark room, the central circle appears white. In print media or on a video monitor, it appears mid gray. Adaptation is closely related to the intensity of "white" in your field of view.




**"CONTRAST" control.** Almost every video monitor has two main front panel controls. The CONTRAST control, sometimes called PICTURE, adjusts the electrical gain of the signal, thereby adjusting the white level while having minimal effect on black.

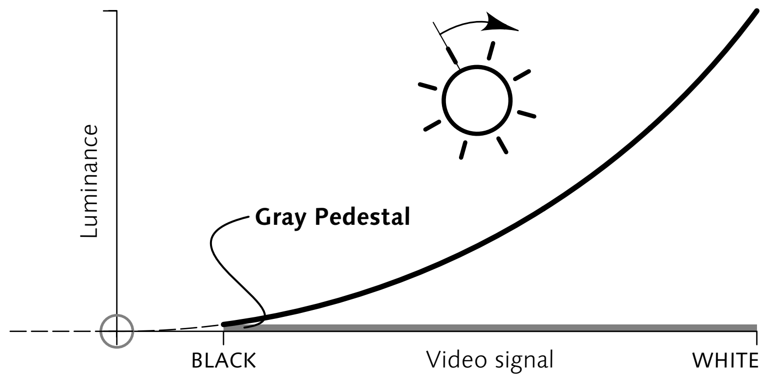



**The "BRIGHTNESS" control,** sometimes called BLACK LEVEL, adjusts the electrical offset of the signal. It has an equivalent electrical effect across the entire black-to-white range of the signal, but due to the nonlinear nature of the transfer function from voltage to intensity its effect is more pronounced near black.

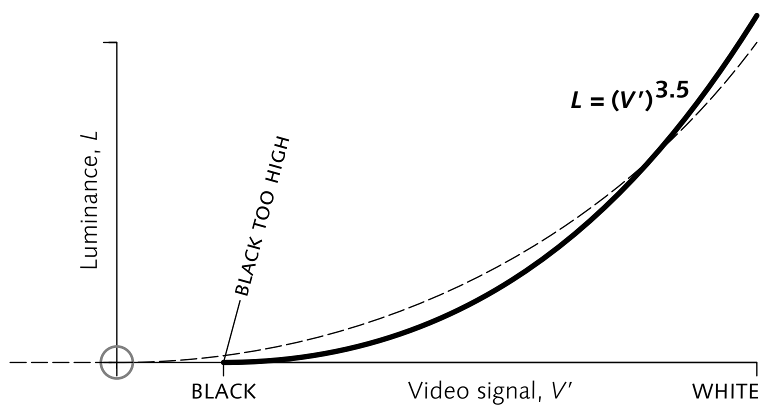



**"BRIGHTNESS" too low.** If BRIGHTNESS is adjusted too low, portions of the video signal near black are clipped (or *swallowed*) – they produce the identical shade of black at the CRT, and cannot be distinguished. This is evident to the viewer as loss of picture information in dark areas of the picture, or as a cinematographer would say, loss of detail in the shadows. BRIGHTNESS is set correctly at the threshold where it is low enough to avoid introducing a gray pedestal, but not so low that codes near black start being clipped.

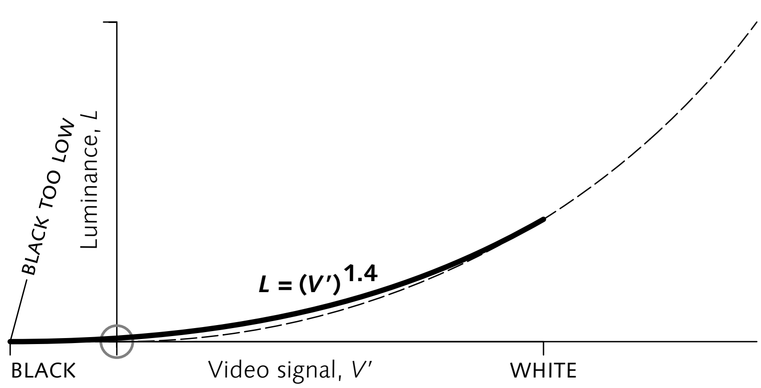
**“Brightness” too high.** If the Brightness control of a CRT monitor is adjusted too high, then the entire image is reproduced on a pedestal of dark gray. This reduces the contrast ratio of the image. Contrast ratio is a determinant of perceived image sharpness, so an image whose black level is too high will appear less sharp than the same image with its black level reproduced correctly.

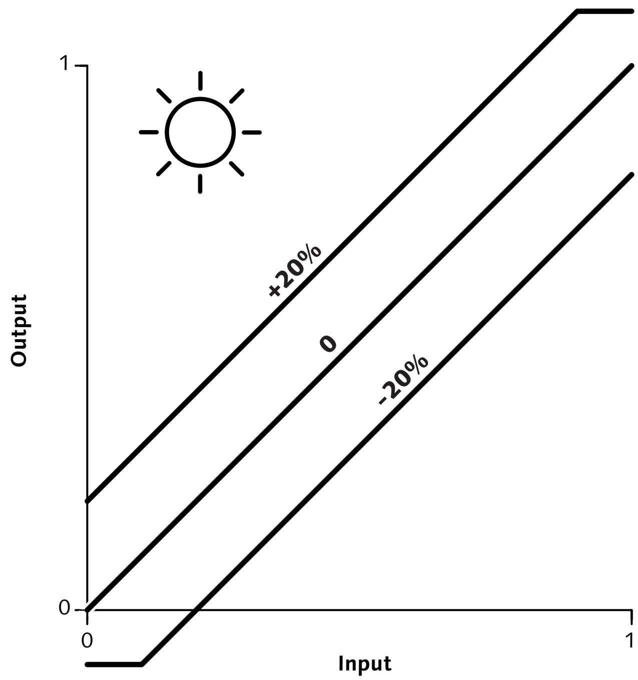



**Gamma 3.5** A naive approach to the measurement of CRT nonlinearity is to model the response as $L = (V')^{\gamma}$, and to find the exponent of the power function that is the best fit to the voltage-to-intensity transfer function of a particular CRT. However, if this measurement is undertaken with Brightness set too high, an unrealistically large value of gamma results from the modelled curve being “pegged” at the origin.

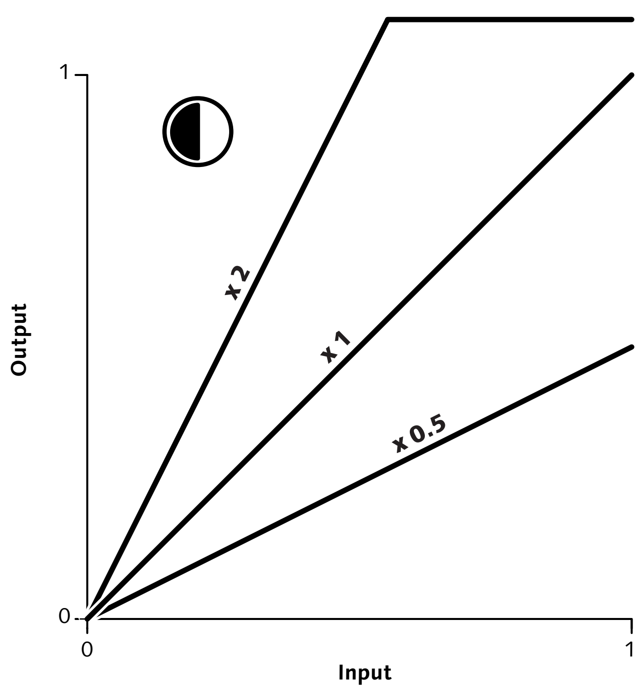



**Gamma 1.4** If the transfer function of a CRT is modelled as $L = (V')^{\gamma}$ with Brightness set too low, an unrealistically small value of gamma results. However, if the transfer function is modeled with a function of the form $L = (V' + \varepsilon)^{2.5}$ that accommodates black level error, then a good fit is achieved. Misintepretations in the measurement of CRT nonlinearity have led to assertions about CRTs being highly unpredictable devices, and have led to image exchange standards employing quite unrealistic values of gamma.



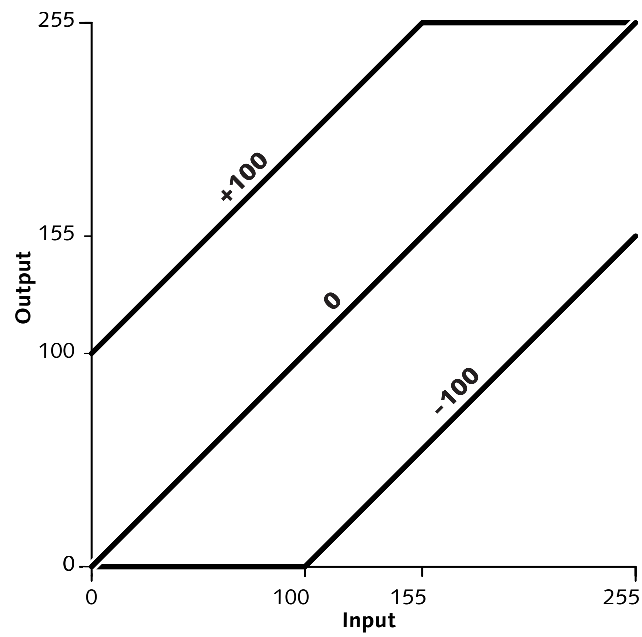



**Brightness (or Black Level) control in video** applies an offset, roughly ±20% of full scale, to *R'G'B'* components. At the minimum and maximum settings, I show clipping to the Rec. 601 studio standard footroom ($-^{15}/_{219}$) and headroom ($^{238}/_{219}$) levels.

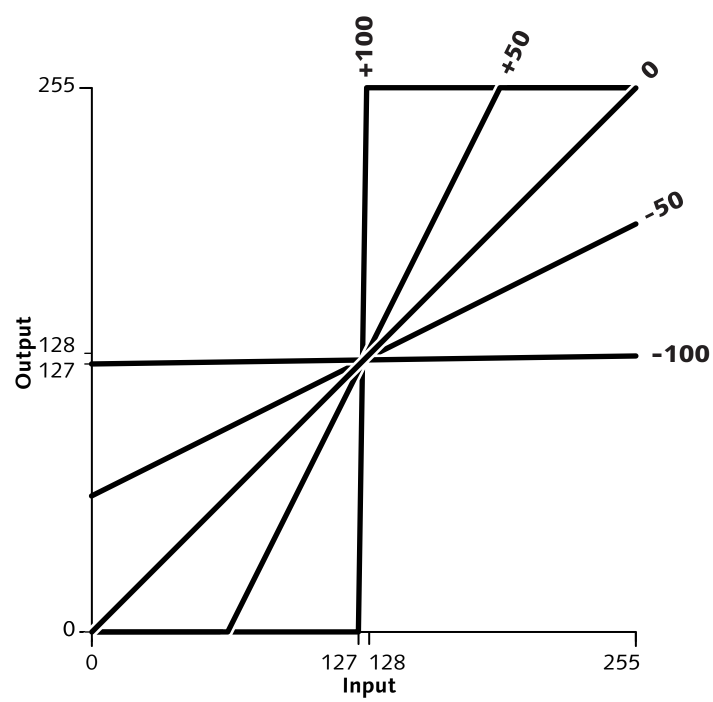


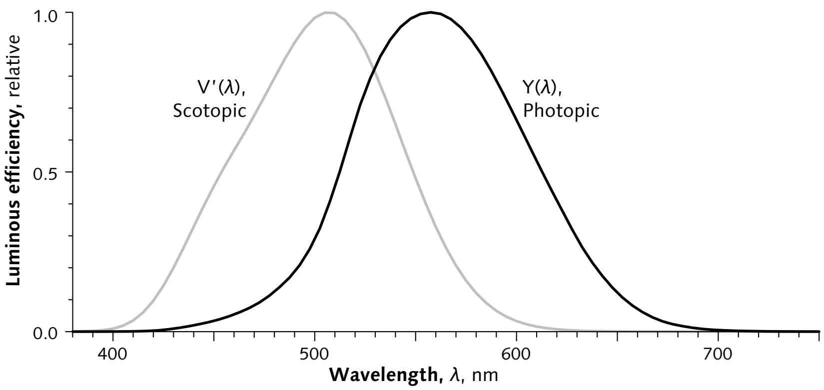

**Contrast (or Gain) control in video** applies a gain factor between roughly 0.5 and 2.0 to *R'G'B'* components, saturating if the result falls outside the range allowed for the coding in use.

**Brightness control in Photoshop** applies an offset of –100 to +100 to *R'G'B'* components ranging from 0 to 255, saturating if the result falls outside the range 0 to 255.

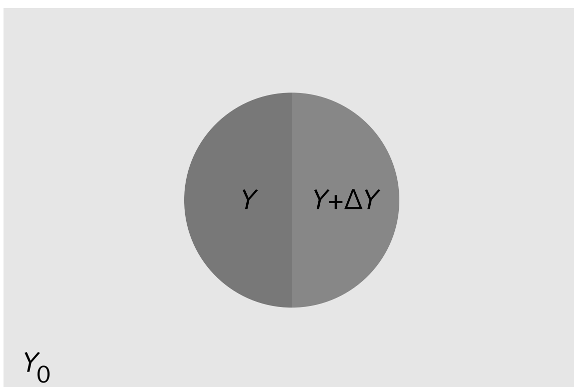



**Contrast control in Photoshop** subtracts 127.5 from the input, applies a gain factor between zero (for –100) and infinity (for +100), then adds 127.5, saturating if the result falls outside the range 0 to 255. This operation is very different from the action of the Contrast control in video.

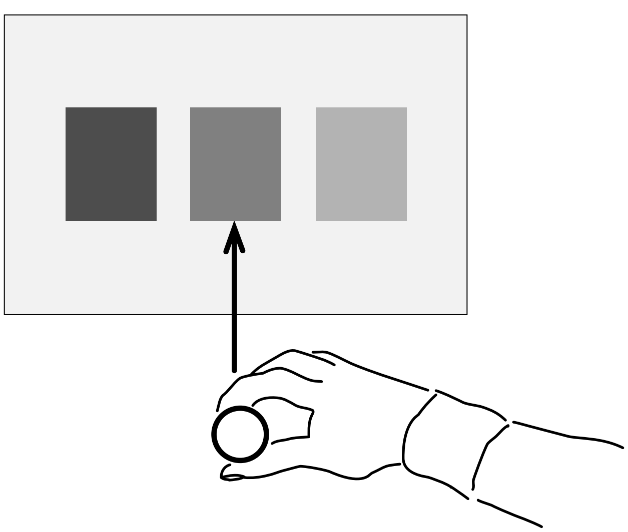





**CIE luminous efficiency function.** Luminance is defined by the CIE as the physical intensity of light, per unit projected area, weighted by the spectral sensitivity of the visual system's lightness sensation. A monochrome scanner or camera must have this spectral response in order to correctly reproduce perceived lightness. The function peaks at about 555 nm. This *analysis* or *spectral sensitivity* function is not comparable to a *spectral power distribution* (SPD). The *scotopic* curve, denoted $V'(\lambda)$ and graphed here in gray, characterizes night vision; it is not useful in image reproduction.

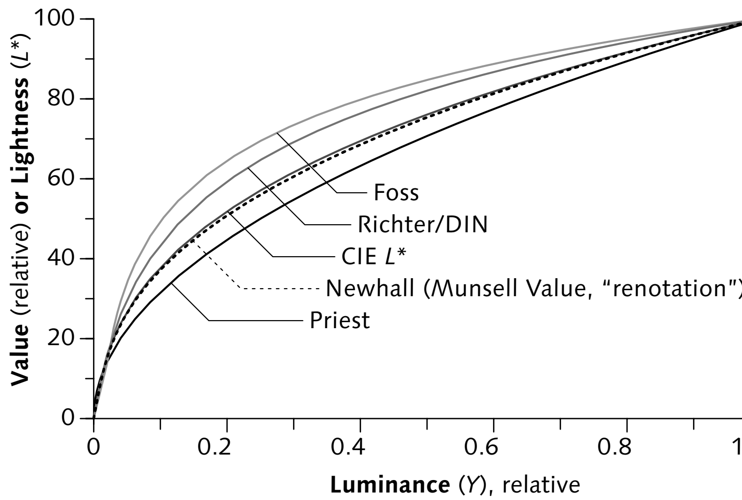



**Contrast sensitivity test pattern** is presented to an observer in an experiment to determine the contrast sensitivity of human vision. The experimenter adjusts $\Delta Y$; the observer reports when he or she detects a difference in lightness between the two halves of the patch. The experiment reveals that the observer cannot detect a difference between luminances when the ratio between them is less than about one percent. Lightness is roughly proportional to the logarithm of luminance.

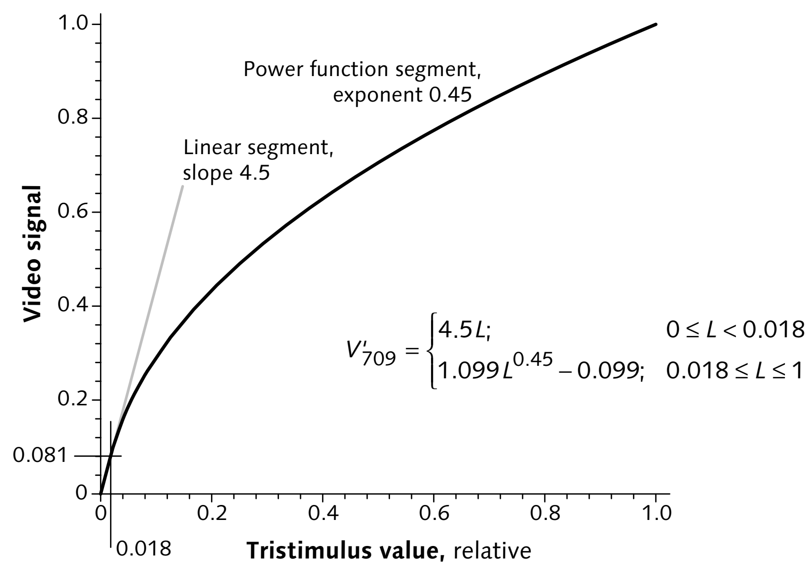


**Lightness Estimation.** This diagram illustrates another experiment to determine the lightness function of human vision. The observer is asked to adjust the lightness of the central patch so that it seems half-way between the lightness of the outside two patches. Measured by this experiment, lightness is approximately proportional to the 0.4-power of luminance. In practice, power functions are generally used instead of logarithmic functions.

**Luminance and lightness.** The relationship between lightness ($L^*$) or value ($V$) and relative luminance $Y$ has been modeled by polynomials, power functions, and logarithms. In all of these systems, 18 percent "mid gray" has lightness about halfway up the perceptual scale. For details, see Fig. 2 (6.3) in Wyszecki and Stiles, *Color Science*.




**Rec. 709 transfer function** is based on a power function with an exponent of 0.45. Theoretically, a pure power function suffices for gamma correction. However, the slope of a pure power function is infinite at zero. In a practical system, such as a television camera, in order to minimize noise in the dark regions of the picture it is necessary to limit the slope (gain) of the function near black. Rec. 709 specifies a slope of 4.5 below a tristimulus value of +0.018. The remainder of the curve is scaled and offset to maintain function and tangent continuity at the breakpoint. Stretching the lower part of the curve also compensates for flare light which is assumed to be present in the viewing environment.




**Ten grayscale patches** are arranged here, from black, with approximately 0% reflectance, to white, with approximately 100% reflectance. In image coding, it is obviously necessary to have a sufficient number of steps from black to white to avoid the boundaries between code values being visible: Obviously, ten steps are not enough. To achieve the fewest number of code values, the luminance (or reflectance) values of each code must be carefully chosen. Ideally, the ratio of luminances from one code to the next would be just on the threshold of visibility. For a contrast ratio of 40:1, typical of a video studio control room, about 100 steps suffice.

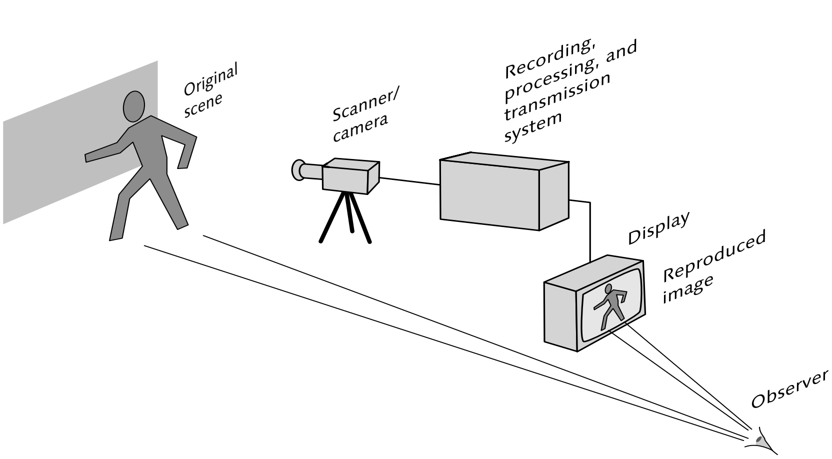
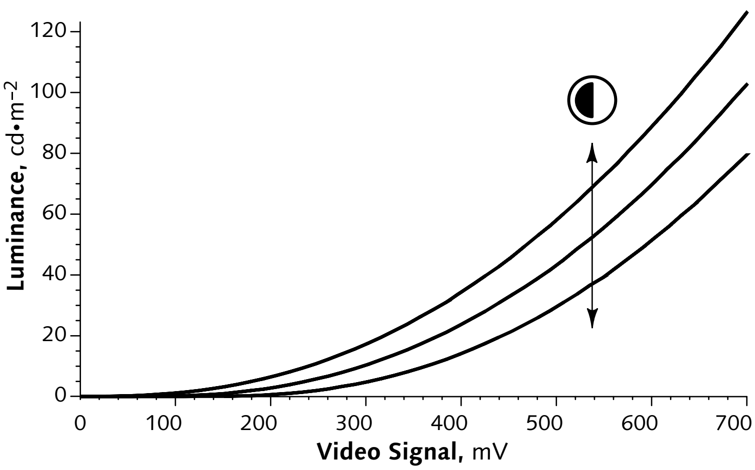
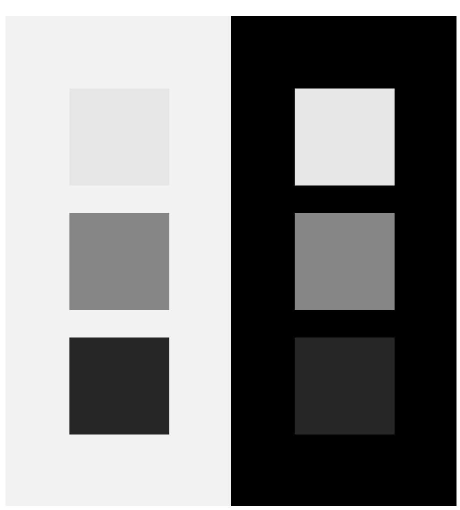


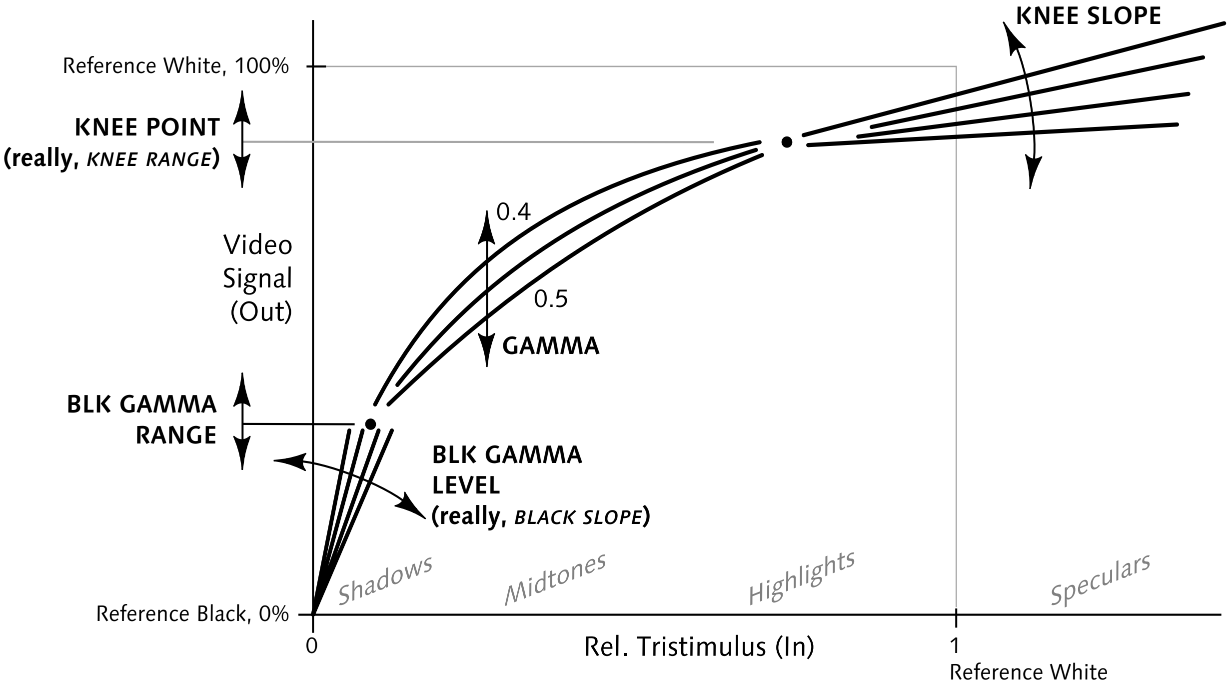

**Image reproduction in video.** Luminance from the scene is reproduced at the display, with a scale factor to account for the difference in overall luminance. However, the ability of vision to detect a luminance difference is not uniform from black to white, but is approximately a constant ratio, about 1 percent. In video, luminance is transformed by a function similar to a square root into a nonlinear, perceptually uniform signal. The camera is designed to mimic the human visual system, in order to "s ee" lightness in the scene the same way that a human observer would. Noise introduced by recording, processing, and transmission then has minimum perceptual impact. The nonlinear signal is transformed back to linear luminance at the display.



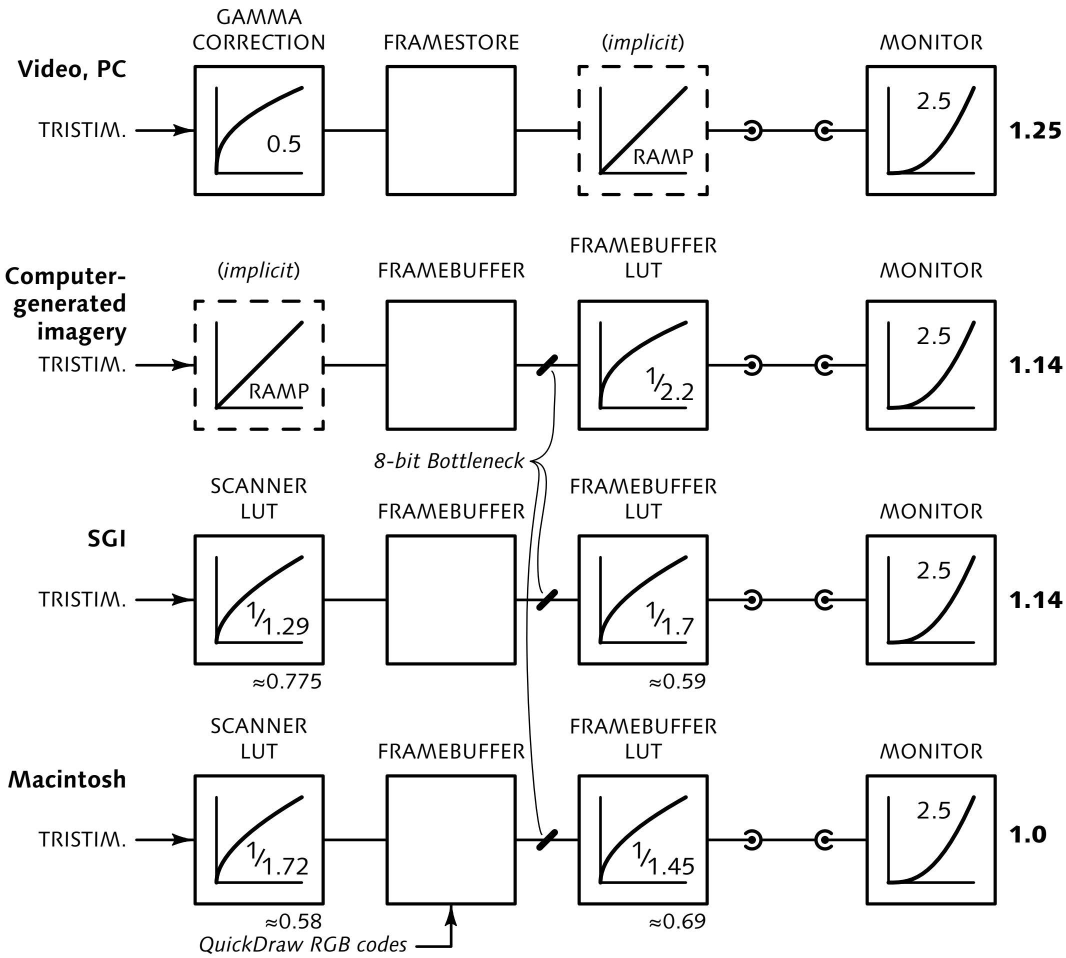

**CRT transfer function** involves a nonlinear relationship between video signal and luminance (or tristimulus value), here graphed for an actual CRT at three different settings of the contrast (or PICTURE) control. Luminance is approximately proportional to input signal voltage raised to the 2.5 power. The *gamma* of a display system – or more specifically, a CRT – is the numerical value of the exponent of the power function.



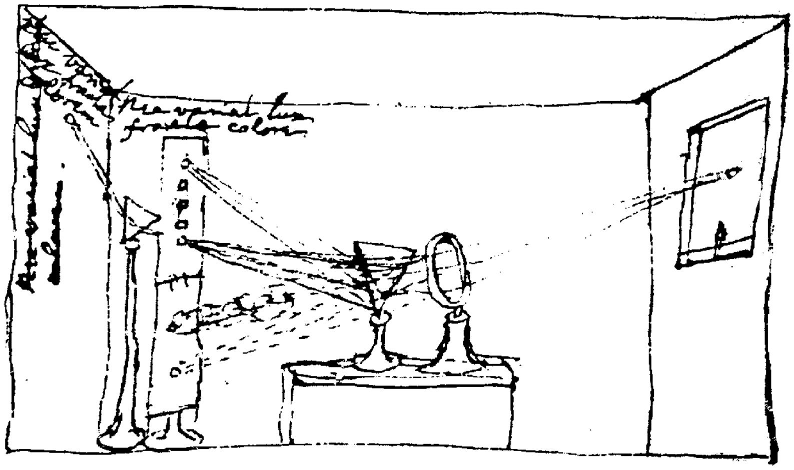
**Surround effect.** The three gray squares surrounded by white are identical to the three gray squares surrounded by black, but the contrast of the black-surround series appears lower than that of the white-surround series. The surround effect has implications for the display of images in dark areas, such as projection of movies in a cinema, projection of 35 mm slides, or viewing of television in your living room. If an image is viewed in a *dark* or *dim surround*, and the relataive luminance of the scene is reproduced correctly, the image will appear to lack contrast. Overcoming this effect requires altering the image data.



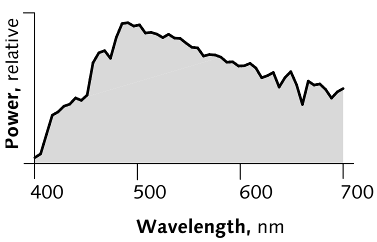

**Camera OETF controls** are manipulated by the cinematographer, to adapt the tone scale of the scene to relative luminance at the display. The slope of the linear segment near black, nominally 4.5, is controlled by BLK GAMMA LEVEL. The linear segment is effective up to video level 0.081 by default; this range is adjustable through the BLK GAMMA RANGE and BLK GAMMA LEVEL controls. In the midtones, the power function exponent, nominally 0.45, is set by GAMMA, typically adjustable from about 0.4 to 0.5. By default, a linear segment is in imposed above reference white. This "knee" region can be set to take effect below 100% video level through adjustment of the KNEE POINT control. Settings below 70% are liable to interfere with skin tone reproduction. Gain in the knee region is controlled by KNEE SLOPE; knee slope should be reduced from its default when it is important to retain a scene's specular highlights.

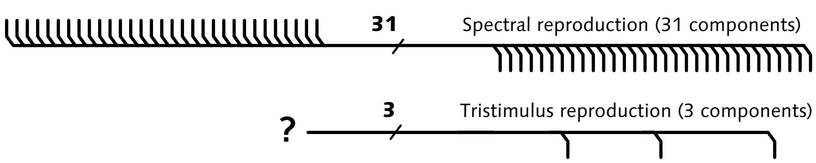



**Gamma in video, computer graphics, SGI, and Macintosh.** In a video system, sketched in the top row, a transfer function that mimics the lightness sensitivity of vision is imposed at the camera. The second row illustrates computer graphics: Calculations are performed in the linear light domain; gamma correction is applied in a lookup table (LUT) at the output of the framebuffer. In SGI computers, a $^{1}/_{1.7}$ power function is loaded into the LUT. Macintosh computers assume that tristimulus values have been raised to the $^{1}/_{1.72}$-power; a $^{1}/_{1.45}$ power function is loaded into the output LUT. The boldface number at the far right indicates the default end-to-end power (rendering) that is applied to tristimulus values from input to output.



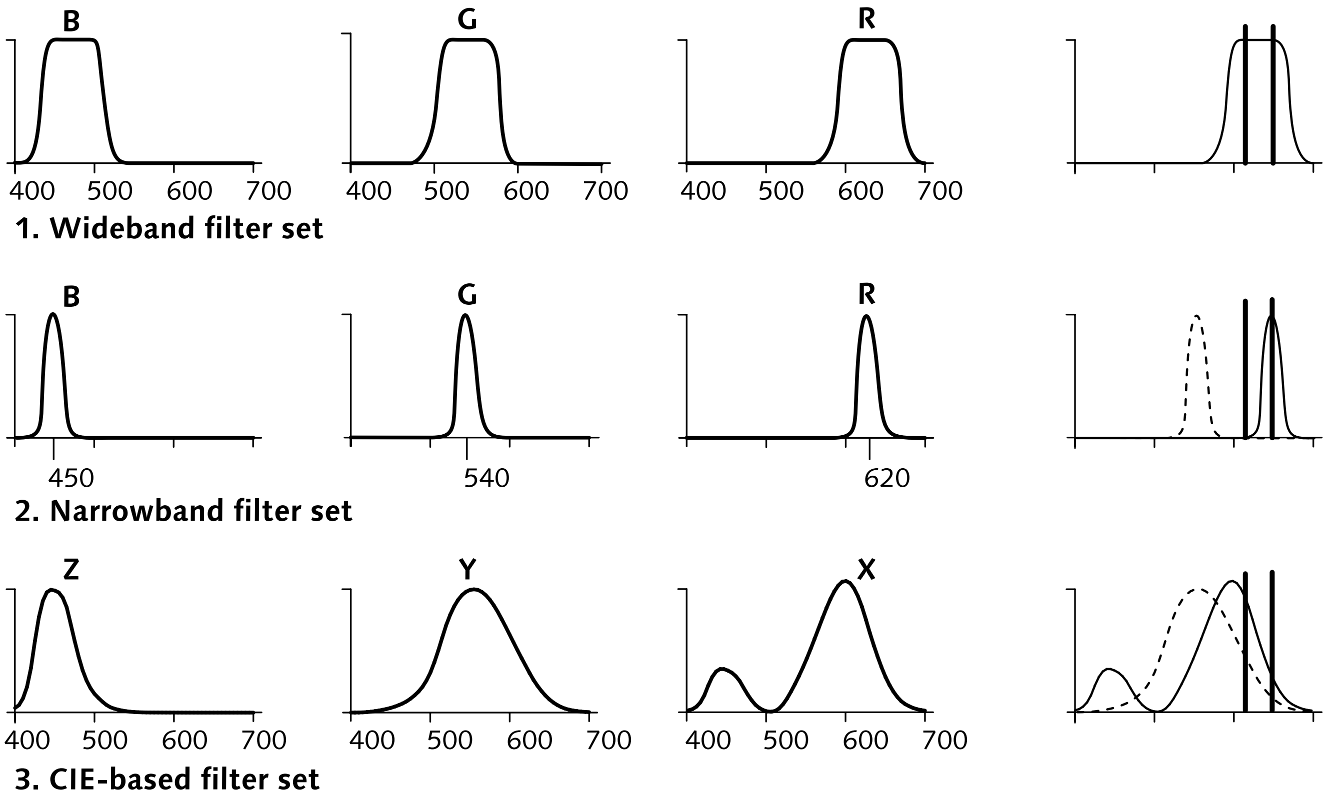
**The visible spectrum** is produced when a prism separates electromagnetic power at wavelengths in the range 400 to 700 nanometers into its spectral components. This experiment was done by Isaac Newton, and documented by his sketch, in Cambridge in about 1666. Light from about 400 to 500 nm appears blue, from 500 to 600 appears green, and from 600 to 700 appears red. The perception of violet arises from wavelengths in the range of 420 nm, but the color purple is not produced by any single wavelength: To stimulate the sensation of purple requires both longwave and shortwave power, with little or no power in medium wavelengths.



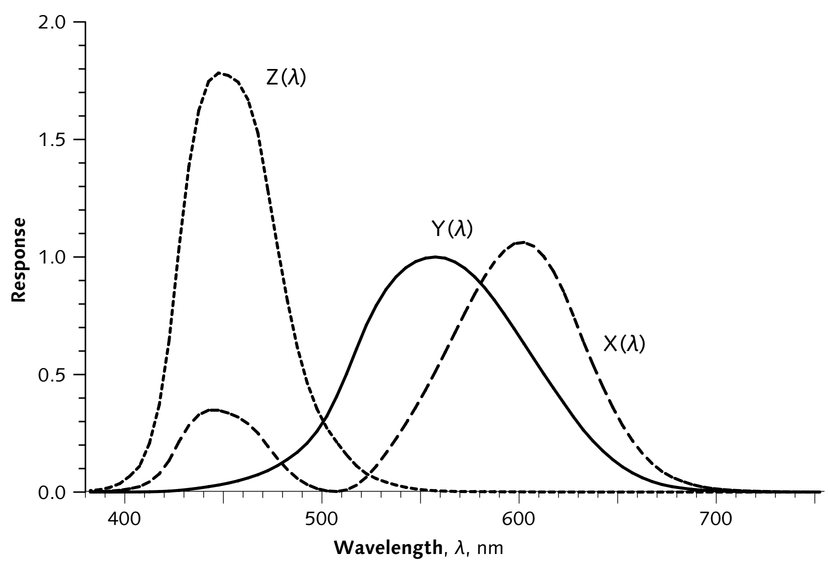

**Tristimulus color reproduction.** A color can be described as a *spectral power distribution* (SPD), perhaps in 31 components representing optical power in 10 nm intervals over the range 400 nm to 700 nm. The SPD shown here is the $D_{65}$ daylight illuminant standardized by the CIE. However, there are exactly three kinds of color photoreceptor (cone) cells in the retina: If appropriate spectral weighting functions are used, three numerical values are necessary and sufficient to describe any color. The challenge is to determine what spectral weighting functions to use.

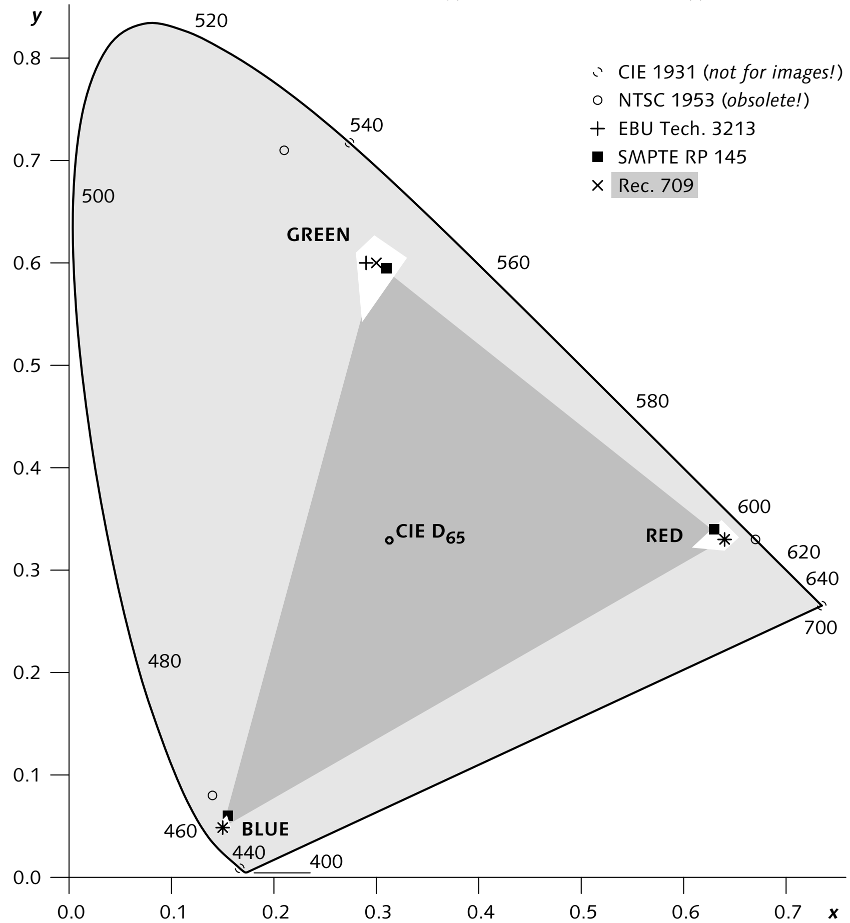



**Scanner spectral constraints** associated with scanners and cameras are shown here. The **wideband filter set** of the top row shows the spectral sensitivity of filters having uniform response across the shortwave, mediumwave, and longwave regions of the spectrum. With this approach, two monochromatic sources seen by the eye to have different colors – in this case, saturated orange and a saturated red – cannot be distinguished by the filter set. **The narrowband filter set** in the middle row solves that problem, but creates another: Many monochromatic sources "fall between" the filters, and are seen by the scanner as black. To see color as the eye does, the three filter responses must be closely related to the color response of the eye. **The CIE-based filter set** in the bottom row shows the *color matching functions* (CMFs) of the CIE Standard Observer.

**CIE color matching functions** were standardized in 1931 by the *Commission Internationale de L'Éclairage* (CIE). These weighting curves map a *spectral power distribution* (SPD) to a *triple* of numerical *tristimulus* components, denoted *X, Y,* and *Z,* that are the mathematical coordinates of color space. Other coordinate systems, such as *RGB*, can be derived from *XYZ*. A camera must have these spectral response curves, or linear combinations of them, in order to capture all colors. However, practical considerations make this difficult. (Though the CMFs are graphed similarly to to spectral power distributions, beware! CMFs analyse SPDs into three color components; they are not comparable to SPDs, which are used to synthesize color.)

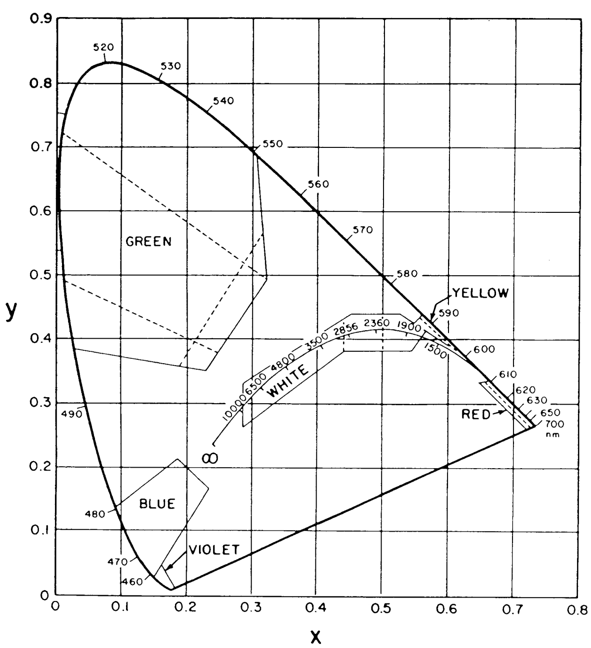


$$x = \frac{X}{X+Y+Z}; \qquad y = \frac{Y}{X+Y+Z}$$




**CIE [*x*, *y*] chromaticity diagram** The *spectral locus* is an inverted U-shaped path traced through [*x*, *y*] coordinates by a monochromatic source as it is tuned from 400 nm to 700 nm. The set of all colors is closed by the *line of purples*, which traces SPDs that combine longwave and shortwave power but have no mediumwave contribution. There is no unique definition of white, but it lies near the center of the chart. All colors lie within the U-shaped region: points outside this region are not associated with colors.

**RGB primaries of video standards** are plotted on the CIE [*x*, *y*] chromaticity diagram. Colors that can be represented in positive *RGB* values lie within the triangle formed by the primaries. Rec. 709 specifies no tolerance. SMPTE tolerances are specified as ±0.005 in *x* and *y*. EBU tolerances are shown as white quadrilaterals; they are specified in *u'*, *v'* coordinates related to the color discrimination of vision. The EBU tolerance boundaries are not parallel to the [*x*, *y*] axes.




**Colors of signal lights** are defined in publication CIE 2.2-1975, *Colours of Signal Lights.* The colors are specified in [x, y] chromaticity coordinates. This is an example of the use of the CIE system outside the domain of image reproduction.

**Color matching functions** (CMFs) of forty nine observers are shown here. (These functions are based upon the CIE monochromatic primaries, at 700 nm, 546.1 nm, and 435.8 nm.) Although it is evident that there are differences among observers, the graph is remarkable for the similarities. The negative excursion of the red component is a consequence of matches being obtained by the addition of white light to the test stimulus. This is Figure 3(5.5.6) from Wyszecki and Stiles' *Color Science*, Second Edition (New York: Wiley, 1982).

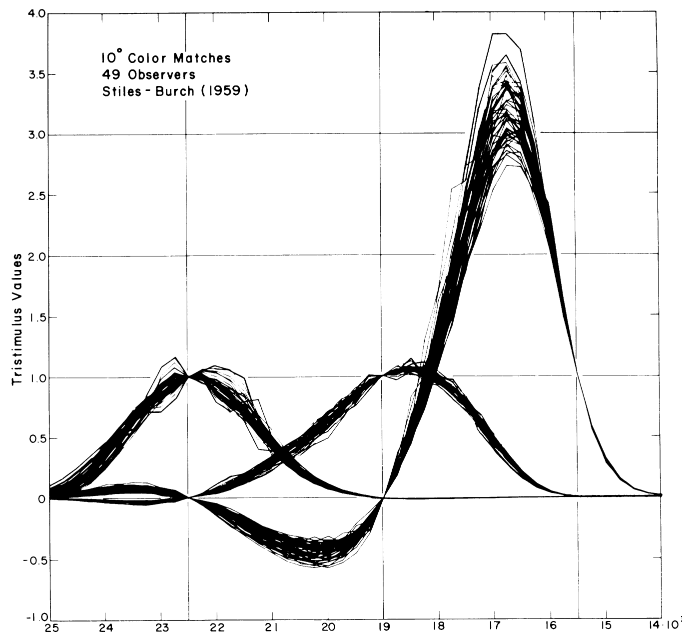
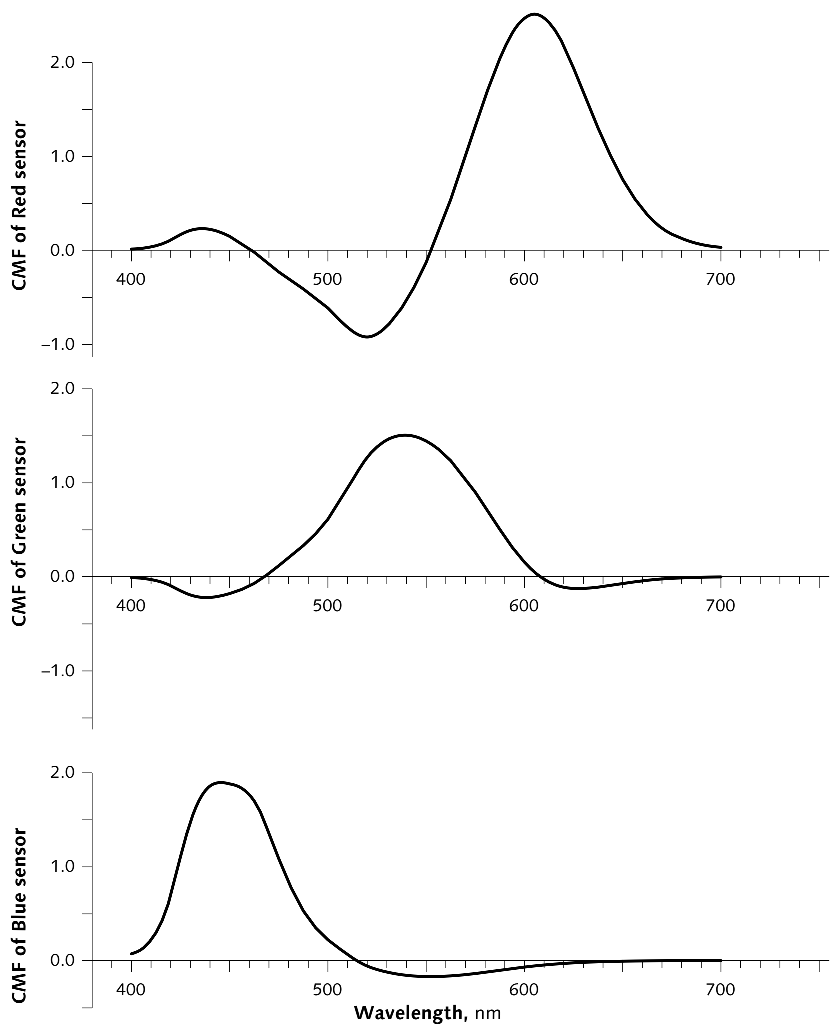
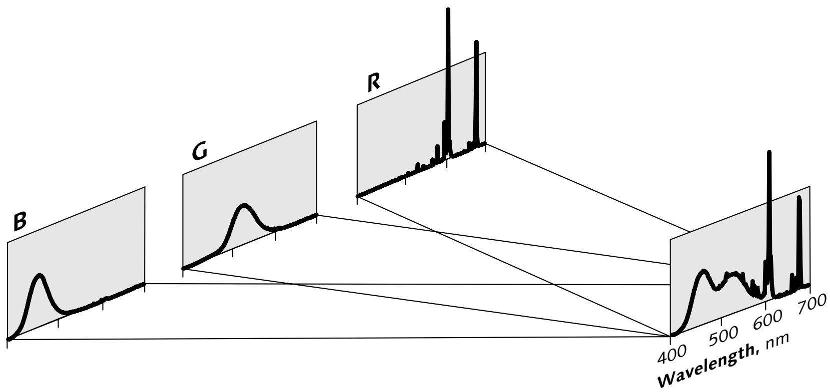


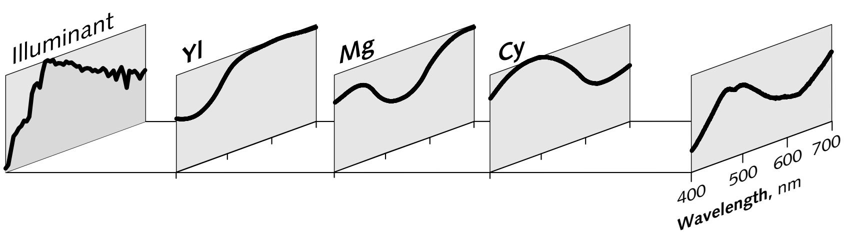

Fig. 3(5.5.6). 10° color-matching functions of 49 observers participating in the Stiles–Burch (1959) experiment. All functions refer to primary stimuli at $m_R = 15{,}500$, $m_G = 19{,}000$, and $m_B = 22{,}500\ \mathrm{cm}^{-1}$.

**CMFs for Rec. 709** are the theoretically correct analysis functions to acquire *RGB* components for display using Rec. 709 primaries. The functions are not directly realizable in a camera or a scanner, due to their negative lobes. But they can be realized through use of the the CIE *XYZ* color matching functions, followed by signal processing involving a 3×3 matrix transform.

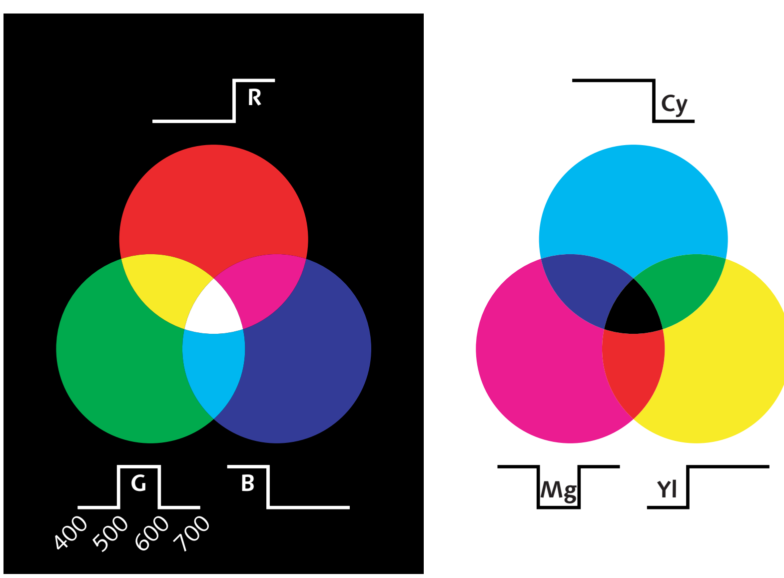





**Additive mixture.** This diagram illustrates the physical process underlying additive color mixture, as is used in color television. Each colorant has an independent, direct path to the image. The spectral power of the image is, at each wavelength, the sum of the spectra of the colorants. The colors of the mixtures are completely determined by the colors of the primaries; analysis and prediction of mixtures is reasonably simple. The SPDs shown here are those of a Sony Trinitron monitor.

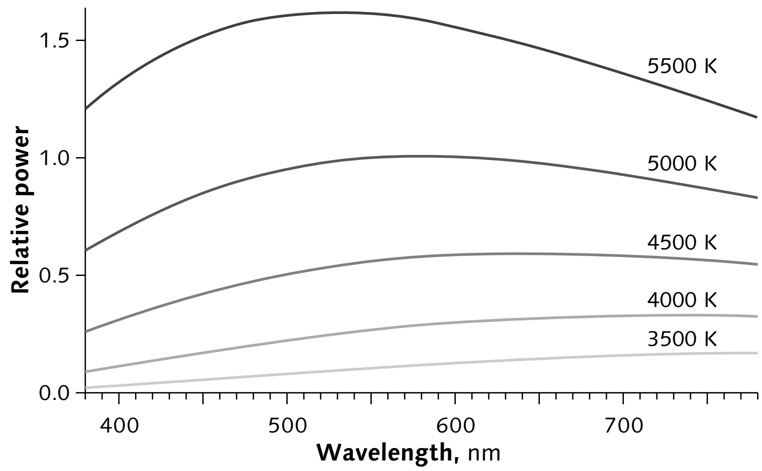



**Subtractive mixture** is employed in color photography and color offset printing. The colorants act in succession to remove spectral power from the illuminant. In physical terms, the spectral power of the mixture is, at each wavelength, the product of the spectrum of the illuminant and the transmission of the colorants: The mixture could be called *multiplicative*. If the amount of each colorant is represented in the form of *spectral optical density* – the base 10 logarithm of the reciprocal of transmission at each wavelength – then color mixtures can be determined by subtraction. Color mixtures in subtractive systems are complex because the colorants absorb power not only in the intended region of the spectrum but also in other regions.

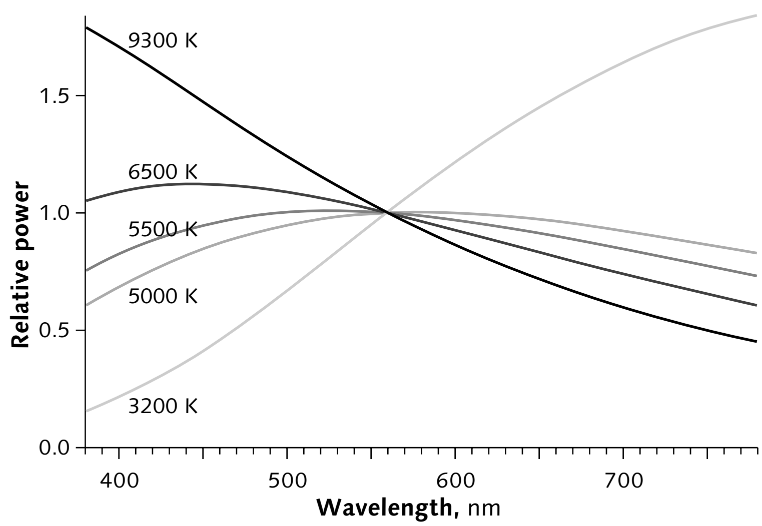



**"One-minus-RGB"** can be used as the basis for subtractive image reproduction. If the color to be reproduced has a blue component of zero, then the yellow filter must attenuate the shortwave components of the spectrum as much as possible. To increase the amount of blue to be reproduced, the attenuation of the yellow filter should decrease. This reasoning leads to the "one-minus-*RGB*" relationships. Cyan in tandem with magenta produces blue, cyan with yellow produces green, and magenta with yellow produces red. A challenge in using subtractive color mixture is that any overlap among the absorption spectra of the colorants results in nonlinear "unwanted absorption" in the mixture.

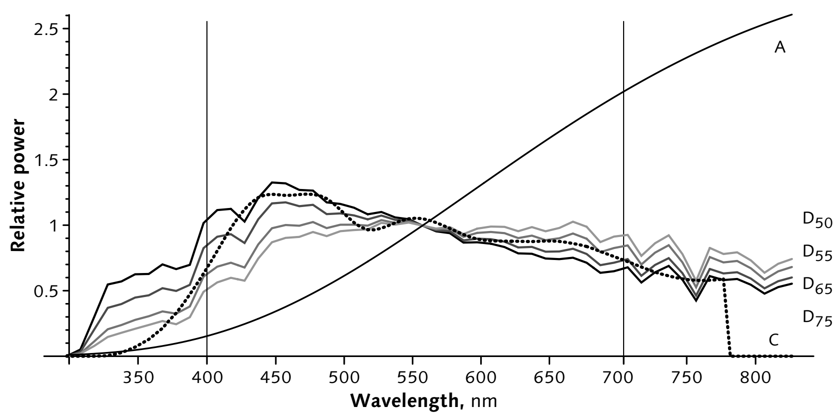
**SPDs of blackbody radiators** at several temperatures are graphed here. Many light sources emit light through heating a metal. Such a source is called a *blackbody radiator.* The spectral power distribution of such a source depends upon absolute temperature: As the temperature increases, the absolute power increases; in addition, the peak of the spectral distribution shifts toward shorter wavelengths.




**SPDs of blackbodies, normalized** to equal power at 560 nm, are graphed here. The dramatically different spectral character of different blackbody radiators is evident. For image capture and for image display, the balance of the red, green, and blue components must be adjusted so as to reproduce the intended color for white.




**CIE illuminants** are graphed here. Illuminant A is an obsolete standard representative of tungsten illumination; its SPD resembles a blackbody radiator at 3200 K. Illuminant C was an early standard for daylight; it too is obsolete. The family of D illuminants represents daylight at several color temperatures.



$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.412453 & 0.357580 & 0.180423 \\ 0.212671 & 0.715160 & 0.072169 \\ 0.019334 & 0.119193 & 0.950227 \end{bmatrix} \bullet \begin{bmatrix} R_{709} \\ G_{709} \\ B_{709} \end{bmatrix}$$

**Transformations between RGB and CIE XYZ.** *RGB* values in a particular set of primaries can be transformed to and from CIE *XYZ* by a 3×3 matrix transform. These transforms involve *tristimulus values*, that is, sets of three linear-light components that conform to the CIE color matching functions. CIE *XYZ* is a special set of tristimulus values. In *XYZ*, every color is represented by an all-positive set of values. SMPTE has standardized a procedure for computing these transformations. To transform from Rec. 709 *RGB* (with its $D_{65}$ white point) into CIE *XYZ*, use this transform. Because white is normalized to unity, the middle row sums to unity.

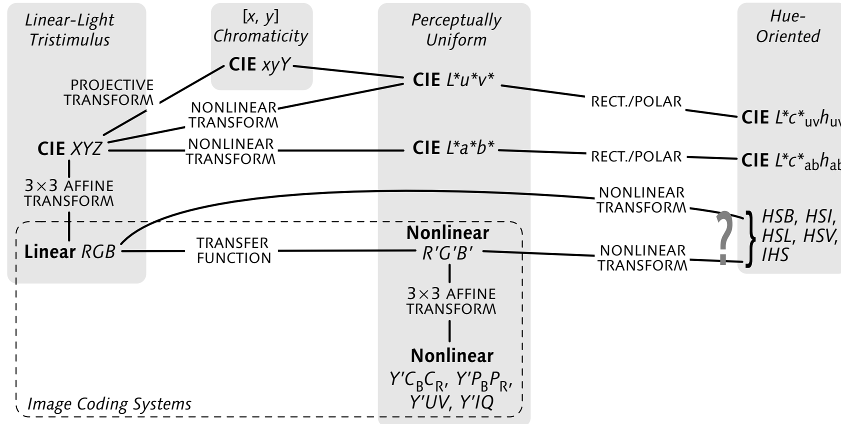
$$\begin{bmatrix} R_{709} \\ G_{709} \\ B_{709} \end{bmatrix} = \begin{bmatrix} 3.240479 & -1.537150 & -0.498535 \\ -0.969256 & 1.875992 & 0.041556 \\ 0.055648 & -0.204043 & 1.057311 \end{bmatrix} \bullet \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

**Transforms from CIE XYZ to RGB.** To transform from CIE *XYZ* into Rec. 709 *RGB*, use this transform. This matrix has some negative coefficients: *XYZ* colors that are *out of gamut* for Rec. 709 *RGB* transform to *RGB* components where one or more components are negative or greater than unity.

$$\begin{bmatrix} R_{709} \\ G_{709} \\ B_{709} \end{bmatrix} = \begin{bmatrix} 0.939555 & 0.050173 & 0.010272 \\ 0.017775 & 0.965795 & 0.016430 \\ -0.001622 & -0.004371 & 1.005993 \end{bmatrix} \bullet \begin{bmatrix} R_{145} \\ G_{145} \\ B_{145} \end{bmatrix}$$

**Transforms among RGB systems.** *RGB* values in a system employing one set of primaries can be transformed to another set by a 3×3 linear-light matrix transform. Generally these matrices are normalized for a white point luminance of unity. This is the transform from SMPTE RP 145 *RGB* to Rec. 709 *RGB.* Transforming among *RGB* systems may lead to an *out of gamut RGB* result, where one or more *RGB* components are negative or greater than unity.

**A variety of color systems** can be classified into four groups that are related by different kinds of transformations. The systems useful for color specification are all based on CIE *XYZ*. A color specification system needs to be able to represent any color with high precision. Since few colors are handled at a time, a specification system can be computationally complex. For image coding, the strict relationship to the CIE system can be relaxed somewhat, and efficiency is important. *Tristimulus systems* and *perceptually uniform systems* are useful for image coding.



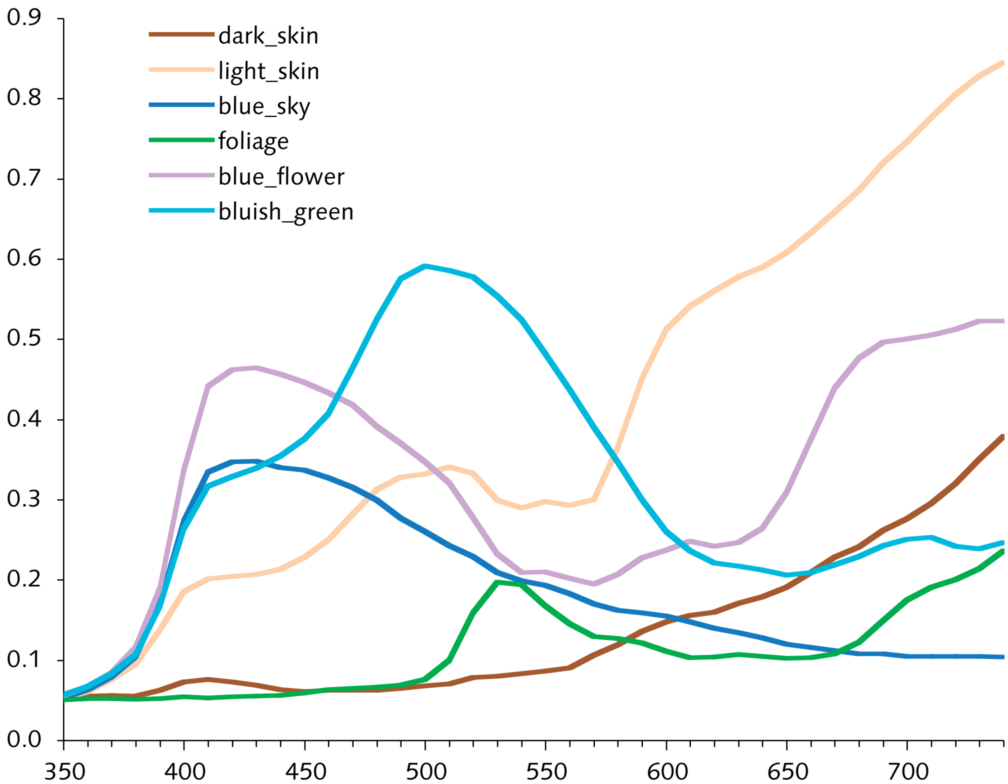



Figure 1 **Macbeth chart spectra, top row.**

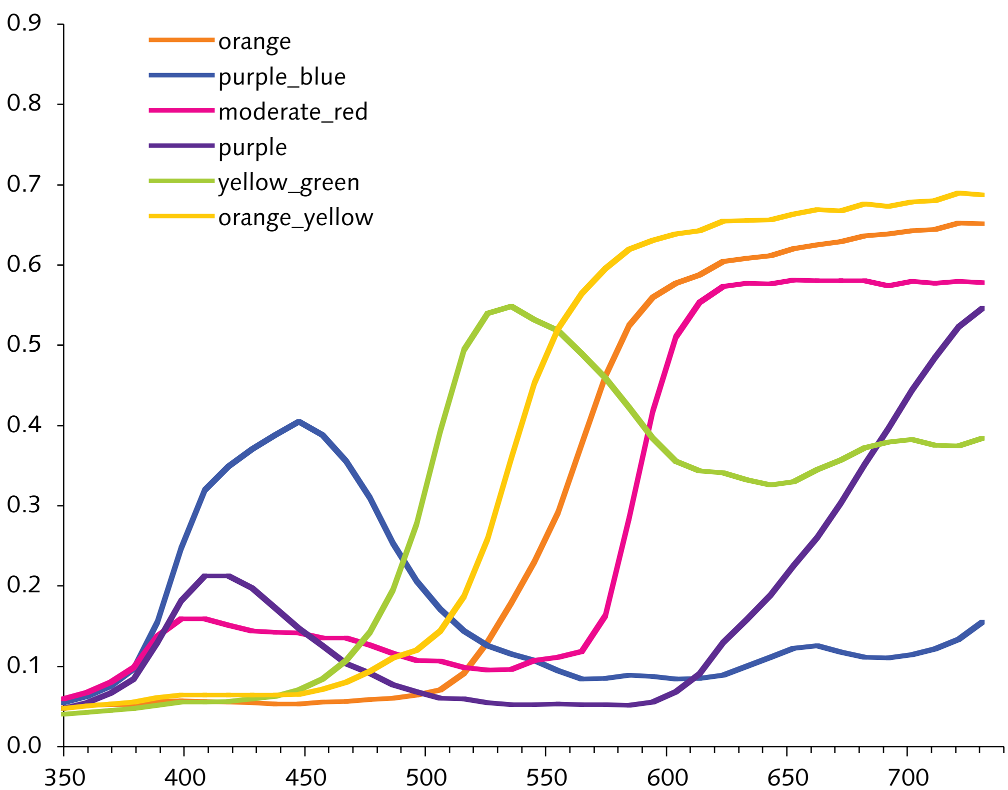



Figure 2 **Macbeth chart spectra, second row.**

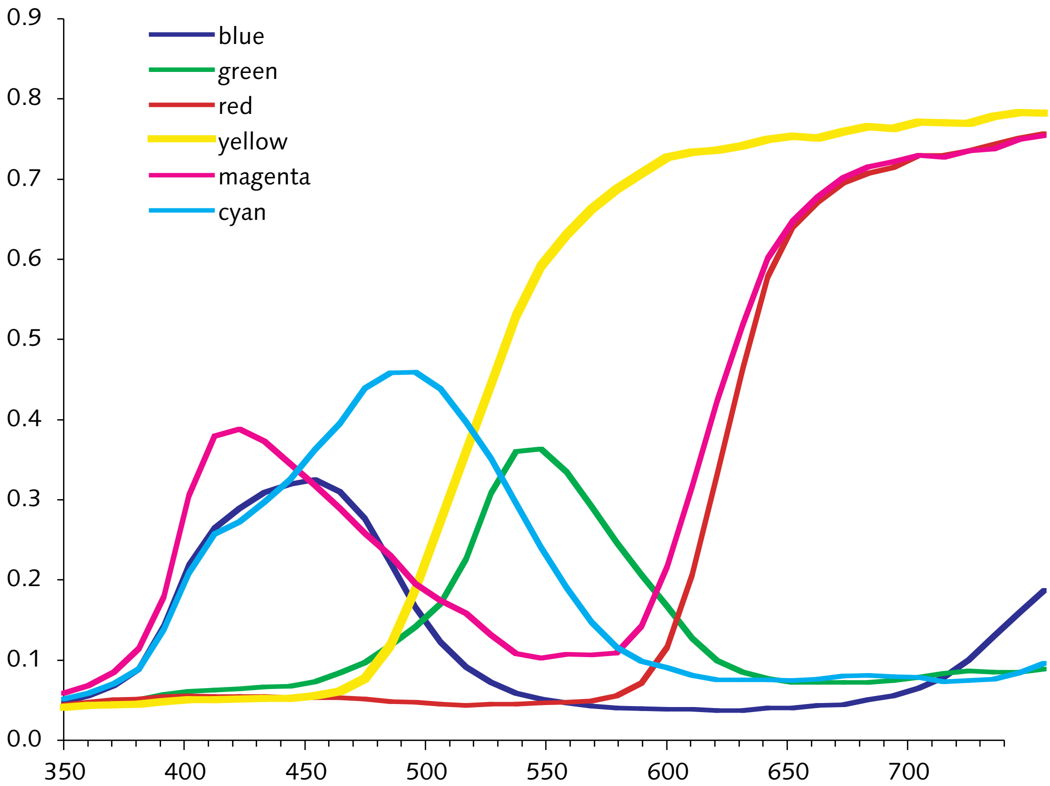



Figure 3 **Macbeth chart spectra, third row.**

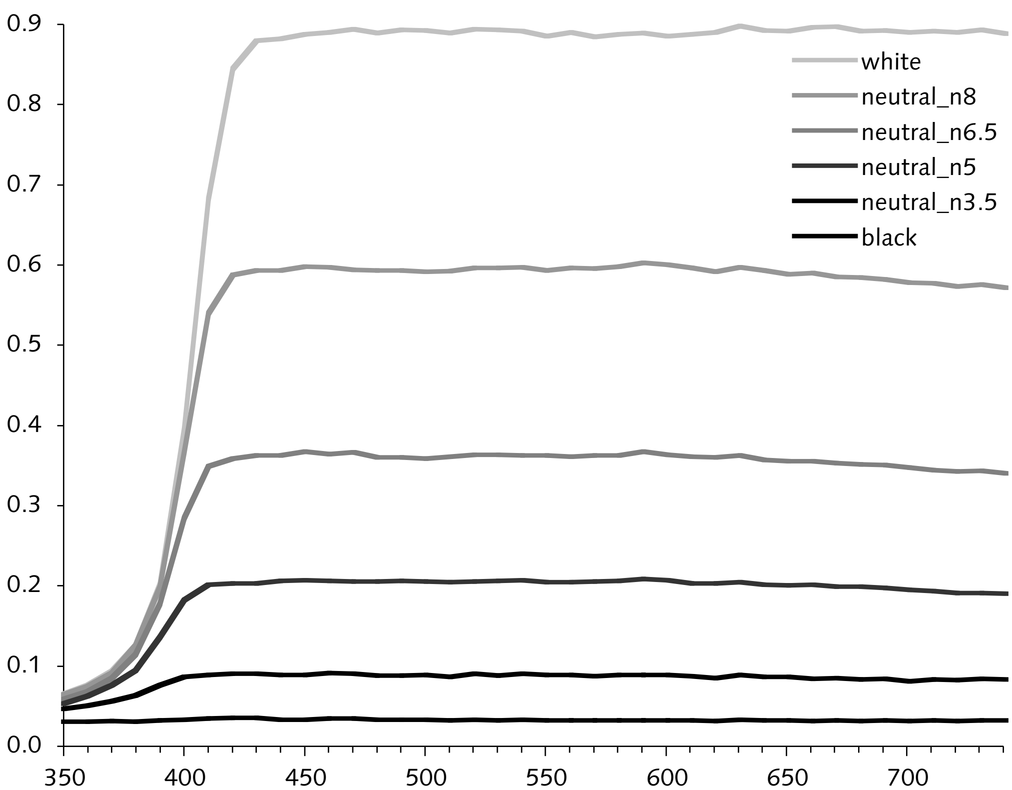
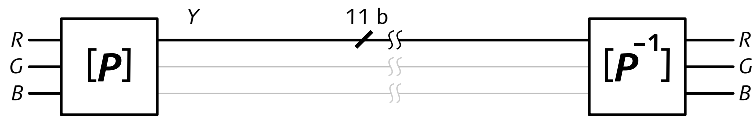
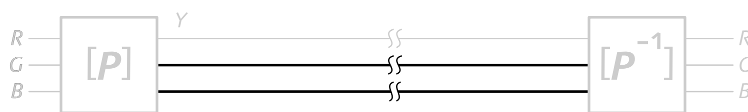


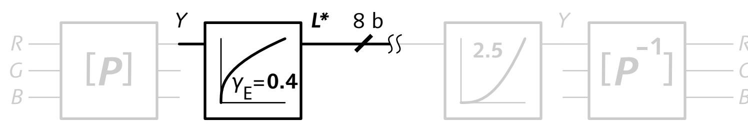

Figure 4 **Macbeth chart spectra, bottom row** (neutral series)



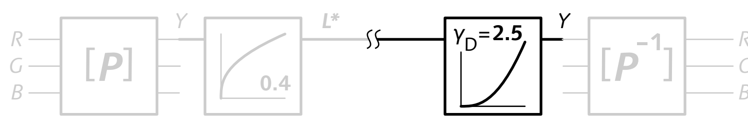

**Ideally.** a video system would compute true, CIE luminance as a properly-weighted sum of linear *R, G,* and *B* tristimulus components (each proportional to intensity). At the decoder, the inverse matrix would reconstruct the linear *R, G,* and *B* components.



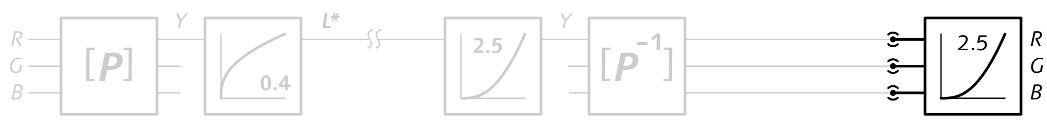
**Two color difference components** are computed, to enable chroma subsampling. Disregard these for now: No matter how the color difference signals are coded in this idealized system, all of the true (CIE) luminance is conveyed through the monochrome channel.



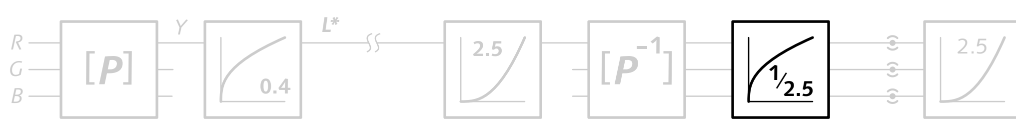

**Nonlinear coding** of luminance involves the application of a transfer function roughly similar to the lightness sensitivity of human vision – that is, roughly similar to the CIE *L** function. This permits the use of 8-bit quantization.



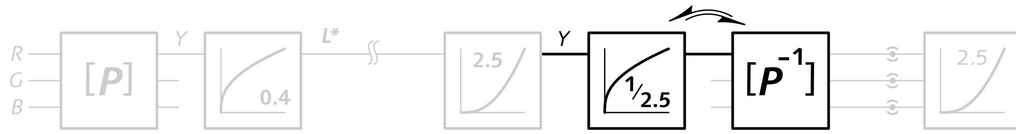

**At the decoder.** the inverse transfer function is applied. If a video system were to operate in this manner, it would be said to exhibit the *constant luminance principle:* All of the true (CIE) luminance would be conveyed by – and recoverable from – the lightness component.



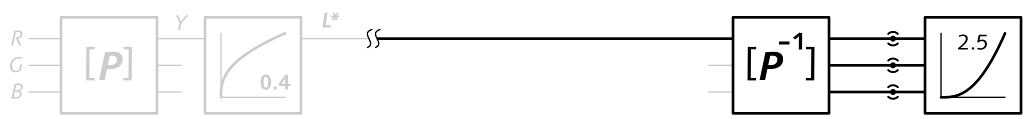

**The electron gun** of a CRT introduces a power function having an exponent between about 2.35 and 2.55. In a constant luminance system, this would have to be compensated.




**Correction for the monitor's** power function would require insertion of a compensating transfer function – roughly a 0.4 power function – at the decoder (or in the monitor). This would be expensive and impractical. Notice that the decoder would include two transfer functions, with powers 0.4 (approximately) and 2.5 – the functions are inverses! These near-inverses would cancel, but the matrix is in the way. It is tempting to rearrange the block diagram to combine them!



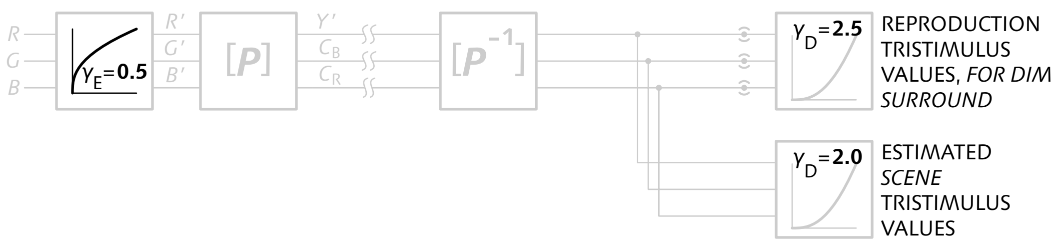

**To avoid the complexity** of building into a decoder both 2.5- and 0.4-power functions, we rearrange the block diagram to interchange the order of the decoder's matrix and transfer function. The inverse *L** function and the 0.4-power function are nearly inverses of each other. The combination of the two has no net effect; the pair can be dropped from the decoder. The decoder no longer operates on, or has direct access to, linear-light signals.




**The decoder** now comprises just the inverse of the encoder matrix, and the 2.5-power function that is intrinsic to the CRT.


**Rearranging the decoder**requires that the encoder is also rearranged, so as to mirror the operations of the decoder. First, the linear *RGB* components are subject to gamma correction. Then, gamma-corrected *R'G'B'* components are matrixed. When decoded, physical intensity is reproduced correctly; however, true (CIE) luminance is no longer computed at the encoder. Instead, a nonlinear quantity *Y'*, loosely representative of luminance, is computed and transmitted. I call the nonlinear quantity *luma*. (Many television engineers mistakenly call the nonlinear quantity *luminance* and assign to it the symbol *Y*. This leads to great ambiguity and confusion.)

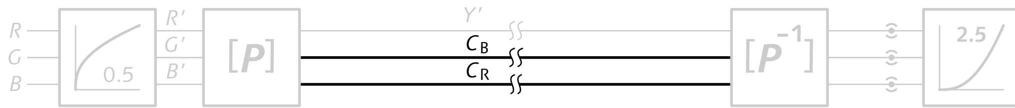

**When viewing** a reproduced image, the viewer invariably prefers a reproduction whose contrast ratio has been stretched slightly to a reproduction that is physically correct. The subjective preference depends somewhat upon the viewing environment. For television, an end-to-end power function with an exponent of about 1.25 should be applied, to produce pictures that are subjectively correct. This correction could be applied at the decoder.



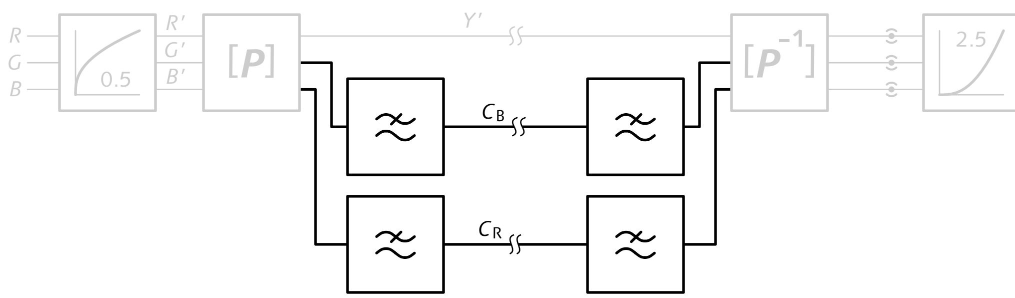

**Rather than introducing** circuitry that implements a power function with an exponent of about 1.25 at the decoder, we modify the encoder to apply approximately a 0.5-power, instead of the physically-correct 0.4 power. Consider the subjective rendering as being accomplished at the display: The image coding is accomplished assuming that a 2.0-power function relates the coded signals to scene tristimulus values.

| *Imaging system* | *Encoding exponent* | *“Advertised” exponent* | *Decoding exponent* | *Typ. Surround* | *End-to-end exponent* |
|---|---|---|---|---|---|
| Cinema | 0.6 | 0.6 | 2.5 | Dark | 1.5 |
| Television (Rec. 709) | 0.5 | 0.45 | 2.5 | Dim | 1.25 |
| Office (sRGB) | 0.45 | 0.42 | 2.5 | Light | 1.125 |

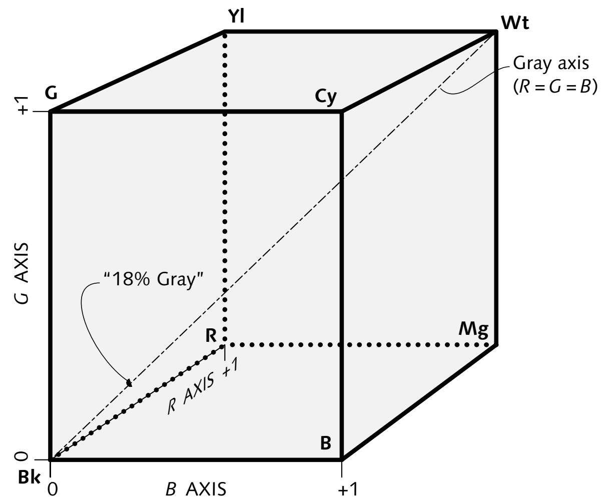
**End-to-end power functions** for several imaging systems. The encoding exponent achieves approximately perceptual coding. (The “advertised” exponent neglects the scaling and offset associated with the straight-line segment of encoding.) The decoding exponent acts at the display to approximately invert the perceptual encoding. The product of the two exponents sets the end-to-end power function that imposes the required rendering.



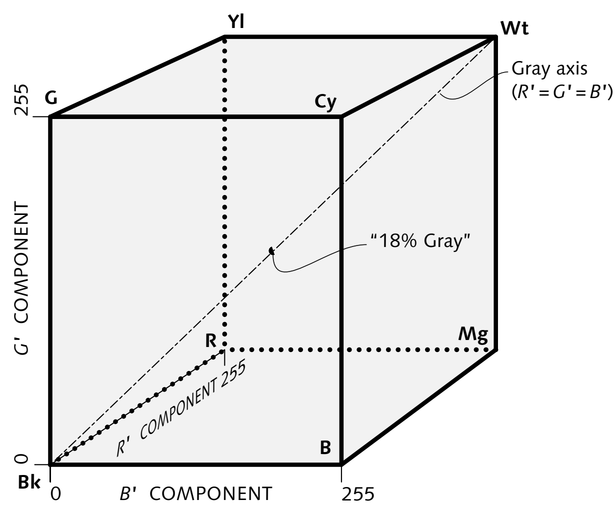

**Color difference** components are transmitted from the encoder to the decoder. In an ideal constant luminance decoder, no matter how the color difference signals are treated, all of the true, CIE luminance is present in the luminance channel. But with the rearranged block diagram, although most CIE luminance information is conveyed through the *Y′* component, some true luminance "leaks" into the color difference components. If color difference subsampling were not used, this would not present a problem.



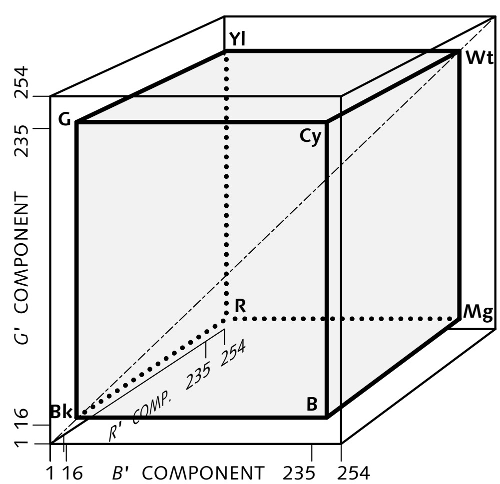

**Subsampling of color difference** components allows color video signals to be conveyed efficiently. In a true constant luminance system, the subsampling would have no impact on the true luminance signal. But with the modified block diagram of non-constant luminance coding, in addition to removing detail from the color components, subsampling removes detail from the "leaked" luminance. This introduces luminance reproduction errors, whose magnitude is noticable but not objectionable in normal scenes: In areas where detail is present in saturated colors, relative luminance is reproduced too low.




***RGB* cube.** Red, green, and blue tristimulus primary components, proportional to intensity, can be considered to be the coordinates of a three-dimensional color space. Coordinate values between zero and unity define the unit cube of this space. The drawback of conveying *RGB* components of an image is that each component requires relatively high spatial resolution: Transmission or storage of a color image using *RGB* components requires a channel capacity three times that of a grayscale image.



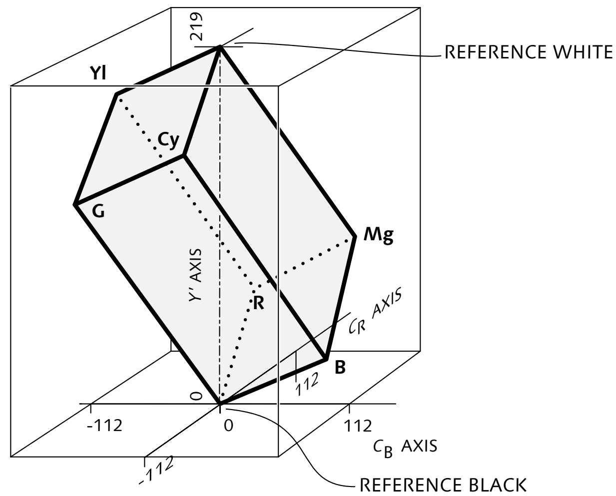

***R'G'B'* cube** represents nonlinear (gamma corrected) *R'G'B'*, typical of computer graphics, JPEG, and video. Though superficially similar to the linear-intensity *RGB* cube, it is dramatically different in practice, because the *R'G'B'* values are perceptually uniform.



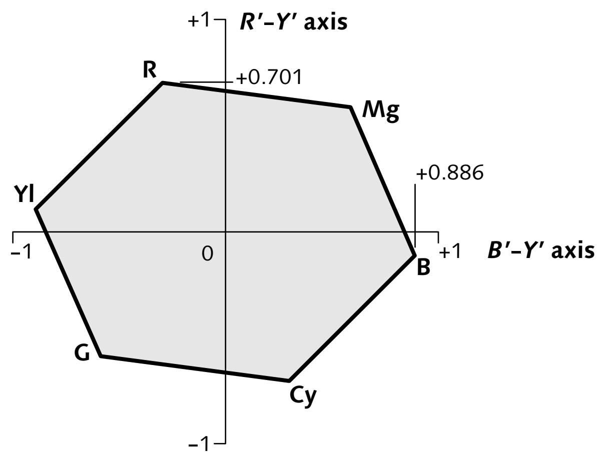

***R'G'B'* in studio video** includes headroom and footroom to accommodate transients that result from analog and digital filtering. In video signal processing, black is typically coded at zero; undershoots are coded in the signal domain between –16 and –1. An offset of +16 is applied at the interface; at the interface, footroom extends from code 0 to code 15. In video signal processing, reference white is coded at +219; the interface offset places the headroom region between codes 236 and 255.

***R'G'B'* cube, transformed** to *Y'*, *B'*-*Y'*, *R'*-*Y'* coordinates. Human vision has considerably less spatial acuity for color than for brightness. As a consequence of the poor color acuity of vision, a color image can be coded into a wide-band luma component *Y'*, and two color difference components from which luma has been removed by subtraction. Each color difference component can then be filtered to have substantially less spatial resolution than lightness. Green dominates luma: Between 60 and 70 percent of lightness comprises green information, so it is sensible – and advantageous for signal-to-noise reasons – to base the color difference signals on the other two primaries.

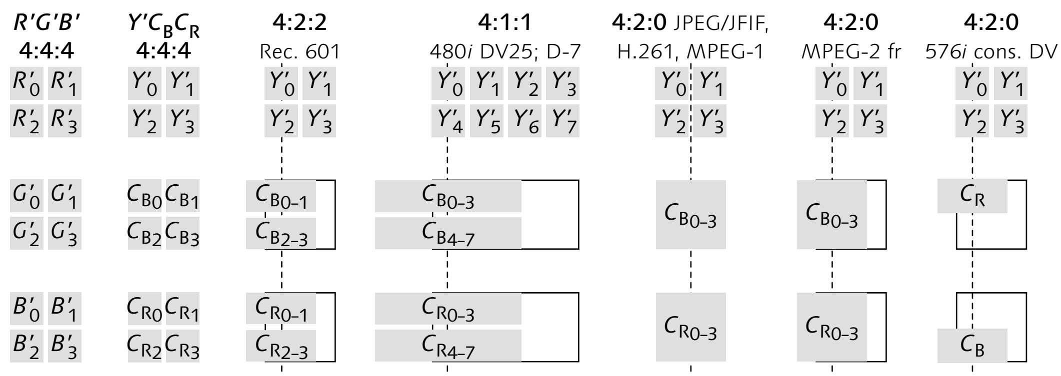



***B'*–*Y'*, *R'*–*Y'* components.** The extrema of *B'*–*Y'* occur at yellow and blue, at values ±0.886. The extrema of *R'*–*Y'* occur at red and cyan, at values ±0.701. These are inconvenient values for both digital and analog systems. The systems $Y'P_BP_R$, $Y'C_BC_R$, and *Y'UV* all employ versions of (*Y'*, *B'*–*Y'*, *R'*–*Y'*) that are scaled to place the extrema of the component signals at more convenient values.

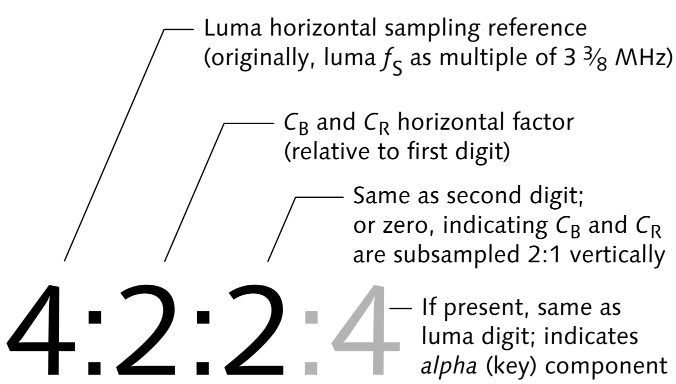



**Luma and *B'*-*Y'*, *R'*-*Y'* encoding matrix.** To obtain (*Y'*, *B'*–*Y'*, *R'*–*Y'*), from R'G'B', for Rec. 601 luma, use this matrix transform. The numerical values used here, and to follow, are based on the Rec. 601 luma coefficients. Unfortunately, SMPTE and ATSC have – for no good technical reason – chosen different coefficients for HDTV. All of the associated equations and scale factors are different.

$$\begin{bmatrix} {}^{601}Y' \\ B'-{}^{601}Y' \\ R'-{}^{601}Y' \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.299 & -0.587 & 0.886 \\ 0.701 & -0.587 & -0.114 \end{bmatrix} \bullet \begin{bmatrix} R' \\ G' \\ B' \end{bmatrix}$$




**Chroma subsampling.** Providing full luma detail is maintained, vision's poor color acuity enables color detail to be reduced by subsampling. A 2×2 array of *R'G'B'* pixels is transformed to a luma component *Y'* and two color difference components $C_B$ and $C_R$. The color difference components are then filtered (averaged). Here, $C_B$ and $C_R$ samples are drawn wider or taller than the luma samples to indicate their spatial extent. The horizontal offset of $C_B$ and $C_R$ is due to cositing. (In 4:2:0 in JPEG/JFIF, MPEG-1, and H.261, chroma samples are not cosited, but are sited *interstitially*.)




**Chroma subsampling notation** indicates, in the first digit, the relative horizontal sampling rate of luma. (The digit *4* is a historical reference to four times 3 3⁄8 MHz, approximately four times the color subcarrier of NTSC.) The second digit specifies the horizontal subsampling of $C_B$, with respect to luma. The third digit was intended to reflect the horizontal subsampling of $C_R$. The designers of the notation did not anticipate vertical subsampling, and the third digit has now been subverted to that purpose: A third digit of zero denotes that $C_B$ and $C_R$ are subsampled vertically by a factor of two. An optional fourth digit signifies an *alpha* (key, or opacity) component.

**Luminance** ($Y$) can be computed by forming a weighted sum of linear (tristimulus) red, green, and blue primary components, where $R$, $G$, and $B$ are formed from appropriate spectral weighting functions. The coefficients for the primaries of Rec. ITU-R BT.709 ("Rec. 709"), representative of modern video and computer graphics equipment, are indicated in this equation. Unfortunately, the word *luminance* and the symbol $Y$ are often used mistakenly to refer to *luma*; when you see that term or symbol used, you should determine exactly what is meant.

**CIE Luminance:**

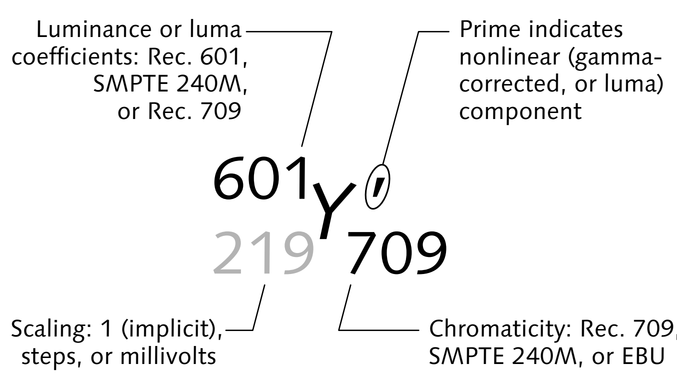
$$^{709}Y = 0.2126\,R + 0.7152\,G + 0.0722\,B$$

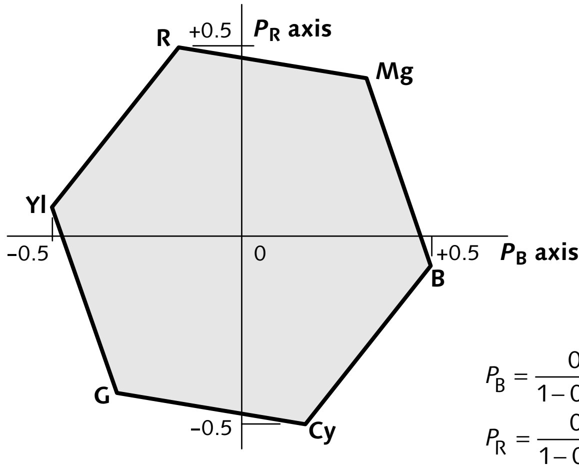
**Luma** refers to a nonlinear quantity that is used to represent lightness in a video system. A nonlinear transfer function – *gamma correction* – is applied to each of the linear (tristimulus) $R$, $G$, and $B$ components. Then a weighted sum of the nonlinear components is computed to form *luma*, denoted $Y'$. Luma is roughly perceptually uniform. Conventional television systems form luma according to the coefficients standardized in Rec. ITU-R BT.601. Many television engineers use the word *luminance* to refer to this nonlinear quantity, and omit the prime symbol that denotes the nonlinearity. But luma is not comparable to CIE luminance; in fact, it cannot even be computed from CIE luminance.

**Video Luma:**

$$\begin{aligned} ^{601}Y' &= 0.299\,R' + 0.587\,G' + 0.114\,B' \\ &\approx 0.299\sqrt{R} + 0.587\sqrt{G} + 0.114\sqrt{B} \end{aligned}$$

**Luma notation** became necessary when different chromaticities, different luma coefficients, and different scalings were introduced to luminance and luma. The subscript denotes the chromaticities of the primaries. An unprimed $Y$ indicates true CIE luminance (as a weighted sum of linear-intensity $R$, $G$, and $B$). A prime symbol (′) indicates *luma*, formed as a weighted sum of gamma-corrected $R'$, $G'$, and $B'$. The leading superscript indicates the weights used to compute luma or luminance; historically, the weights standardized in Rec. 601 were used, but HDTV standards use different weights. The leading subscript indicates the overall scaling of the signal; if omitted, an overall scaling of unity is implicit.



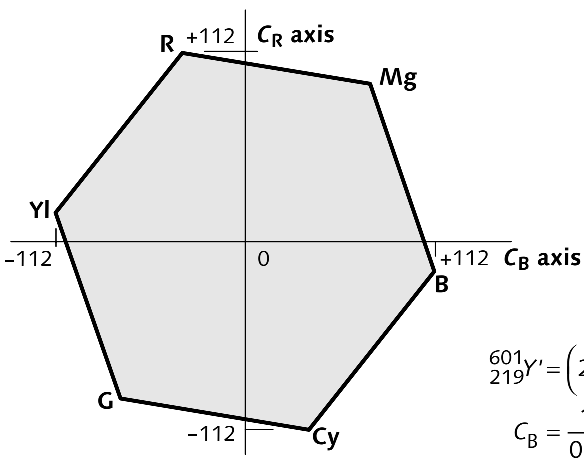



$$P_B = \frac{0.5}{1-0.114}\left(B' - {}^{601}Y'\right)$$

$$P_R = \frac{0.5}{1-0.299}\left(R' - {}^{601}Y'\right)$$

**$P_BP_R$ components.** If two color difference components are to be formed having identical unity excursions, then $P_B$ and $P_R$ color difference components are used. For Rec. 601 luma, these equations are used. The scale factors, sometimes written 0.564 and 0.713, are chosen to limit the excursion of *each* color difference component to the range −0.5 to +0.5 with respect to unity luma excursion: 0.114 in the first expression is the luma coefficient of blue, and 0.299 in the second is for red. In SMPTE standards for component analog, the luma signal ranges from 0 mV (black) to 700 mV (white), and $P_B$ and $P_R$ signals range ±350 mV.

.



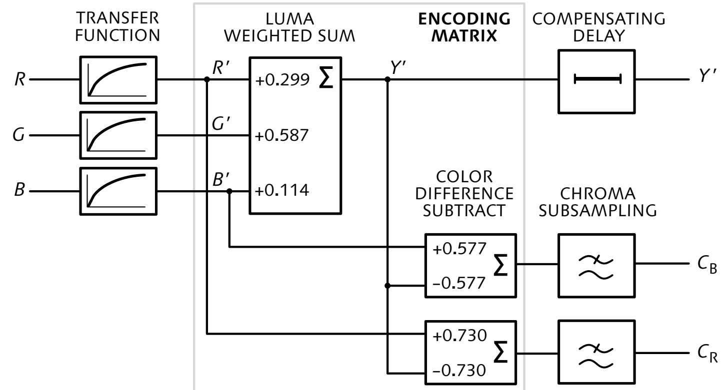

$${}^{601}_{219}Y' = \left(219 \cdot {}^{601}Y'\right) \cdot 2^{k-8}$$

$$C_B = \frac{112}{0.886}\left(B' - {}^{601}Y'\right) \cdot 2^{k-8}$$

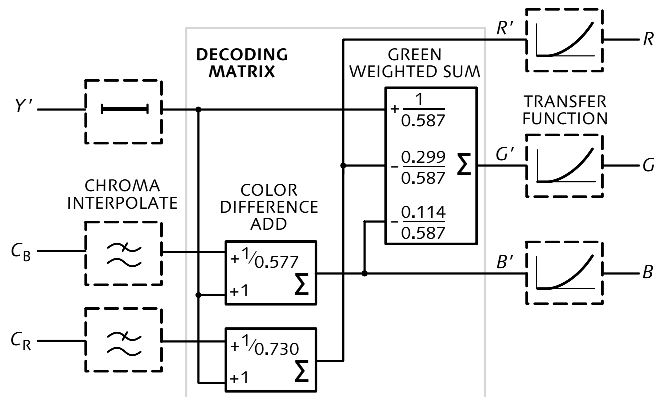
$$C_R = \frac{112}{0.701}\left(R' - {}^{601}Y'\right) \cdot 2^{k-8}$$

**$C_BC_R$ components.** Rec. ITU-R BT.601-4 is the international standard for component digital studio video. Luma coded in 8 bits has an excursion of 219. Color differences $C_B$ and $C_R$ are coded in 8-bit offset binary form with excursions of ±112. $Y'C_BC_R$ coding has a slightly smaller excursion for luma than for chroma: Luma has 219 "risers," compared to 224 for $C_B$ and $C_R$. The notation $C_BC_R$ distinguishes this set from $P_BP_R$, where the luma and chroma excursions are nominally identical. (At the interface, offsets are used. Luma has an offset of +16: Black is at code 16, and white is at code 235. Color differences have an offset of +128, for a range of 16 through 240 inclusive. Levels and equations are shown here without interface offsets.)

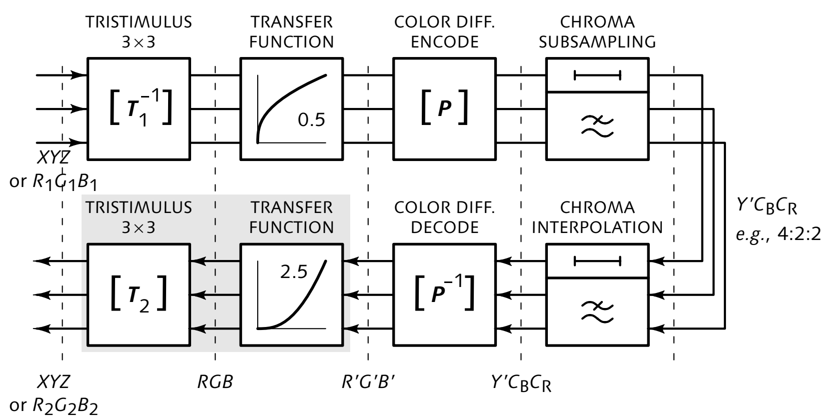
**Conventional (nonconstant luminance) encoder.** The NTSC adopted this *nonconstant luminance* design in 1953. This scheme has been adopted in all practical video systems, including NTSC, PAL, SECAM, component video, JPEG, MPEG, and HDTV. The three blocks enclosed in the dotted outline are equivalent to a single 3×3 matrix multiplication.

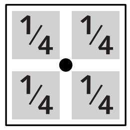



**Conventional (nonconstant luminance) decoder.** Luma is added to the scaled color difference components to recover nonlinear blue and red components. (In a digital decoder, the omitted color difference components are interpolated.) A weighted sum of luma, nonlinear blue, and nonlinear red is formed to recover the nonlinear green component. Finally, all three components are subject to the 2.5-power function that is intrinsic to the CRT display. The nonlinear signals in the channel are coded according to the lightness sensitivity of human vision. If the display device is not a CRT, its intrinsic transfer function must be corrected to obtain an effect equivalent to a 2.5-power function.

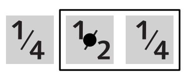


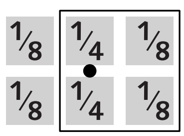

**Color difference encoding and decoding.** From linear *XYZ* – or linear $R_1G_1B_1$ whose chromaticities differ from the interchange standard – apply a 3×3 matrix transform to obtain linear *RGB* according to the interchange primaries. Apply a nonlinear transfer function (gamma correction) to each of the components to obtain nonlinear $R'G'B'$. Apply a 3×3 matrix to obtain luma and color difference components, typically $Y'P_BP_R$ or $Y'C_BC_R$. If necessary, apply a subsampling filter to obtain subsampled color difference components.



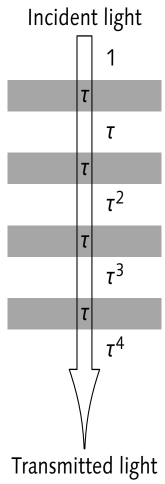


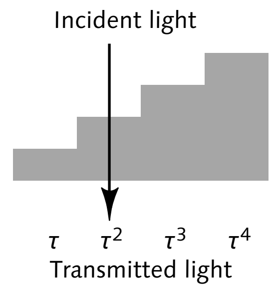

**Interstitial 4:2:0 filter.** Some systems implement 4:2:0 subsampling with minimum computation by simply averaging $C_B$ over a 2×2 block, and averaging $C_R$ over the same 2×2 block. Simple averaging causes subsampled chroma to take an effective position centered among a 2×2 block of luma samples, what I call *interstitial* siting. Low-end decoders simply replicate the subsampled 4:2:0 $C_B$ and $C_R$ to obtain the missing chroma samples, prior to conversion back to *R'G'B'*. This technique is widely used in MPEG-1, in ITU-R Rec. H.261 videoconferencing, and in JPEG/JFIF stillframes in computing. However, this approach is inconsistent with standards for studio video and MPEG-2, where $C_B$ and $C_R$ need to be *cosited* horizontally.

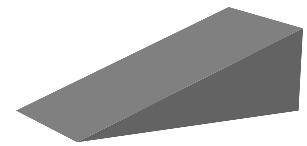



**Cosited filters.** For 4:2:2 sampling, weights of [$^1/_4$, $^1/_2$, $^1/_4$] can be used to achieve cositing as required by Rec. 601, while still using simple computation. That filter can be combined with [$^1/_2$, $^1/_2$] vertical averaging, so as to be extended to 4:2:0. Simple averaging filters have acceptable performance for stillframes, or for desktop PC quality video. However, they exhibit poor image quality. High-end digital video and film equipment uses sophisticated subsampling filters, where the subsampled $C_B$ and $C_R$ of a 2×1 pair (4:2:2) or 2×2 quad (4:2:0) take contributions from many surrounding samples.

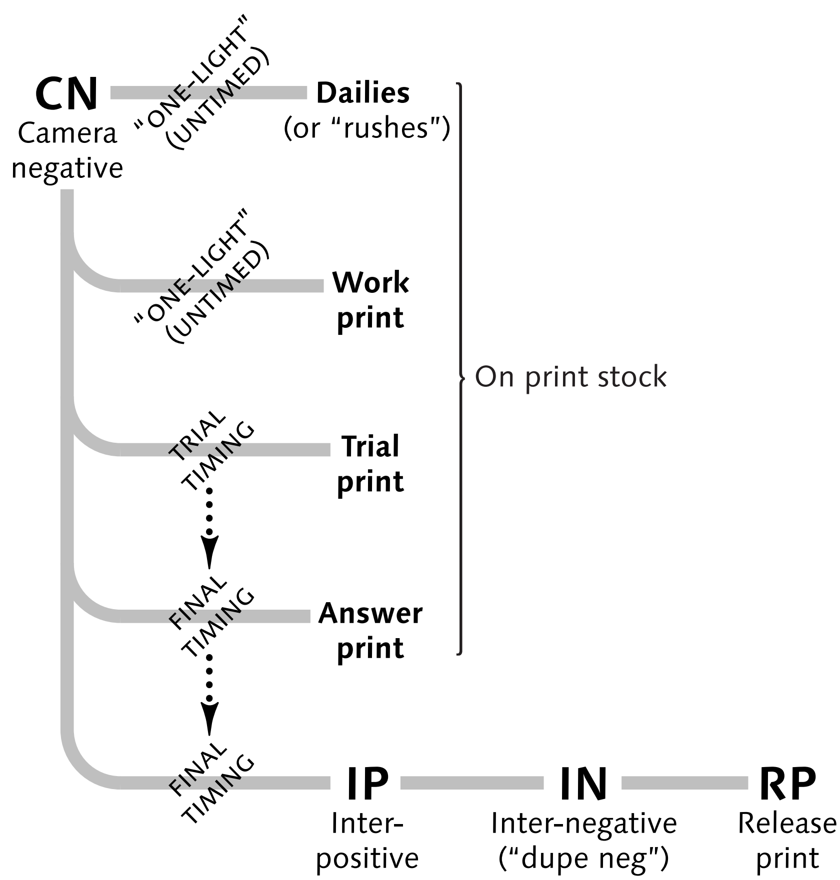


**Light transmission** through layers of equal transmittance *a* is depicted here; light transmitted through *n* layers is $a^n$. Transmittance is proportional to an exponential function of dye thickness (or concentration); this phenomenon is known as Beer's Law. *Optical density* is defined as minus 1 times the base-10 logarithm of transmittance. Owing to Beer's Law, optical density varies linearly with dye concentration.

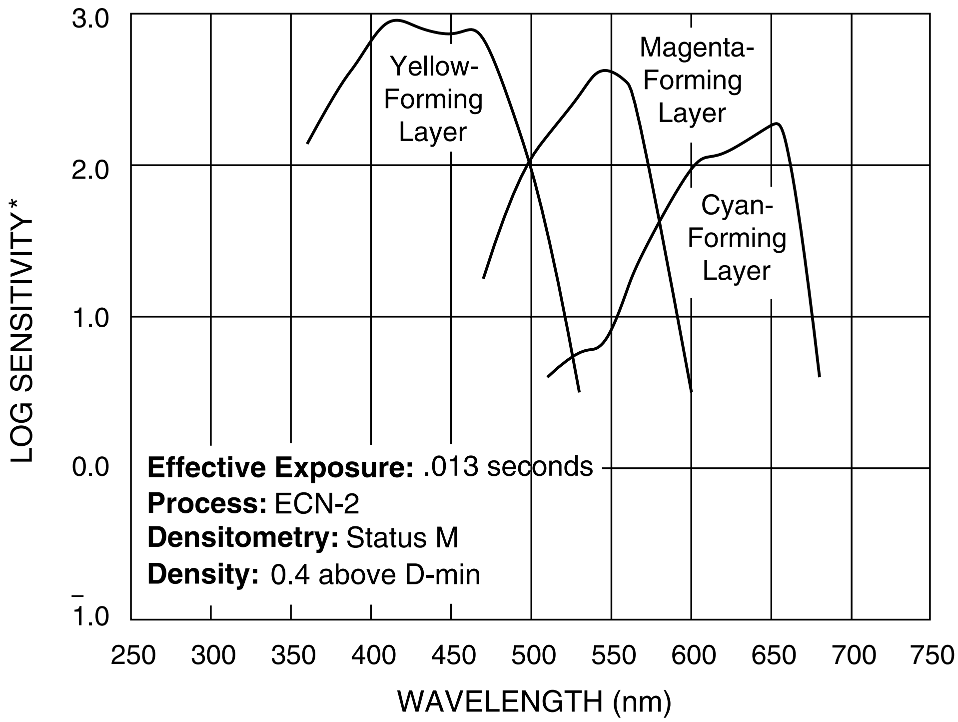
**Density wedge** is constructed with a material such as gelatin infused with colloidal carbon. Transmittance varies exponentially as a function of displacement from the thin end; optical density therefore varies linearly across the length. The combination of Beer's Law and the logarithmic nature of lightness perception causes the the wedge to exibit a roughly perceptually uniform tone scale.




**Traditional film workflow** includes several processing steps, summarized in this sketch. Live action is normally captured on *camera negative* (CN) film, sometimes called *original camera negative* (OCN). Captured scenes are printed onto *print stock*, as dailies, work prints, trial prints, or when color grading has been completed, as an *answer print.* Color grading in film is effected when the OCN is printed onto intermediate film stock to produce an *interpositive* (IP). Though recorded as a positive, the IP is not intended to be directly viewed: It is printed onto intermediate stock to produce an *internegative* (IN). When the movie is ready for distribution, the internegative is printed using ahigh-speed contact printer onto print stock to make *release prints* that are distributed to the theaters.

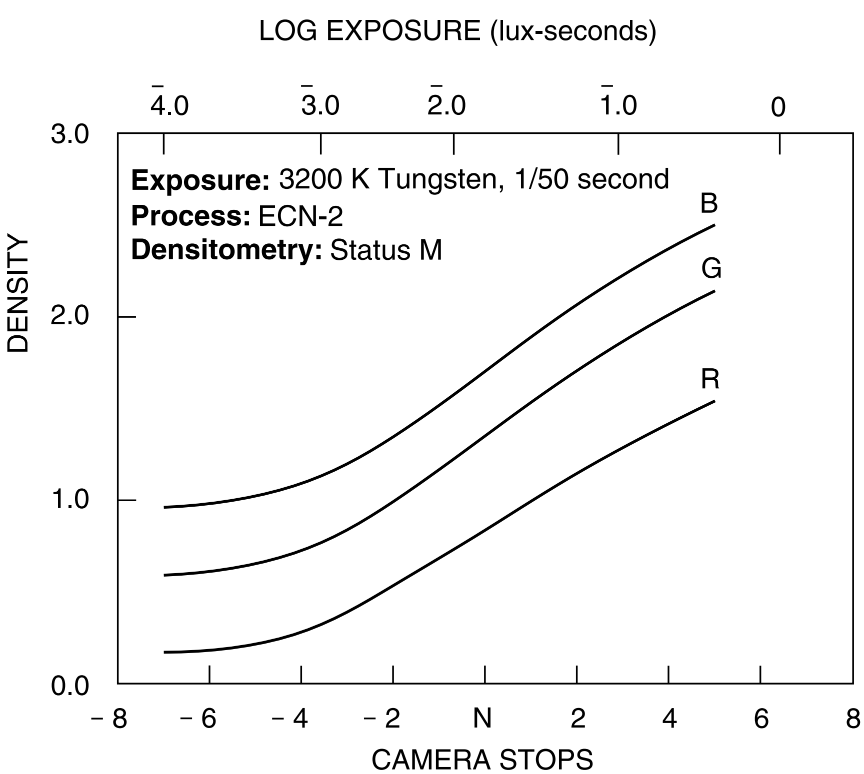
 

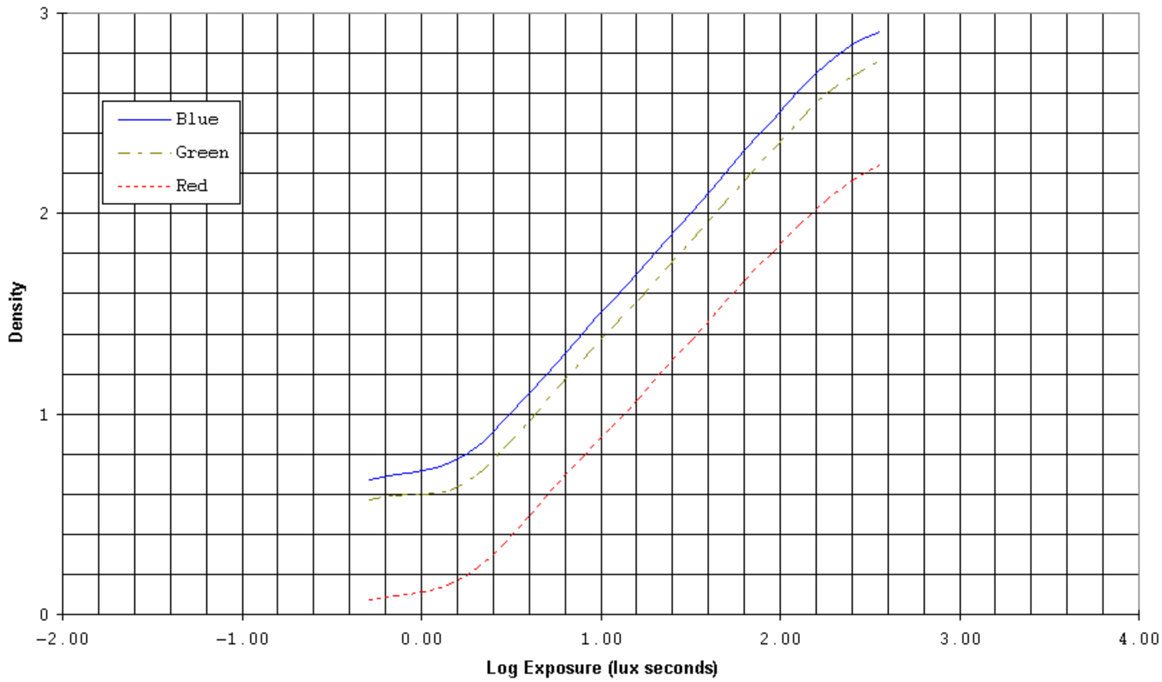
SPECTRAL-SENSITIVITY CURVES 5289 / 7289

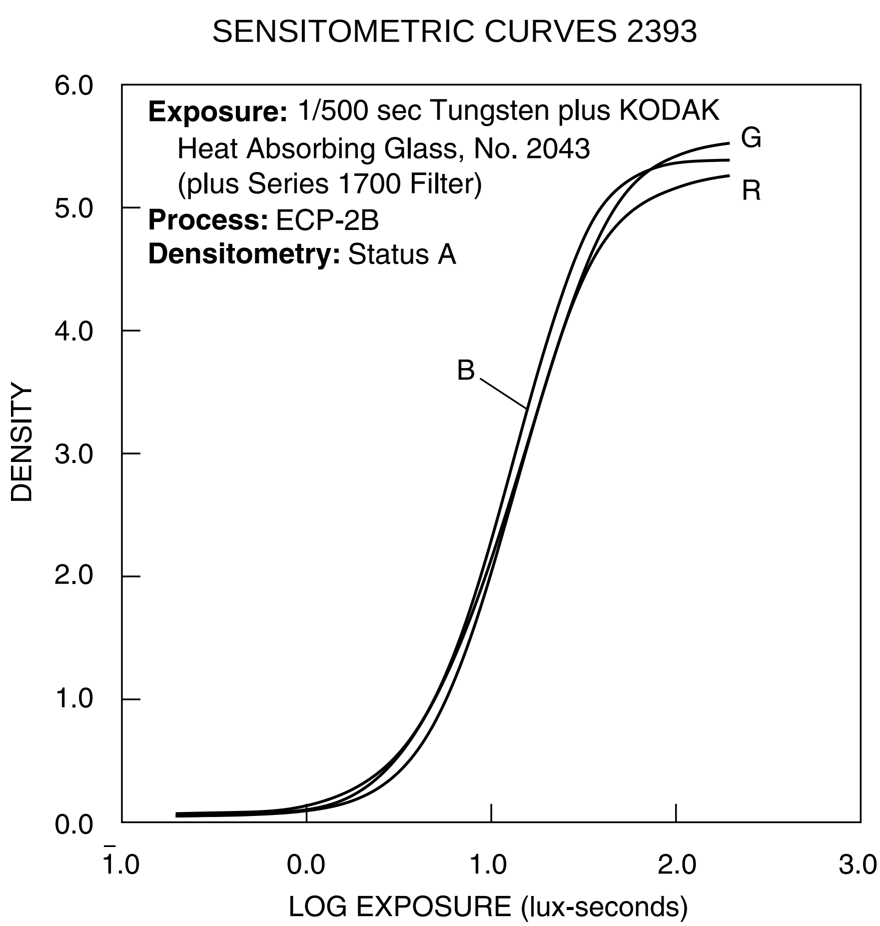


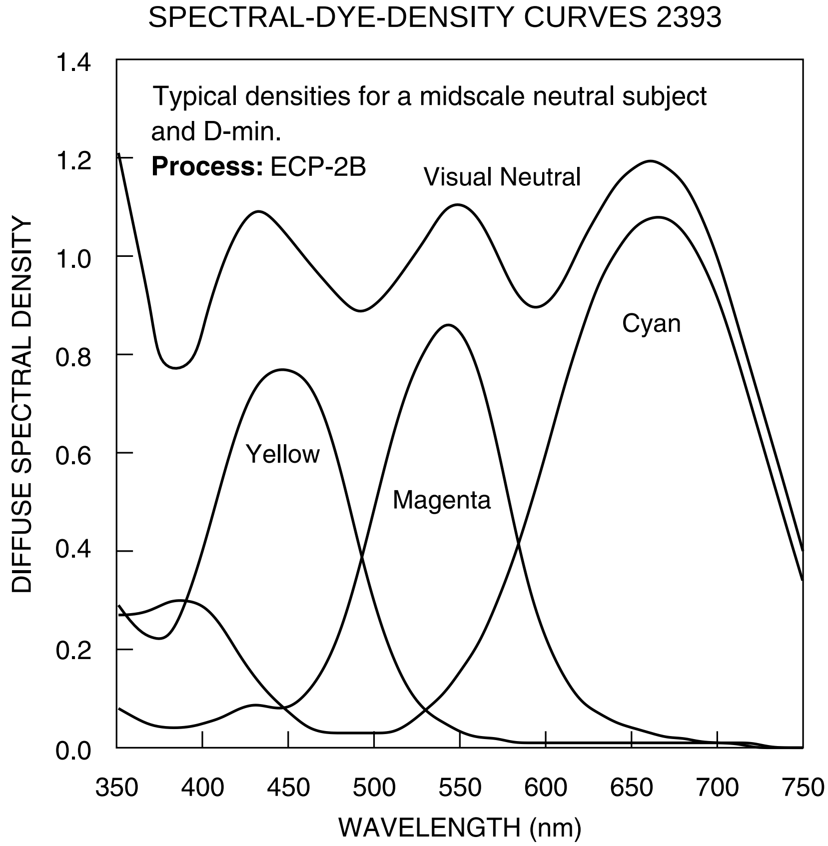

*Sensitivity = reciprocal of exposure (ergs/cm$^2$) required to produce specified density

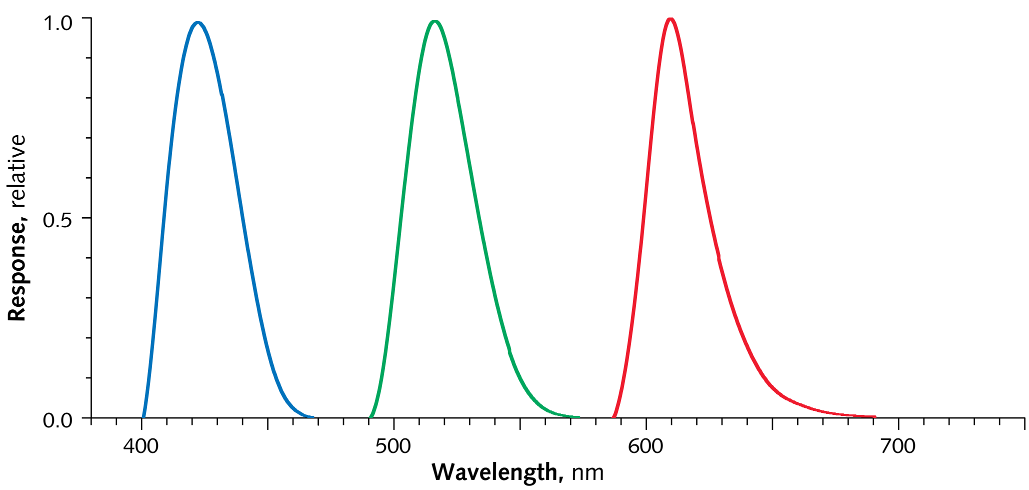
SENSITOMETRIC CURVES 5289 / 7289



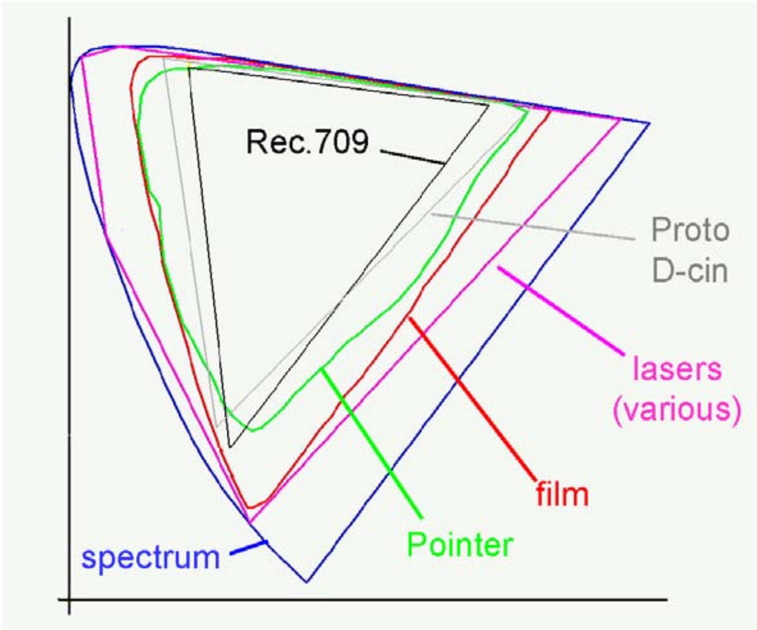

Blue
Green
Red
Density
Log Exposure (lux seconds)


SENSITOMETRIC CURVES 2393
Exposure: 1/500 sec Tungsten plus KODAK Heat Absorbing Glass, No. 2043 (plus Series 1700 Filter)
Process: ECP-2B
Densitometry: Status A
G
R
B
DENSITY
LOG EXPOSURE (lux-seconds)

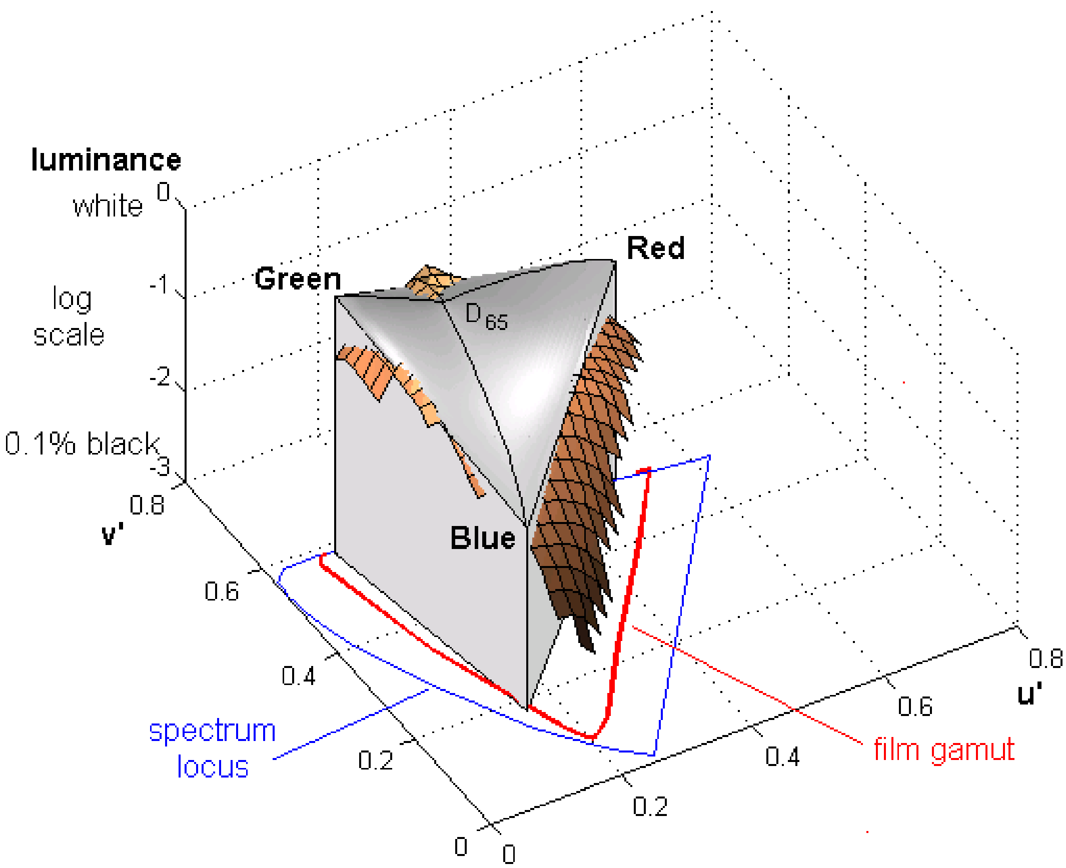


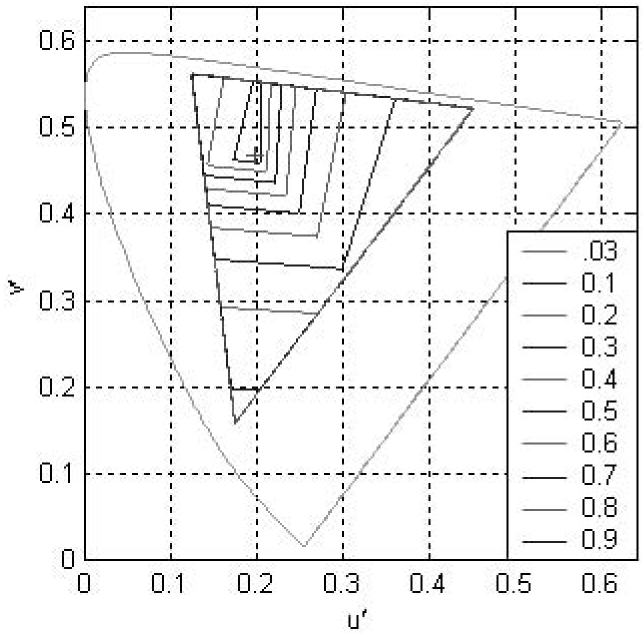

**Status A density** refers to optical density measurements obtained from positive film (intended to be directly viewed), using a standardized set of spectral weighting functions that are chosen to measure density at wavelengths where the dye absorbtion exhibits minimum overlap. A different set of weighting functions (Status M) is appropriate for measuring optical density in negative material. Cineon printing densities (CPD), the basis of color image coding in the DPX file format, are based upon a set of spectral weighting curves specified in SMPTE RP 180.



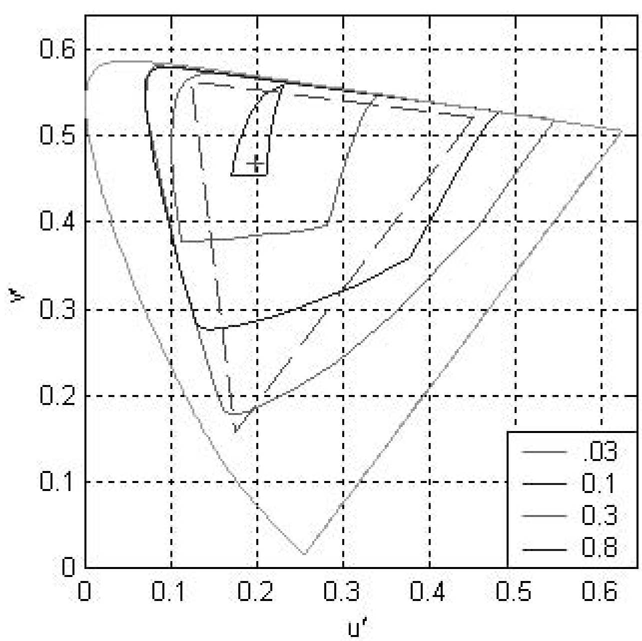

**Gamuts** of various reproduction media are graphed in two dimensions on the CIE [*u'*, *v'*] chromaticity chart. But two dimensions don't tell the whole story: Different ranges of chromaticity values are obtained at different luminance levels. To better visualize gamut, we need to represent the third (luminance) coordinate.



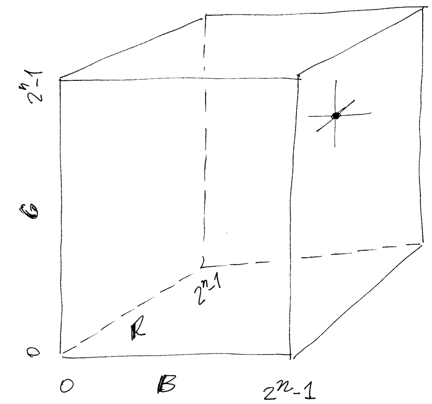

**Gamuts** of Rec. 709, a typical additive *RGB* system, and typical cinema print film, are plotted in three dimensions. Film can reproduce saturated cyan and magenta colors that are outside the Rec. 709 (or sRGB) gamut; however, those colors occur at high luminance levels. The Rec. 709 gamut encompasses a more highly saturated blue than can be reproduced by film. This graphic was created by Chuck Harrison.



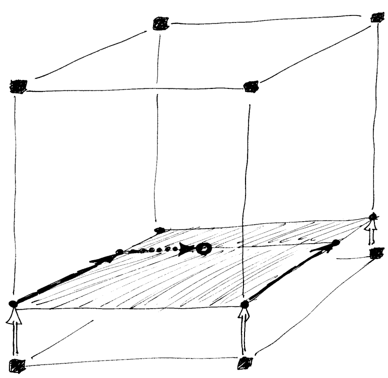

**Luminance contours** of Rec. 709 *RGB:* The chromaticity that is available in an additive *RGB* system depends upon luminance. Highly saturated colors are possible at low luminance; however, as luminance increases, smaller and smaller values of saturation are available. This graph was produced by Dave LeHoty.



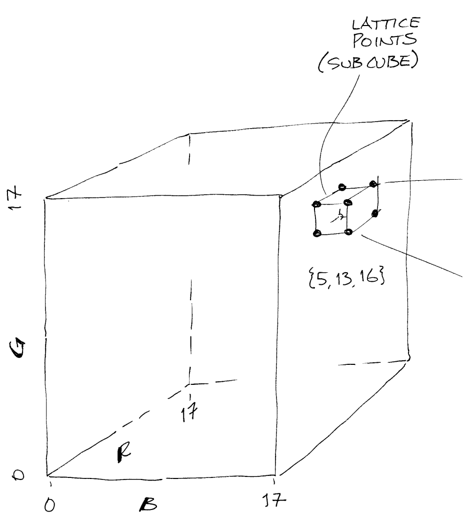

**Luminance contours in film**




**3-D interpolation** could be used to implement a color transform; however, an accurate transform would require a huge lookup table (LUT).

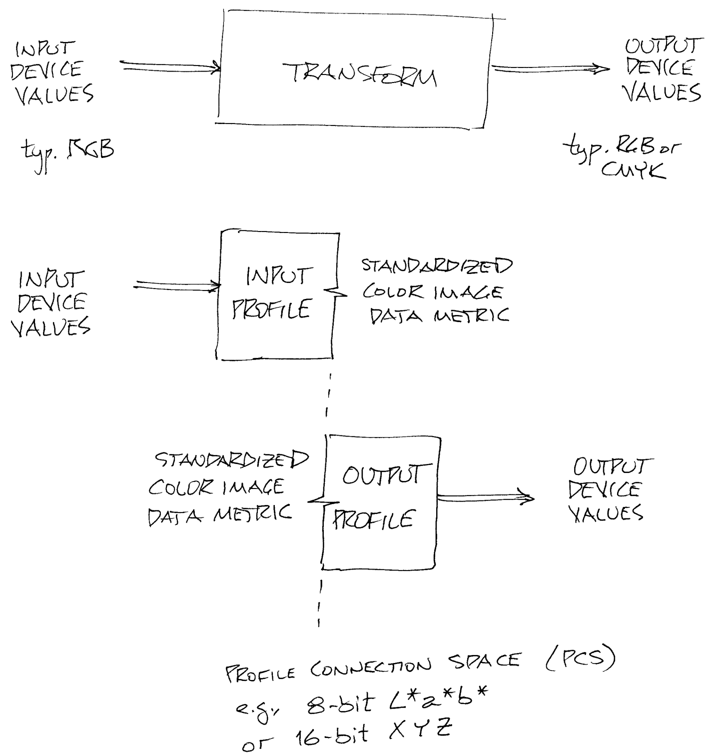


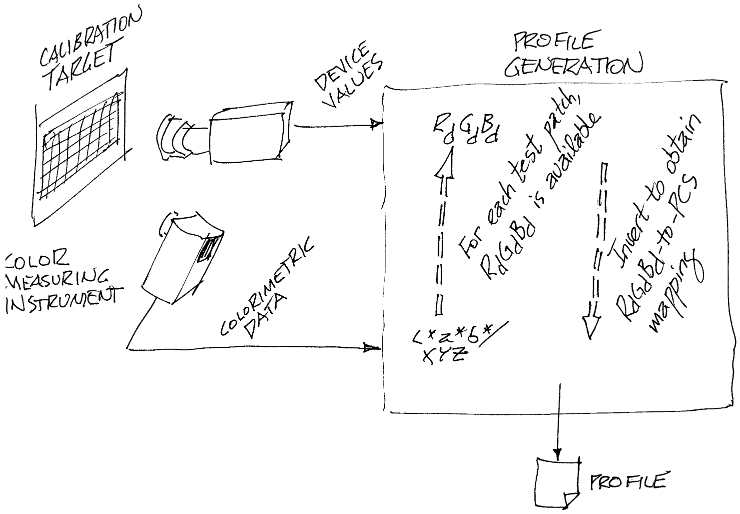
**Trilinear interpolation** starts with output color triples – or for *CMYK*, quads – from eight vertices of a cube. The input values are used to form a suitably-weighted sum of those values, component-wise. This scheme could be used to implement a color transform; good performance would be obtained for certain kinds of well-behaved transforms on input and output color spaces having similar transfer functions. However, the scheme fails to deliver good results for transforms involving color spaces that involve nonlinearities. Many practical transforms, particularly transforms from *RGB* to *CMYK*, are nonlinear.



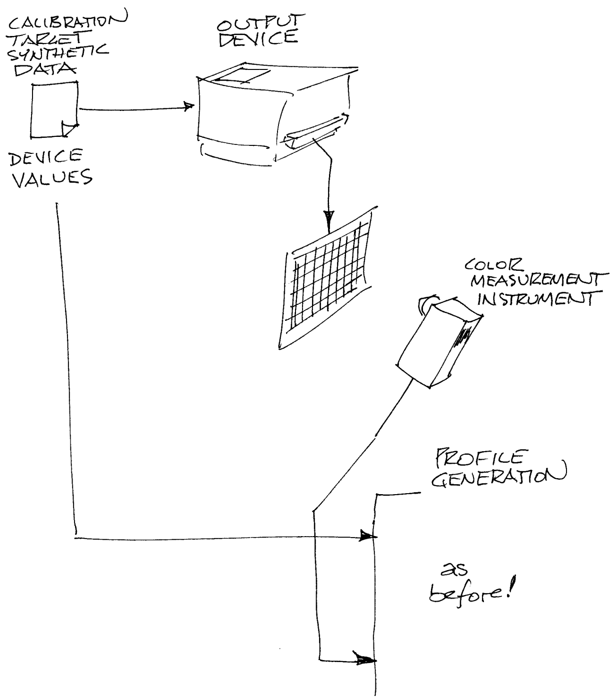

**A combination** of 3-D LUT and trilinear interpolation techniques – *3-D LUT interpolation* – is used in the ICC architecture. The input space is diced into several thousand lattice cubes (perhaps $16^3$, or 4096). Output color triples – or for CMYK, quads – are stored at each vertex. (This example has $17^3$, or 4913, vertices.) To transform a source value, each of its components is partitioned into most-significant and least-significant portions; for 8-bit data, this might be considered a 4 bit integer and a 4-bit fraction. The integer portions of all 3 components are used to access 8 lattice points. The fractional components of the source values are then used as coefficients in trilinear interpolation. The result is a set of destination component values.

**Color transforms** implement the numerical transformation from input device values (typically *RGB*) to output device values (typically either *RGB or CMYK*). In the ICC color management architecture, a device-to-device transform is computed as the concatenation of an input transform and an output transform. The numerical properties of each transform are specified by an ICC *profile.* An *input profile* transforms from device values to values in a standard color space, either CIE *XYZ* or CIE *L*a*b**, denoted the *profile connection space* (PCS). An *output profile* transforms from the profile connection space to device values.

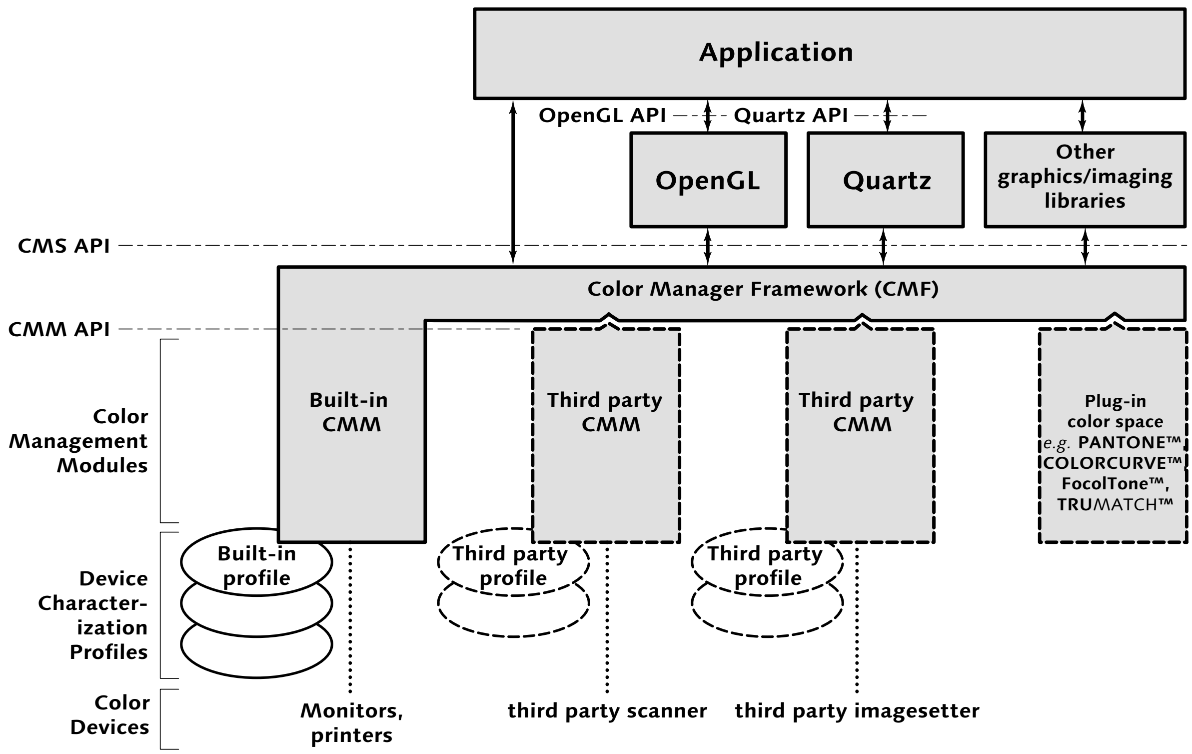





**Input devices are characterized** by scanning a test target containing several dozen or several hundred patches, to obtain device values. These patches are also measured by a color measuring instrument such as a colorimeter or a spectrophotometer. Given access to device values and the corresponding colorimetric values, profile generation software uses numerical optimization techniques to construct an ICC profile that, when passed to a color management system, allows a transform from arbitrary device values to the corresponding colorimetric values.




**Output devices are characterized** by generating a test stimulus (a monitor display, or printer output) containing several dozen or several hundred patches. These patches are measured by a color measuring instrument. An output profile is generated in a manner nearly identical to generation of an input profile.




**Color management systems** are accessed by an application program, illustrated at the top of this block diagram. Underneath the application is a set of graphics libraries, each presenting an *application program interface* (API). Underneath the graphics libraries is the *color management system* (CMS), which serves as a dispatcher for color management capabilities that are available. The mathematical transformations of color are performed by *color management modules* (CMMs) that plug into the CMS through a private CMM API. Each CMM accesses *device profiles;* each device profile is specific to input device (such as a scanner or a camera) or an output device (such as a printer or imagesetter).

# The rehabilitation of *gamma* A

Charles Poynton
poynton@poynton.com
www.poynton.com

**Abstract**

*Gamma* characterizes the reproduction of tone scale in an imaging system. Gamma summarizes, in a single numerical parameter, the nonlinear relationship between code value – in an 8-bit system, from 0 through 255 – and luminance. Nearly all image coding systems are nonlinear, and so involve values of gamma different from unity.

Owing to poor understanding of tone scale reproduction, and to misconceptions about nonlinear coding, gamma has acquired a terrible reputation in computer graphics and image processing. In addition, the world-wide web suffers from poor reproduction of grayscale and color images, due to poor handling of nonlinear image coding. This paper aims to make gamma respectable again.

## Gamma's bad reputation

The left-hand column in this table summarizes the allegations that have led to gamma's bad reputation. But the reputation is ill-founded – these allegations are false! In the right column, I outline the facts:

| *Misconception* | *Fact* |
|---|---|
| A CRT's phosphor has a nonlinear response to beam current. | The *electron gun* of a CRT is responsible for its nonlinearity, not the phosphor. |
| The nonlinearity of a CRT monitor is a defect that needs to be corrected. | The nonlinearity of a CRT is very nearly the inverse of the lightness sensitivity of human vision. The nonlinearity causes a CRT's response to be roughly perceptually uniform. Far from being a defect, this feature is highly desirable. |
| The main purpose of gamma correction is to compensate for the nonlinearity of the CRT. | The main purpose of gamma correction in video, desktop graphics, prepress, JPEG, and MPEG is to code luminance or tristimulus values (proportional to intensity) into a perceptually-uniform domain, so as optimize perceptual performance of a limited number of bits in each *RGB* (or *CMYK*) component. |

Reprinted from Rogowitz, B.E., and T.N. Pappas (eds.), *Human Vision and Electronic Imaging III*, Proceedings of SPIE/IS&T Conference 3299, San Jose, Calif., Jan. 26–30, 1998 (Bellingham, Wash.: SPIE, 1998). 

| *Misconception* | *Fact* |
|---|---|
| Ideally, linear-intensity representations should be used to represent image data. | If a quantity proportional to intensity represents image data, then 11 bits or more would be necessary in each component to achieve high-quality image reproduction. With nonlinear (gamma-corrected) coding, just 8 bits are sufficient. |
| A CRT is characterized by a power function that relates luminance $L$ to voltage $V'$: $L=(V')^{\gamma}$. | A CRT is characterized by a power function, but including a black-level offset term: $L=(V'+\varepsilon)^{\gamma}$. Usually, $\gamma$ has a value quite close to 2.5; if you're limited to a single-parameter model, $L=(V'+\varepsilon)^{2.5}$ is much better than $L=(V')^{\gamma}$. |
| The exponent $\gamma$ varies anywhere from about 1.4 to 3.5. | The exponent itself varies over a rather narrow range, about 2.35 to 2.55. The alleged wide variation comes from variation in *offset* term of the equation, not the exponent: Wide variation is due to failure to correctly set the black level. |
| Gamma correction is accomplished by inverting this equation. | Gamma correction is roughly the inverse of this equation, but two alterations must be introduced to achieve good perceptual performance. First, a linear segment is introduced into the transfer function, to minimize the introduction of noise in very dark areas of the image. Second, the exponent at the encoder is made somewhat greater than the ideal mathematical value, in order to impose a *rendering intent* that compensates for subjective effects upon image display. |
| CRT variation is responsible for wide variability in tone scale reproduction when images are exchanged among computers. | Poor performance in image exchange is generally due to lack of control over transfer functions that are applied when image data is acquired, processed, stored, and displayed. |
| Macintosh monitors have nonstandard values of gamma. | All CRT monitors, including those used with Macintosh computers, produce essentially identical response to voltage. But the Macintosh QuickDraw graphics subsystem involves a lookup table that is loaded by default with an unusual transfer function. It is the default values loaded into the lookup table, not the monitor characteristics, that impose the nonstandard Macintosh gamma. |
| Gamma problems can be circumvented by loading a lookup table having a suitable gamma value. | Loading a particular lookup table, or a particular value of *gamma,* alters the relationship of data in the frame buffer to linear-light "intensity" (properly, luminance, or tristimulus value). This may have the intended effect on a particular image. However, loading a new lookup table will disturb the code-to-luminance mapping that is assumed by the graphics subsystem, by other images, or by other windows. This is liable to alter color values that are supposed to stay fixed. |
| Macintosh computers are shipped from the factory with gamma set to 1.8. SGI machines default to gamma of 1.7. To make an SGI machine display pictures like a Mac, set SGI gamma to 1.8. | On the Macintosh, setting a numerical gamma setting of $g$ loads into the framebuffer's lookup table a power function with the exponent $g/2.61$. On an SGI, setting a numerical gamma setting of $g$ loads into the lookup table a power function with the exponent $1/g$. To make an SGI machine behave like a Mac, you must set SGI gamma to 1.45. |
| Gamma problems can be avoided when exchanging images by tagging every image file with a suitable gamma value. | Various tag schemes have been standardized; some tags are coded into image files. However, application software today generally pays no attention to the tags, so tagging image files is not helpful today. It is obviously a good idea to avoid subjecting an image file to cascaded transfer functions during processing. However, the tag approach fails to recognize that image data should be originated and maintained in a perceptually-based code. |
| JPEG compresses *RGB* data, and reproduces *RGB* data upon decompression. The JPEG algorithm itself is completely independent of whatever transfer function is used. | JPEG and other lossy image compression algorithms depend on discarding information that won't be *perceived*. It is vital that the data presented to a JPEG compressor be coded in a perceptually-uniform manner, so that the information discarded has minimal perceptual impact. Also, although standardized as an image *compression* algorithm, JPEG is so popular that it is now effectively an image *interchange* standard. Standardization of the transfer function is necessary in order for JPEG to meet its users' expectations. |

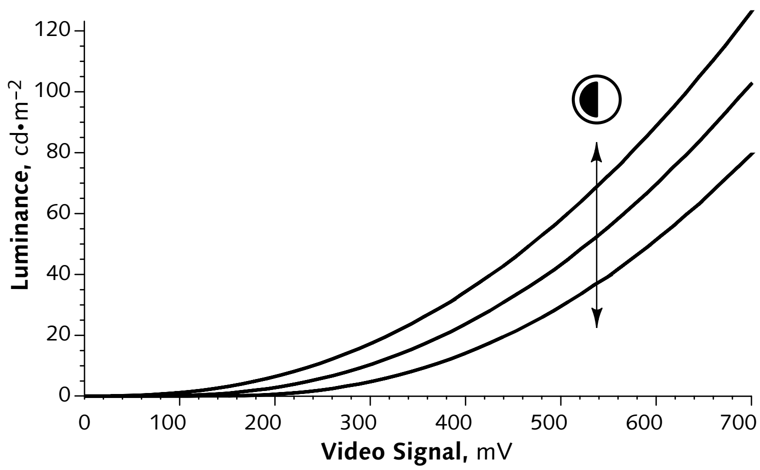



Figure A.1 **CRT's transfer function** is shown at three different settings of the CONTRAST (or PICTURE) control. Here I show CONTRAST altering the *y*-axis (luminance) scaling; owing to the properties of a power function, scaling the *x*-axis (video signal) has an equivalent effect. The graph indicates a video signal having a voltage from zero to 700 mV. In a typical eight-bit digital-to-analog converter in a computer graphics subsystem, black is at code zero, and white is at

## Intensity

*Intensity* is the rate of flow of radiant energy, per unit solid angle – that is, in a particular, specified direction. In image science, we measure power over some interval of the electromagnetic spectrum. We're usually interested in power radiating from or incident on a surface. Intensity is what I call a *linear-light* measure, expressed in units such as watts per steradian.

The CIE has defined *luminance*, denoted *Y*, as intensity per unit area, weighted by a spectral sensitivity function that is characteristic of vision. The magnitude of luminance is proportional to physical power; in that sense it is like intensity. But the spectral composition of luminance is related to the brightness sensitivity of human vision.

Video equipment forms a *luma* component $Y'$ as a weighted sum of *nonlinear* $R'G'B'$ primary components. The nonlinear quantity is often incorrectly referred to as *luminance* by video engineers who are unfamiliar with color science.

Luminance can be computed as a properly-weighted sum of linear-light (tristimulus) red, green, and blue primary components. For contemporary video cameras, studio standards, and CRT phosphors, the luminance equation is this:

$$^{709}Y = 0.2126\,R + 0.7152\,G + 0.0722\,B$$

See Olson, Thor, "Behind Gamma's Disguise," in *SMPTE Journal*, v. 104, p. 452 (June 1995).

The luminance generated by a physical device is usually not proportional to the applied signal – usually, there is a nonlinear relationship. A conventional CRT has a power-law response to voltage: Luminance produced at the face of the display is approximately the applied voltage raised to the five-halves power. The numerical value of the exponent of this power function, 2.5, is colloquially known as *gamma*. This nonlinearity must be compensated in order to achieve correct reproduction of luminance. An example of the response of an actual CRT is graphed, at three settings of the CONTRAST control, in Figure A.1 above.

Berns, Roy S., Ricardo J. Motta, and M.E. Gorzynski, "CRT Colorimetry: Part 1, *Theory and Practice*; Part 2, *Metrology*," in *Color Research and Application*, v. 18, 299–325 (1993).

It is alleged that the power function exponent $\gamma$ of a CRT varies over a wide range; values as low as 1.4 and as high as 3.5 are cited in the literature. The graphs and captions in Figures A.4 and A.6 opposite show that wide variation in the apparent gamma value will result if the monitor's BLACK LEVEL (or "BRIGHTNESS") control is improperly adjusted.

## Lightness

Publication CIE No 15.2, *Colorimetry*, Second Edition. (Vienna: Central Bureau of the Commission Internationale de L'Éclairage, 1986)

At a particular level of adaptation, human vision responds to about a hundred-to-one *contrast ratio* of luminance from white to black. Within this range, vision has a nonlinear response to luminance: Lightness perception is roughly logarithmic. A source having a luminance only 18% of a reference luminance appears about half as bright. The perceptual response to luminance is called *Lightness.* Vision researchers have modeled lightness sensitivity with various mathematical functions, as shown in Figure A.2 below. The CIE has adopted a standard function $L^*$ (pronounced "EL-star"), defined as a modified cube root:

$$L^* = \begin{cases} 903.3\dfrac{Y}{Y_n}; & \dfrac{Y}{Y_n} \le 0.008856 \\ 116\left(\dfrac{Y}{Y_n}\right)^{\frac{1}{3}} - 16; & 0.008856 < \dfrac{Y}{Y_n} \end{cases}$$

$Y_n$ is the luminance of the white reference. A linear segment with a slope of 903.3 is applied near black. $L^*$ has a range of 0 to 100. A unit change in $L^*$ is taken to be approximately the threshold of visibility. In other words, you can detect a difference between intensities when the ratio between them is greater than about one percent.

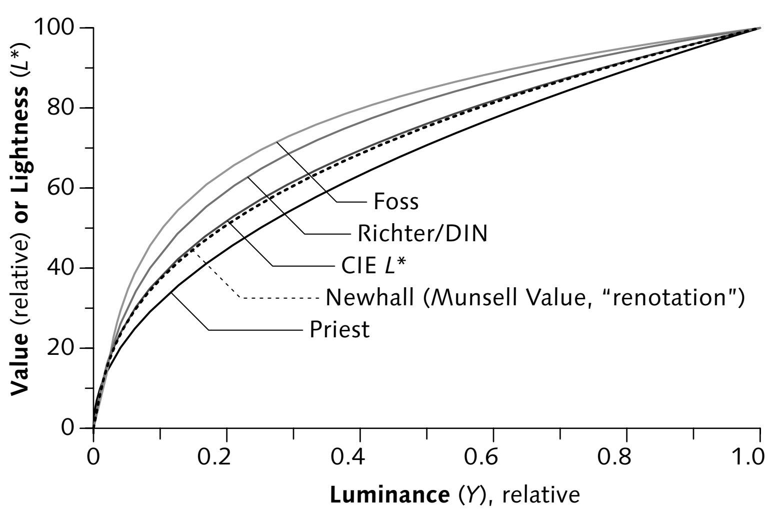



Figure A.2 **Luminance and lightness.** The relationship between lightness-scale value $V$ and luminance factor $Y$ is plotted in accordance with different formulae. Redrawn from Fig. 2 (6.3) from Wyszecki and Stiles, *Color Science* (New York: Wiley, 1982).

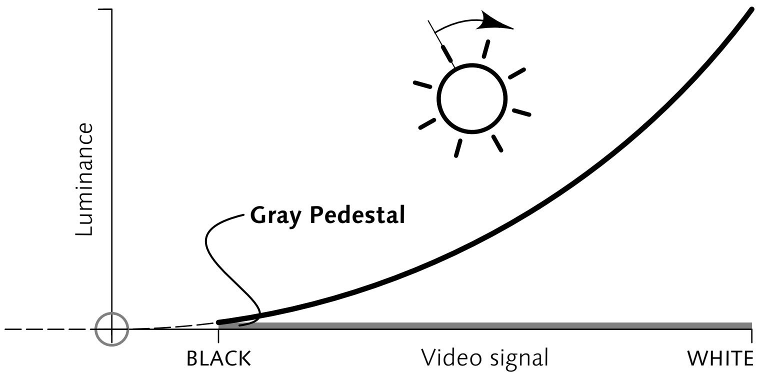



Figure A.3 **BRIGHTNESS control** has the effect of sliding the black-to-white video signal scale left and right along the 2.5-power function of the display. Here, BRIGHTNESS is set too high; a significant amount of luminance is produced at zero video signal level. No video signal can cause true black to be displayed, and the picture content rides on an overall pedestal of gray. Contrast ratio is degraded.

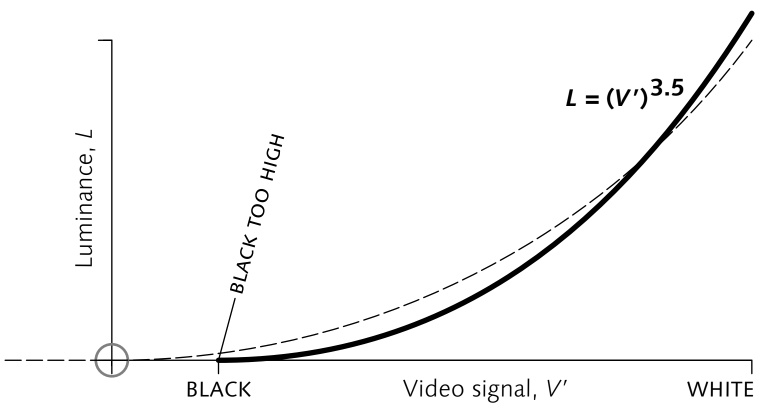



Figure A.4 **Gamma 3.5** A naive approach to the measurement of CRT nonlinearity is to model the response as $L = (V')^{\gamma}$, and to find the exponent of the power function that is the best fit to the voltage-to-intensity transfer function of a particular CRT. However, if this measurement is undertaken with BRIGHTNESS set too high, an unrealistically large value of gamma results from the modelled curve being "pegged" at the origin.

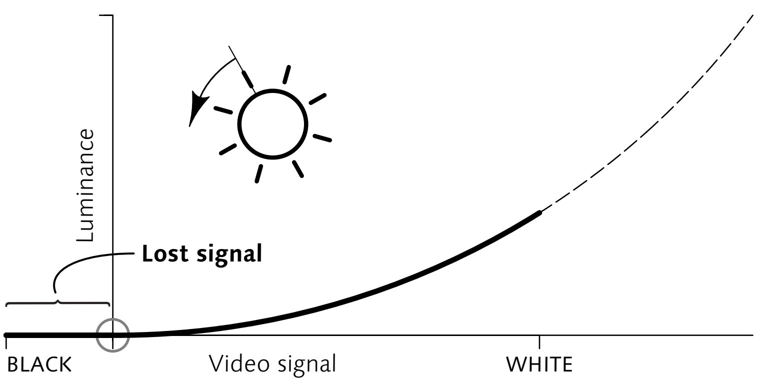



Figure A.5 **BRIGHTNESS control set too low** causes a range of input signal levels near black to be reproduced "crushed" or "swallowed," reproduced indistinguishably from black. A cinematographer might refer to this situation as "lack of details in the shadows," however, *all* information in the shadows is lost, not just the details.

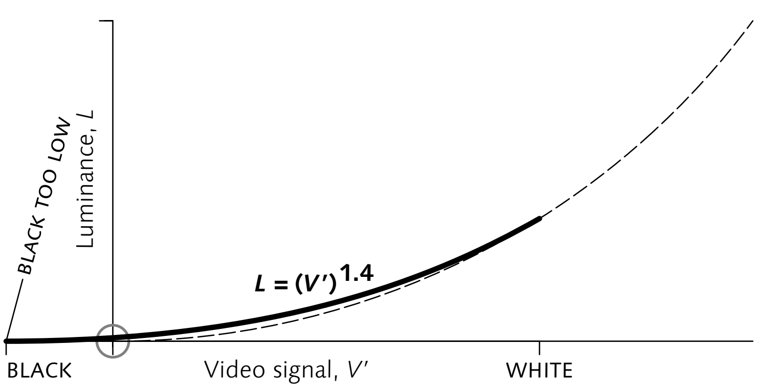



Figure A.6 **Gamma 1.4** If the transfer function is modelled as $L = (V')^{\gamma}$ with *Black Level* set too low, an unrealistically small value of gamma results. However, if the transfer function is modeled with a function of the form $L = (V' + \varepsilon)^{2.5}$ that accommodates black level error, then a good fit is achieved. Misintepretations in the measurement of CRT nonlinearity have led to assertions about CRTs being highly unpredictable devices, and have led to image exchange standards employing quite unrealistic values of gamma.

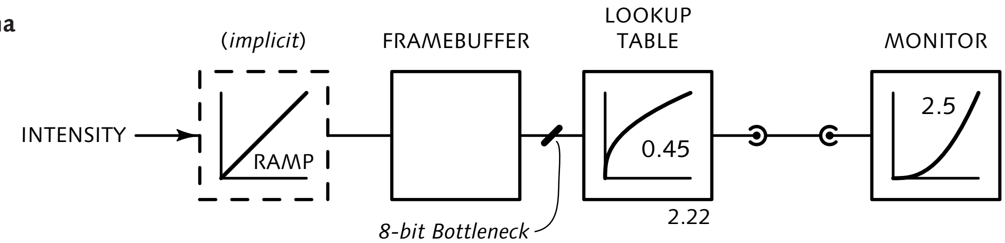



Figure A.8 **Gamma in computer-generated imagery (CGI)**

## Linear light coding

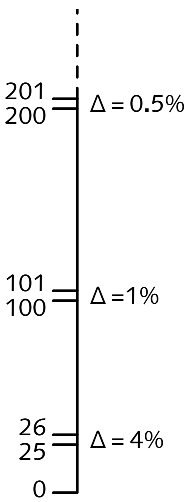



Figure A.7 **Linear light coding: the code 100 problem**

Suppose that you wish to convey luminance values of an image through a channel having a few hundred or a few thousand discrete levels. Consider linear light coding, sketched in the margin, where code zero represents black. No matter what code is at the top end, code 100 represents a shade of gray that lies approximately at the perceptual threshold. For codes below 100, the ratio of intensities between adjacent code values is greater than 1 percent. At code 25, the ratio between adjacent codes is 4 percent. In smooth-shaded regions of an image, the luminance difference between adjacent code values, such as between code 25 and code 26, will cause visible *banding* or *contouring*. For codes above 100, the ratio of luminance values between adjacent codes is less than 1 percent: Code 201 is perceptually useless, and could be discarded without being noticed. In an 8-bit system, the highest code value – the brightest white – is at code 255. In an 8-bit linear-light system, the ratio between the brightest white and the darkest grey that can be reproduced without contouring is a mere 2.55:1.

To avoid perceptible steps at the black end of the scale, it is necessary to have coding that represents different luminance levels 1.00, 1.01, 1.02, and so on. If linear light coding is used, an absolute "delta" of 0.01 must be maintained all the way up the scale to white. To encompass the 100:1 luminance range vision of requires about 9900 codes, or about 14 bits for each of the *R, G,* and *B* components of the image.

Stokes, Mike, Mark D. Fairchild, and Roy S. Berns, "Precision requirements for digital color reproduction," in *ACM Transactions on Graphics*, v. 11, n. 4 (Oct. 1992), 406–422.

If you use nonlinear coding, then the 1.01 "delta" required at the black end of the scale applies as a ratio – not as an absolute increment – and progresses like compound interest up to white. This results in about 463 codes, or about nine bits per component. Eight bits, nonlinearly coded according to Rec. 709, is sufficient for broadcast-quality digital television at a contrast ratio of about 50:1.

In computer-generated imagery (CGI), linear-light coding is typically used in the frame buffer, as sketched in Figure A.8 above. Often only 8 bits are provided in the framebuffer. When luminance data traverses the *8-bit Bottleneck* indicated in the sketch, serious contouring results.

## Video coding

To code luminance into a small number of steps, say 256, then the codes should be assigned to intensities according to the properties of perception, in order for the most effective perceptual use to be made of

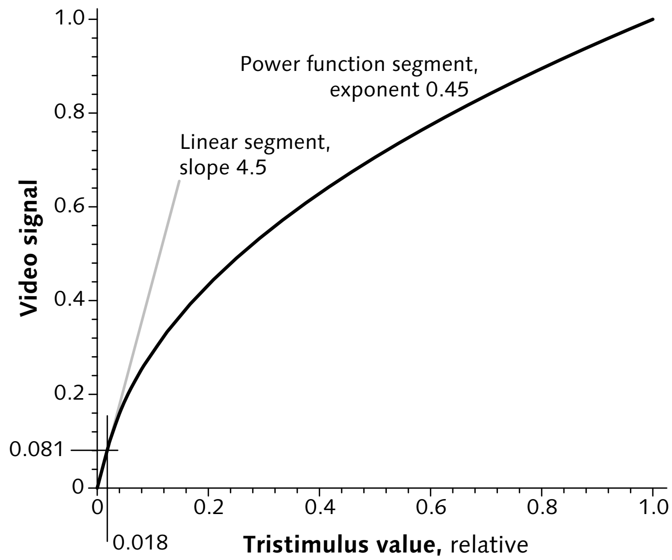


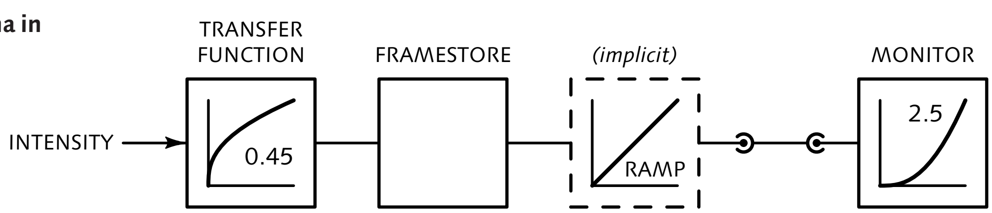

Figure A.9 **The Rec. 709 transfer function of video** mimics the lightness sensitivity of vision. The standard is based on a power function with an exponent of 0.45. Theoretically, a pure power function suffices for gamma correction. In a practical system such as a television camera, the slope of the function is limited near zero in order to minimize noise in the dark regions of the picture.

the available codes. A transfer function similar to the lightness sensitivity of vision should be imposed at encoding.

A CRT's response is very nearly the inverse of the lightness sensitivity of vision: When image data is coded for perception at the encoder – for example, by the Rec. 709 transfer function graphed in Figure A.9 above – the coding is inverted by the CRT, without the necessity to dedicate any circuitry to the task. The fact that a CRT's transfer function is very nearly the inverse of the lightness sensitivity of vision is an amazing, and fortunate, coincidence! The Rec. 709 transfer function standardized for 525/59.94 studio video, 625/50 studio video, and HDTV.

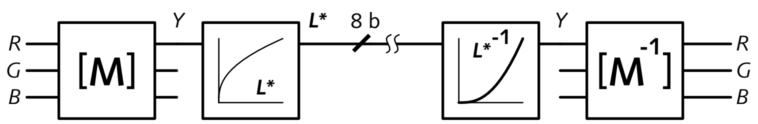
A summary sketch of gamma in video is shown in Figure A.10 below. At the camera, luminance (or, in a color system, a set of three tristimulus values) is subjected to the Rec. 709 transfer function – or loosely, *gamma correction* – whose graph resembles the lightness sensitivity of vision. Video data is stored, processed, recorded, and transmitted in the perceptual domain. The monitor inverts the transform. The main purpose of gamma correction is to code luminance into a perceptually uniform domain, so as to obtain the best perceptual performance from a limited number of bits in each of $R'$, $G'$, and $B'$. (The prime symbols denote the nonlinearity.)

Figure A.10 **Gamma in video**



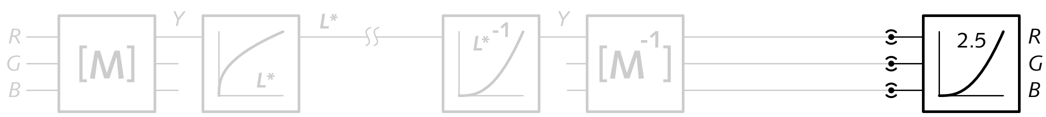
## Rendering

Nonlinear encoding involves applying a transfer function similar to the lightness sensitivity of human vision. Ideally, luminance would first be *matrixed*, that is, formed as a weighted sum of linear-light (tristimulus) *RGB* signals. Then, the CIE *L** transfer function would be applied, to code the signal into a perceptually uniform domain. At the decoder, the inverse of the *L** function would restore luminance, then the inverse matrix would reconstruct *RGB*. The *L** signal would be accompanied by two other signals, to enable the representation of color:

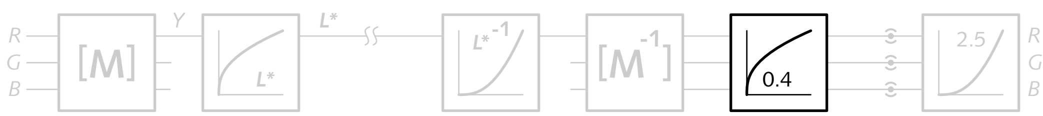



Coding *L** with eight bits achieves good image quality. Coding *Y* directly would require 11 bits or more to achieve similar quality.

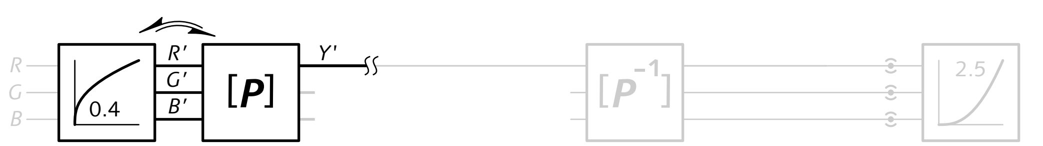
As I have outlined, the electron gun of a CRT monitor introduces a power function having an exponent of about 2.5:



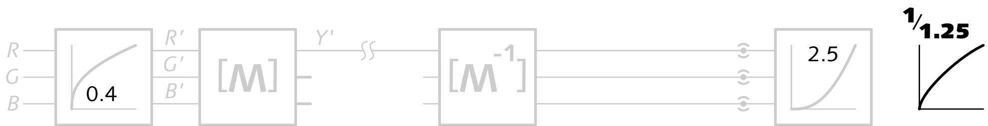

If we were to encode according to the *L** function, the decoder would have to invert that function, then impose the inverse of the 2.5-power function of the CRT:



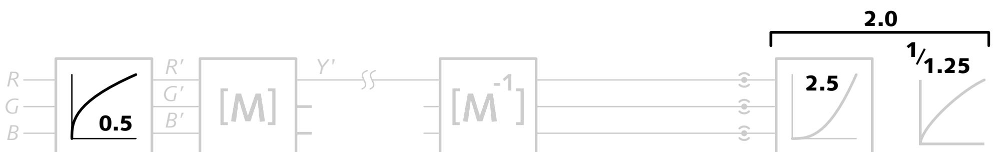

The CRT's power function is so similar to the inverse of the *L** function that we make an engineering compromise: Instead of encoding *Y* using the *L** transfer function, we encode *RGB* intensities to the inverse of the CRT's function. This allows us to dispense completely with transfer function circuitry at the display. We must then interchange the order of the matrix and the transfer function at the encoder. Changing the order of operations causes a departure from the *Principle of constant luminance*. In theory, the encoder would require a 0.4-power function:




This arrangement reproduces physical luminance correctly. However, it has a serious problem: The pictures do not look very good! When viewing a reproduced image, human viewers prefer a reproduction whose contrast ratio has been stretched slightly to a reproduction that is physically correct. The subjective preference depends somewhat upon the viewing environment. In effect, the visual system of the viewer imposes a power function with an exponent of about $^1/_{1.25}$:




For television, a power function with an exponent of about 1.25 must be applied to overcome this effect, in order to produce images that are subjectively pleasing. Rather than introducing circuitry at the display to apply this function, we modify the transfer function at encoder. We use an exponent of about 0.5, instead of the physically-correct 0.4:




If you think of encoding in physical terms, you could consider a video image to be encoded such that a 2.0-power function would reproduce physically-correct luminance at the display. (The NTSC video standard is often said to have gamma of 2.2, because the FCC standards described gamma in this way.) However, I think it is more evocative consider the application of the power function at the encoder to impose a *rendering intent* upon the image data. The encoding assumes the image is to be reproduced in a subjectively-acceptable manner through a physical 2.5-power function at the display.

Though ubiquitous in video, this subjective correction is rarely considered explicitly in computer graphics; belief in the "bits are bits" philosophy suggests to programmers that luminance should be reproduced in the physically-correct manner. However, subjective correction is as necessary in computer graphics as it is in video, and the Rec. 709 transfer function is appropriate for computer graphics. In traditional computer-generated imagery (CGI), as in Figure A.8, the subjective correction is typically accomplished by "gamma correction" using a $^1/_{2.2}$-power function (instead of $^1/_{2.5}$). This is called "gamma of 2.2".

Poynton, Charles, *A Technical Introduction to Digital Video* (New York: Wiley, 1996).

The Rec. 709 transfer function is standard for 525/59.94 and 625/60 conventional video, and for HDTV. The Rec. 709 function is based on a power function exponent of 0.45, but the pure power function is modified by the insertion of a linear segment near black. The overall function is very similar to a square root. For details, consult the *Gamma* chapter in my book. In the diagrams in this section, I use the notation *0.5* as shorthand for the Rec. 709 function.

Rec. 709 appears to strictly define the transfer function at the camera. However, real video cameras have controls that can alter the transfer function. These controls are routinely used by cinematographers and videographers to achieve their artistic intents. Obviously the artistic intention of the cinematographer must be imposed at the camera, not at the display – it ought to be the *displays* that are standardized, not the cameras! But there is no mechanism to impose standards on displays, so we standardize the reference transfer function at the camera instead. In effect, Rec. 709 is standardized so as to produce acceptable reproduction on a conventional display. Despite the lack of standards, CRT displays are tacitly considered to have similar response.

The engineering of video systems – and, by extension, of desktop computer systems – involves an implicit assumption about the 2.5-power function of the monitor. Alternate display devices, such as LCDs, plasma panels, DMDs, and so on, do not have the 2.5-power function of the CRT. But the most important aspect of image coding is the establishment of a nearly perceptually-uniform image code. In a closed system employing an alternate display technology, you might be tempted to use of a transfer function at encoding that is the inverse of the transfer function at the display. However, if the transfer function of the display was very different than a 2.5-power function, more than 8 bits would be required to code luminance.

More significantly, there is a huge installed base of encoding and decoding equipment that assumes image coding similar or identical to that of video. The installed base includes roughly 1,300,000,000 television receivers, 400,000,000 VCRs, 250,000,000 camcorders, and 300,000,000 desktop computers. These devices are all, in effect, wired to directly reproduce *R'G'B'* signals encoded according to Rec. 709. Any proposal for a new encoder transfer function would compromise the interchange of images among these systems.

Poynton, Charles, "Luminance, luma, and the migration to DTV," presented at *32nd SMPTE Advanced Motion Imaging Conference*, Toronto (Feb. 6, 1998).

I have discussed the reproduction of black-and-white images. These concepts extend into the domain of luma and color difference coding, used in video, JPEG, and MPEG. At a SMPTE conference, I discussed the effect of transfer functions in luma and color difference coding.

## Pseudocolor, hicolor, and truecolor

The block diagrams of pseudocolor, hicolor, and truecolor systems used in desktop computing are sketched in Figures A.11, A.12, and A.13 opposite. These sketches show the hardware pipeline from the framebuffer to the monitor. The interface from application software to the graphics subsystem (and window system) assumes the same processing. Comparable processing is implicit in file formats for pseudocolor and truecolor images. (File formats for hicolor are rare.)

A pseudocolor image is always accompanied by its color lookup table (CLUT). The CLUT may be optimized for the particular image, or it may contain a *system palette*. Upon display of a pseudocolor image, the graphics subsystem may directly load the colormap that accompanies the image. Alternatively, the graphics subsystem may recode the image

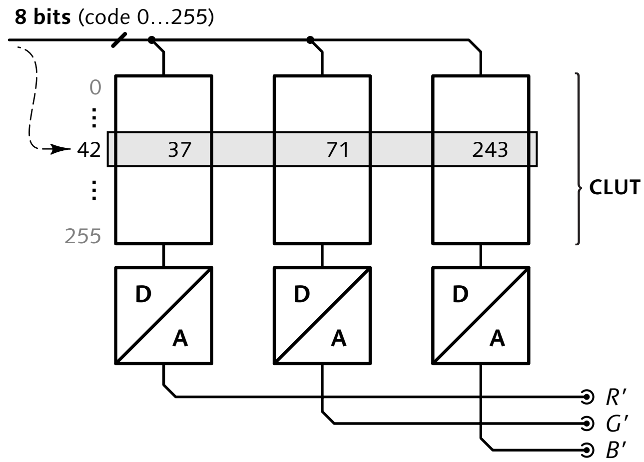



Figure A.11 **Pseudocolor (8-bit)** graphics systems are common in low-end PCs. For each pixel, the framebuffer stores a color index value (typically 8 bits). Each index value is mapped, through a *color lookup table* (CLUT) that is part of the display hardware, to a triplet of *R′G′B′* codes. When a pixel is accessed from the framebuffer, the corresponding triplet is accessed from the CLUT; those values are applied to the digital-to-analog converter (DAC). *R′G′B′* codes from the CLUT translate linearly into voltage applied to the monitor, so code values are comparable to video *R′G′B′* values – the *R′G′B′* values are proportional to displayed intensity raised to the 0.4 power, comparable to video *R′G′B′* codes.

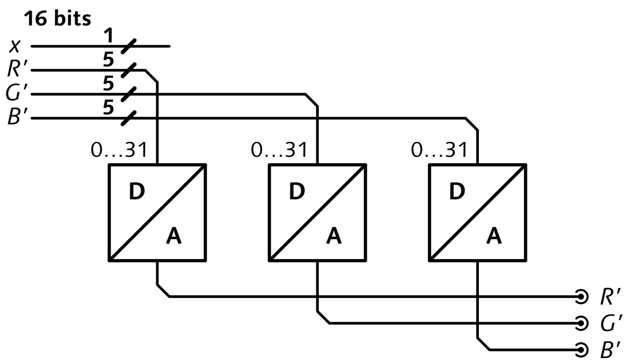



Figure A.12 **Hicolor (16-bit)** graphics systems store, in the framebuffer, *R′G′B′* codes partitioned into three components of five bits each (5-5-5), or partitioned five bits for red, six bits for green, and five bits for blue (5-6-5). In low-end systems, these codes are applied directly to the DACs with no intervening colormap. Because the *R′G′B′* codes are translated linearly into monitor voltage, the code values are implicitly proportional to displayed intensity raised to the 0.4 power.

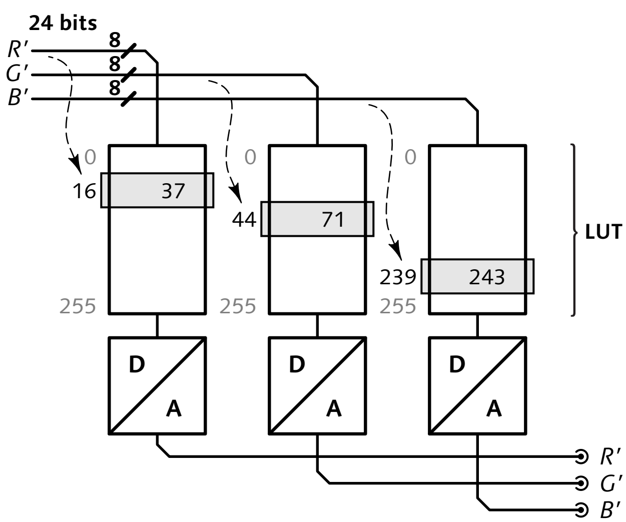



Figure A.13 **Truecolor (24-bit)** graphics systems store 8 bits for each of the red, green, and blue components. Truecolor systems usually implement a set of *lookup tables* (LUTs) between the framebuffer memory and the DACs. The LUTs allow a transfer function to be imposed between the *R′G′B′* codes and the DACs: The *R′G′B′* values in the framebuffer need not be related to displayed intensity raised to the 0.4 power. Application software or system software can impose arbitrary functions. In order for the application software to provide the same default behavior as low-end pseudocolor and hicolor graphics systems, each LUT is set by default to a *ramp* corresponding to the identity (or *unity*) function.

according to some other map that is already loaded into the hardware, or according to the system palette native to the application. Recoding of pseudocolor image data may introduce color errors. Pseudocolor image data is always coded in terms of monitor *R′G′B′* – that is, pseudocolor image colors are implicitly perceptually coded.

A hicolor system has no lookup tables. Image data is coded in terms of monitor *R′G′B′*: Image data is implicitly perceptually coded (though coarsely quantized).

In truecolor, each of the *RGB* channels is associated with a lookup table (LUT) that applies a transfer function. (Ordinarily, the three tables have identical contents.) Different default lookup tables are in use, for different platforms. Truecolor image files are ordinarily stored without any lookup tables; most truecolor file formats make little or no provision for conveying the transfer function that is expected at display.

Hicolor and truecolor display hardware can typically be operated in pseudocolor mode. But this mode switch applies to the whole display. If a pseudocolor image is to be displayed in a window of a display that is operating in hicolor or truecolor mode, the graphics subsystem must perform the pseudocolor color lookup operation in software. If the truecolor system is operating with a LUT that is not a ramp, then the *RGB* codes from the pseudocolor CLUT must be mapped through the *inverse* of the truecolor LUT prior to being stored in the framebuffer.

If a hicolor or truecolor image is to be presented on a display that is operating in pseudocolor mode, the graphics subsystem must find, for each hicolor or truecolor pixel (*RGB* triplet), the index of the closest color that is available in the CLUT currently in use. If the CLUT is organized systematically, then this operation can be fairly rapid; if the CLUT is unstructured, then the conversion proceeds slowly. The translation to pseudocolor causes coarse quantization of the image colors. Dithering may be applied, to spread the quantization error over a small area of the image. In any event, colormap quantization generally causes serious degradation of color fidelity.

A generic application program on a PC must assume the lowest common denominator of display capability: It must be prepared to operate without a lookup table. (Even some graphics cards with 24-bit capability have no lookup tables.) Even if a hardware lookup table is present, PC software generally operates as if there is no table. If a LUT is present, it is ordinarily loaded with a ramp so that it has no effect. Image data exchanged among PCs is therefore coded as monitor *R′G′B′*. Though the situation arose by accident, this is quite comparable to video coding, and is nearly optimal for perception! Image data that originates on (or is intended for display on) a PC carries the implicit assumption that the lookup table contains a ramp, that is, that the image data is represented in gamma-corrected monitor *R′G′B′*. So Figure A.10 applies to video *and* to PCs: the coding is comparable.

## Macintosh gamma

Apple Computer, Inc., *Inside Macintosh* (Reading, Mass.: Addison-Wesley-Longman, 1992). 27 volumes.

Contrary to popular belief, Macintosh computers use monitors that have the same physics as monitors used in video systems and other brands of computers. Though it is nowhere documented in the 27 volumes of the *Inside Macintosh* series of books, the QuickDraw graphics subsystem loads an unusual transfer function into the lookup tables of a Mac. The default lookup table (in hexadecimal code) is this:

Figure A.14 **Default Macintosh LUT**

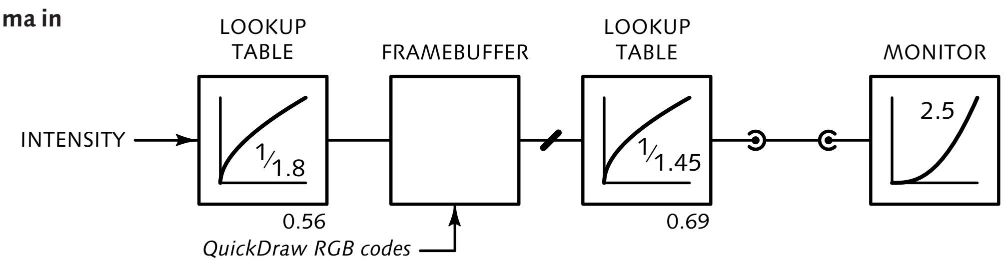
```
00: 00, 05, 09, 0B, 0E, 10, 13, 15, 17, 19, 1B, 1D, 1E, 20, 22, 24
10: 25, 27, 28, 2A, 2C, 2D, 2F, 30, 31, 33, 34, 36, 37, 38, 3A, 3B
20: 3C, 3E, 3F, 40, 42, 43, 44, 45, 47, 48, 49, 4A, 4B, 4D, 4E, 4F
30: 50, 51, 52, 54, 55, 56, 57, 58, 59, 5A, 5B, 5C, 5E, 5F, 60, 61
40: 62, 63, 64, 65, 66, 67, 68, 69, 6A, 6B, 6C, 6D, 6E, 6F, 70, 71
50: 72, 73, 74, 75, 76, 77, 78, 79, 7A, 7B, 7C, 7D, 7E, 7F, 80, 81
60: 81, 82, 83, 84, 85, 86, 87, 88, 89, 8A, 8B, 8C, 8C, 8D, 8E, 8F
70: 90, 91, 92, 93, 94, 95, 95, 96, 97, 98, 99, 9A, 9B, 9B, 9C, 9D
80: 9E, 9F, A0, A1, A1, A2, A3, A4, A5, A6, A6, A7, A8, A9, AA, AB
90: AB, AC, AD, AE, AF, B0, B0, B1, B2, B3, B4, B4, B5, B6, B7, B8
A0: B8, B9, BA, BB, BC, BC, BD, BE, BF, C0, C0, C1, C2, C3, C3, C4
B0: C5, C6, C7, C7, C8, C9, CA, CA, CB, CC, CD, CD, CE, CF, D0, D0
C0: D1, D2, D3, D3, D4, D5, D6, D6, D7, D8, D9, D9, DA, DB, DC, DC
D0: DD, DE, DF, DF, E0, E1, E1, E2, E3, E4, E4, E5, E6, E7, E7, E8
E0: E9, E9, EA, EB, EC, EC, ED, EE, EE, EF, F0, F1, F1, F2, F3, F3
F0: F4, F5, F5, F6, F7, F8, F8, F9, FA, FA, FB, FC, FC, FD, FE, FF
```

This table contains a pure power function with an exponent of $^{1}/_{1.45}$. Image data that originates on – or is intended for display on – a Macintosh computer carries the implicit assumption that the lookup table contains this function. This default lookup table, in combination with a conventional monitor, causes the *R*, *G*, and *B* values presented to QuickDraw to represent the $^{1}/_{1.8}$-power of luminance.

Although Apple has historically failed to publish any meaningful documentation of gamma, a Macintosh is widely considered to have a default gamma of 1.8. This *de facto* nomenclature was established by the *Gamma* control panel, by Knoll Software, which was distributed with Adobe Photoshop up to and including version 4.

Figure A.15 below summarizes the gamma situation for Macintosh.

Figure A.15 **Gamma in Macintosh**




Poynton, Charles, *A Technical Introduction to Digital Video* (New York: Wiley, 1996).

In the *Gamma* chapter of my book, I explain the *dot gain* phenomenon of offset printing. Offset printing uses code values proportional to the 1.8-power of reflectance. But QuickDraw *RGB* codes are related to

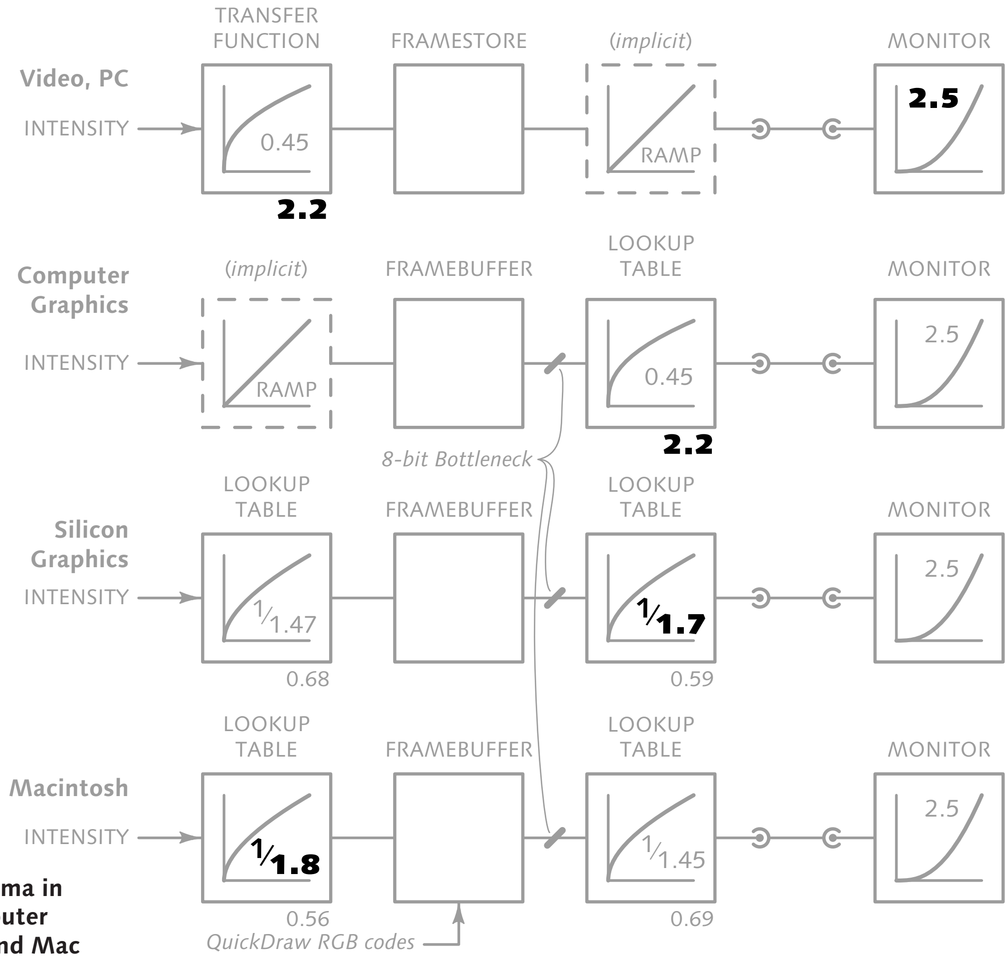



Figure A.16 **Gamma in video, PC, computer graphics, SGI, and Mac**

luminance by the 1.8 power! Though the situation arose by accident, QuickDraw *RGB* coding is well suited to offset printing, and it is ubiquitous in desktop publishing and prepress. (QuickDraw coding is also widely used in multimedia, and on the World-wide Web, though in these applications it is not as suitable as coding according to Rec. 709.)

## System issues

Figure A.16 above collects the three gamma sketches already presented (video, computer-generated imagery, and Macintosh), and adds a fourth sketch, for Silicon Graphics (SGI). Given the diverse transfer functions, it is no surprise that it is difficult to exchange image data. I have indicated in bold type the numerical quantity that is referred to as *gamma* in each of these four domains: You can see that the *gamma* number is applied in four different places! So even if you know the gamma value, it is difficult to determine where it is applied! As I have mentioned, a Macintosh is considered to have a default gamma of 1.8. An SGI computer has a default gamma of 1.7. But Figure A.16 shows that these two numbers are not comparable!

The graphics subsystems of most computers allow the lookup table to be changed. On a Mac, the *Gamma* control panel accomplishes this. When a *gamma* value of *g* is specified, the control panel loads the Mac lookup table with a power function whose exponent is $g/2.61$. When a *gamma* value of *g* is specified to an SGI computer, system software loads the SGI lookup table with a power function whose exponent is $^{1}/_{g}$. The convention differs from that on a Mac. To program an SGI computer to behave like a Mac, you must set SGI gamma to 1.45.

Image data in the framebuffer is not usually changed upon a change of the lookup table. Any time you jam a particular value of *gamma* into the back end of the graphics subsystem, you override assumptions that may have been made about the color interpretation of image data. When you change gamma, the colors of displayed objects (icons, menus, and windows) and the colors of displayed images, will change!

## Computer graphics standards

To exchange images using computer graphics standards requires knowledge of the transfer function. Standards such as PHIGS and CGM stem from computer-generated imagery (CGI), where linear-light (tristimulus) coding is the norm, and in PHIGS and CGM it is implicit that *RGB* data is coded in linear-light (tristimulus).

JPEG and other lossy image compression algorithms depend on discarding information that won't be *perceived*. It is vital that the data presented to a JPEG compressor be coded in a perceptually-uniform manner, so that the discarded information has minimal perceptual impact. In practice, JPEG works well only on nonlinearly-coded (gamma-corrected) image data.

But nowhere in the PHIGS, CGM, or JPEG standards is gamma or transfer function explicitly mentioned, and nowhere in the data streams or image file formats for PHIGS, CGM, or JPEG, is the transfer function conveyed! The user must handle the transfer function or face poor image quality. If image data is transferred between these systems without regard for the transfer function, then the pictures will have terrible quality. Figure A.17 below summarizes the situation: *RGB* codes [128, 128, 128] produce completely different intensities at the face of the screen in PHIGS (or CGM) and JPEG. But the standards themselves provide absolutely no information concerning this issue.

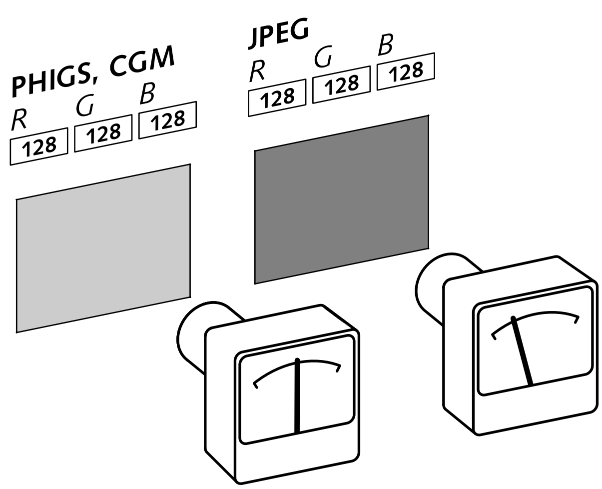
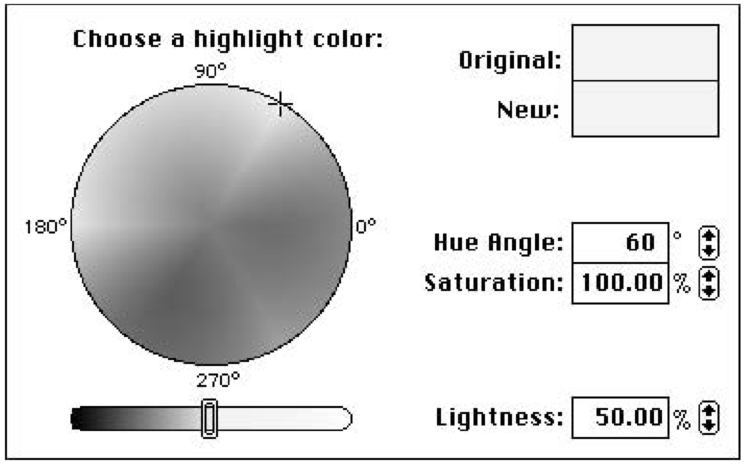



Figure A.17 **Gamma in PHIGS, CGM, and JPEG.** In the PHIGS and CGM standards there is no mention of gamma or transfer function, but it is implicit that image data is coded as linear-light tristimulus values. In the JPEG standard there is no mention of gamma or transfer function, but it is implicit that image data is gamma-corrected.

Figure A.18 **Lightness in HSL.** *Lightness*, in its CIE definition, is a perceptual quantity. However, in the textbook *HSL* color representation used in computer graphics – here exemplified by Apple's Macintosh Color Picker – no account is taken of transfer function. Apple implies that all of the shades in the disk have the same lightness of 50%. Does the disk appear uniformly shaded to you?




Many other computer graphics standards ignore or discount transfer functions. Figure A.18 above shows a screenshot of the Apple Macintosh Color Picker, which implements the textbook *HSL* representation of color. This presentation implies that all of the colors shown share the same lightness value, but clearly the disk is not uniformly shaded. The *HSL* representation has no objective basis in color science.

## World-wide web

The World-wide web uses GIF and JPEG file formats to convey images. (Other file formats are in use, but none of these are widely deployed.)

A GIF file represents an image in pseudocolor form. A web browser operating on a pseudocolor display does not attempt to reconcile the potentially conflicting CLUTs found among the several (or several dozen) GIF images that might share a window on the user's monitor. Instead, a browser typically recodes every pseudocolor image into a *browser palette* comprising a 6×6×6 *colorcube* of monitor *R'G'B'* codes. The browser palette comprises 216 colors. (To display GIF images on a hicolor or truecolor system, the browser's graphics subsystem uses each file's CLUT to translate each image to *R'G'B'*.)

JPEG image coding is based on truecolor. The JFIF specification is the *de facto* standard for JPEG file interchange. JFIF is unclear concerning the handling of transfer function. In practice, an image is encoded into JPEG using the encoding transfer function that is in effect for the platform that it is encoded on. A decoded JPEG image is displayed using the transfer function in effect on the platform upon which it is decoded. Figure A.19 overleaf sketches the gamma situation for JPEG on the web. It's chaos! Image data is exchanged among platforms without regard for the transfer function that will be applied upon display. The same file displays differently on different platforms!

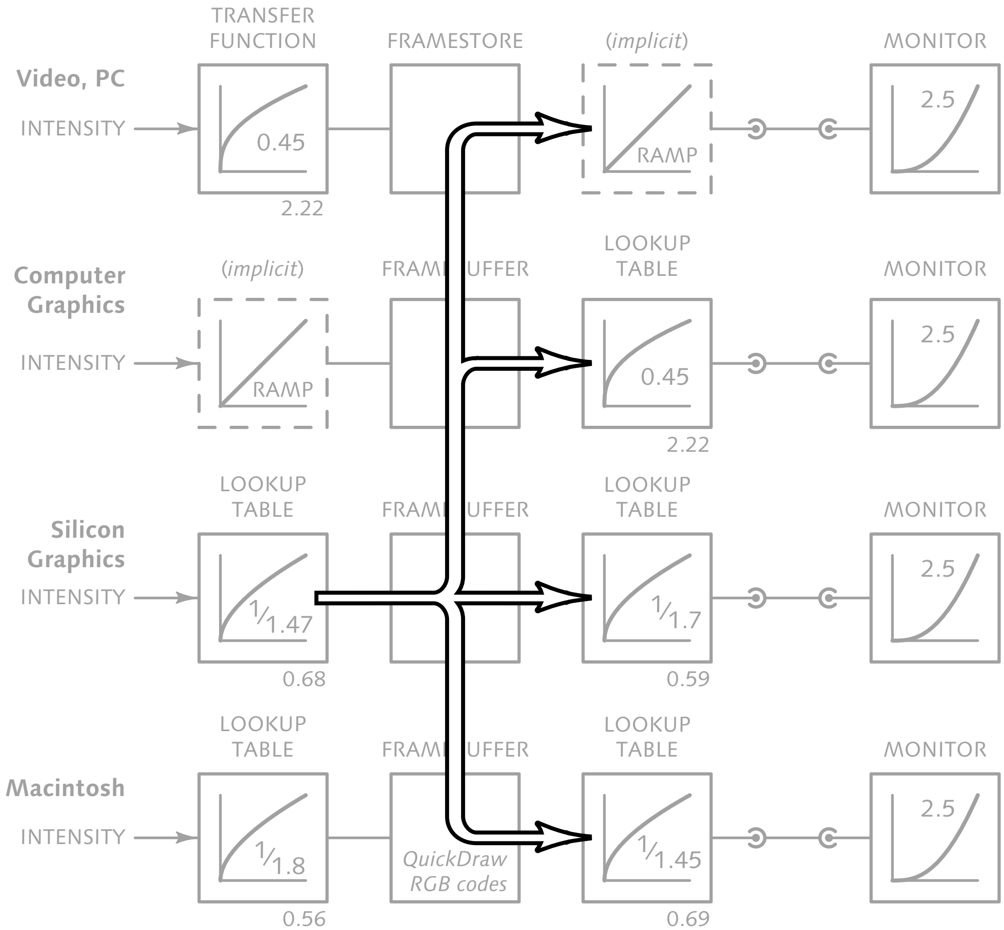



Figure A.19 **Gamma on the web.** Image data is exchanged on the web without regard for the transfer function that will be applied at display. Consequently, the same image is displayed differently on different platforms. Here, the image originates on an SGI computer. In practice, the dominant platforms are PC and Mac.

In Figure A.19 above, I show an image originating on an SGI computer. You can see that it is decoded in four different ways; in particular, it is reproduced incorrectly on PC and Mac, the dominant viewing platforms. If a JPEG image originates on a PC, it displays incorrectly on a Mac; if a JPEG image originates on a Mac, it displays incorrectly on a PC. The relatively low penetration of Macintosh computers in the marketplace might suggest that image origination should be optimized for PCs: let the Mac browsers do what they will. But the tools for image preparation and web page creation are much more capable on Macs than on PCs. A large fraction of web images are prepared on Macs, so the implicit Mac transfer function is important.

Gille, J.L., J.O. Larimer, and J. Luszcz, "Error diffusion using the 'web-safe' colors: how good is it across platforms?", in Rogowitz, B.E. and T.N. Pappas (eds.), *Human Vision and Electronic Imaging III*, Proceedings of SPIE, Volume 3299, 368–375 (Bellingham, Wash.: SPIE, 1998).

The second row of the sketch shows the CGI situation. In practice, images are rarely displayed through a LUT configured for CGI. But the VRML language stems from CGI, and shading in synthetic computer graphics has historically been performed in the linear-light domain. (Shading in nonlinear domains is poorly understood.) VRML is best considered to originate in the computer graphics row of the sketch.

## Color management

Work is underway to implement facilities in graphics systems to allow device-independent specification of color. Users and applications will be able to specify colors, based on the CIE standards, without concern for gamma correction. Color image files will be tagged with their transfer functions (along with other color parameters). When this transition is complete, it will be much easier to obtain color matching across different graphics libraries and different hardware.

However, these developments will not render gamma irrelevant. Proper use of transfer functions will remain necessary in order to code images in a perceptual manner, so as to achieve maximum performance from a reasonable number of bits per component – 8, say! Also, it will take a long time for this technology to be deployed.

In the meantime, if you are a programmer, you can take the following steps:

- Use gamma-corrected *R'G'B'* representations whenever you can. An image coded with gamma correction has good perceptual uniformity, resulting in an image with much higher quality than one coded as 8-bit luminance (or tristimulus) values.
- When you exchange images either in truecolor or pseudocolor form, code *R'G'B'* color values using the Rec. 709 transfer function.
- In the absence of reliable information about your monitor, display pictures assuming a monitor gamma value of 2.5.

If you are a user, you can take these steps:

- Establish good viewing conditions. If you are using a CRT display, you will get better image quality if your overall ambient illumination is reduced.
- Ensure that your monitor's BLACK LEVEL (or BRIGHTNESS) control is set to correctly reproduce black elements on the screen. Consult my note *"BRIGHTNESS" and "CONTRAST" controls*, available on the Internet.
- Demand, from your hardware and software developers and vendors, that they document how they handle transfer functions.

**Charles Poynton**
tel +1 416 413 1377
poynton@poynton.com
www.poynton.com

# *YUV* and *luminance* considered harmful B

This is a plea for precise terminology in video. The notation *YUV*, and the term *luminance*, are widespread in digital video. In truth, digital video almost never uses *Y'UV* color difference components, and never directly represents the *luminance* of color science. The common terms are almost always wrong. This note explains why. I urge video engineers and computer graphics specialists to use the correct terms, almost always $Y'C_BC_R$ and *luma*.

## Cement vs. concrete

I'll demonstrate by analogy why it is important to use correct terms. Next time you're waiting in line for a bus, ask the person next to you in line what building material is used to construct a sidewalk. Chances are that person will answer, "cement."

The correct answer is *concrete*. Cement is calcined lime and clay, in the form of a fine, gray powder. Cement is one ingredient of concrete; the other ingredients are sand, gravel, and water.

In an everyday situation, you need not be precise about which of these terms are used: If you refer to a bridge as being constructed of "cement," people will know what you mean. Lay people are not confused by the term "cement." Interestingly, experts are not confused either. If a bridge superintendent yells out to his foreman, "Get me 500 pounds of cement!" the foreman understands immediately from context whether the superintendent actually wants concrete. However, if you place an order with a building material supplier for 500 pounds of cement, you will certainly not receive 500 pounds of concrete! Lay people have no trouble with the loose nomenclature, and the experts have little trouble. It is the people in the middle who are liable to become confused by loose nomenclature. Worse still, they are liable to use a term without realizing that it is ambiguous or wrong!

### True CIE luminance

The principles of color science dictate that true CIE luminance – denoted $Y$ – is formed as a weighted sum of linear (tristimulus) *RGB* components. If CIE luminance were transmitted in a video system, the system would conform to the *Principle of Constant Luminance*. But in video we implement an engineering approximation that departs from this principle. It was standardized for NTSC in 1953, and remains standard for all contemporary video systems, to form *luma*, denoted $Y'$, as a weighted sum of nonlinear (gamma-corrected) $R'G'B'$ components. The nonlinear transfer function is roughly comparable to a square root. To form *luma*, we use the theoretical coefficients of color science, but we use them in a block diagram different from that prescribed by color science: As detailed in my book, gamma correction is applied *before* forming the weighted sum, not after. The "order of operations" is reversed from what you might expect from color science.

$$^{601}Y' = 0.299\,R' + 0.587\,G' + 0.114\,B'$$

Poynton, Charles, *Digital Video and HDTV Algorithms and Interfaces* (San Francisco: Morgan Kaufmann, 2003).

### The misinterpretation of luminance

Video engineers in the 1950s recognized that the video quantity $Y'$ was very different from CIE luminance, and that it needed to be distinguished from luminance. They described it by the phrase *the quantity representative of luminance*. They used the symbol $Y$, but augmented it with a prime to denote the nonlinearity: $Y'$. Obviously the qualifier "quantity representative of" was cumbersome, and over the decades, it was elided. And over time, the prime symbol was elided as well. Unfortunately, no new word was invented to supplement *luminance*, to reinforce the distinction between the color science quantity and the video quantity. Most video engineers nowadays are unfamiliar with color science, and most do not understand the distinction. Engineers today often carelessly use the word *luminance*, and the symbol $Y$, to refer to the weighted sum of nonlinear (gamma-corrected) $R'G'B'$ components.

The sloppy nomenclature made its way into ostensibly authoritative video references, such as Pritchard's SMPTE paper published in 1977.

Pritchard, D.H., "U.S. Color Television Fundamentals – A Review," in *SMPTE Journal*, v. 86 (Nov. 1977), 819–828.

The computer graphics pioneer Alvy Ray Smith encountered the word *luminance* in his quest to adapt video principles to computer graphics. Smith apparently correlated the use of the term *luminance* with his knowledge of color science, and understandably – though mistakenly – concluded that video "luminance" and color science luminance were identical. Consequently, video *YIQ* was introduced to computer graphics, having its $Y$ component alleged to be identical to CIE luminance.

Smith, A.R., "Color Gamut Transform Pairs," in *Computer Graphics*, v. 12, n. 2 (Aug. 1978, Proc. *SIGGRAPH 78*), 12–19.

Foley, James D., and Andries van Dam, *Fundamentals of Interactive Computer Graphics* (Reading, Mass.: Addison-Wesley, 1984).

Foley, James D., Andries van Dam, Steven Feiner, and John Hughes, *Computer Graphics: Principles and Practice*, Second Edition (New York: Addison-Wesley, 1990).

That incorrect interpretation propagated into authoritative computer graphics textbooks. *Computer Graphics: Principles and Practice*, on page 589, Section 13.3.3, *The YIQ Color Model*, states:

> The *Y* component of YIQ is not yellow but luminance, and is defined to be the same as the CIE **Y** primary.

(The emphasis is in the original. "Yellow" refers to *CMY*, which was mentioned in the immediately preceding section. "CIE **Y** primary" would be more accurately denoted "CIE **Y** component.")

As you have seen, the so-called $Y$ component of video – more properly designated with a prime symbol, $Y'$ – is *not* the same as CIE luminance. Video $Y'$ cannot even be computed from CIE $Y$, unless two other color components are also available. The quoted passage is quite wrong. About 300,000 copies of various editions and adaptations of *CG:PP* have been printed. Confusion is rampant.

Pratt, William K., *Digital Image Processing*, Second Edition (New York: Wiley,1991). p. 64.

The error also propagated into the digital image processing community. A widely used book in that field states:

> N.T.S.C. formulated a color coordinate system for transmission composed of three tristimulus values $YIQ$. The $Y$ tristimulus value is the luminance of a color.

The video quantities are certainly *not* tristimulus values, which are, by CIE's definition, proportional to intensity.

Loose nomenclature on the part of video engineers has misled a generation of digital image processing, computer software, and computer hardware engineers.

### The enshrining of luma

I campaigned for adoption of the term *luma* to designate the nonlinear video quantity. The term had no pre-existing meaning, and by virtue of its being different from *luminance*, it invites readers from other domains to investigate fully before drawing conclusions about its relationship with luminance.

With the help of Fred Kolb, my campaign succeeded: In 1993, SMPTE adopted Engineering Guideline EG 28, *Annotated Glossary of Essential Terms for Electronic Production*. EG 28 defines the term *luma*, and clarifies the two conflicting interpretations of the term *luminance*. While a SMPTE EG is not quite a SMPTE "Standard," at long last the term has received official recognition. There's no longer any excuse for sloppy use of the term *luminance* by the authors of video engineering papers.

It is a shame that today's SMPTE and ITU-R standards for digital video persist in using the incorrect word *luminance,* without ever mentioning the ambiguity – even conflict – with the CIE standards of color science.

### Color difference scale factors

When I say *NTSC* and *PAL*, I refer to color encoding, not scanning: I do not mean 525/59.94 and 625/50.

To represent color, luma is accompanied by two *color difference* – or *chroma* – components, universally based on *blue minus luma* and *red minus luma*, where blue, red, and luma have all been subject to gamma correction: $B'-Y'$ and $R'-Y'$. Different scale factors are applied to the basic $B'-Y'$ and $R'-Y'$ components for different applications. $Y'P_BP_R$ scale factors are optimized for component analog video. $Y'C_BC_R$ scale factors are optimized for component digital video such as 4:2:2 studio video, JPEG, and MPEG. Kodak's PhotoYCC ($Y'C_1C_2$) uses scale factors optimized to record the gamut of film colors. $Y'UV$ and $Y'IQ$ use scale factors optimized to form composite NTSC and PAL video.

ITU-R Rec. BT.601, *Studio encoding parameters of digital television for standard 4:3 and wide-screen 16:9 aspect ratios* (Geneva: ITU).

$Y'C_BC_R$ scaling as defined by Rec. 601 is appropriate for component digital video. $Y'C_BC_R$ chroma is almost always subsampled using one of three schemes: 4:2:2, or 4:2:0, or 4:1:1.

$Y'UV$ scaling is properly used only as an intermediate step in the formation of composite NTSC or PAL video signals. $Y'UV$ scaling is not appropriate when the components are kept separate. However, the $Y'UV$ nomenclature is now used rather loosely, and sometimes – particularly in computing – it denotes *any* scaling of $B'-Y'$ and $R'-Y'$.

Digital disk recorders (DDRs) are generally able to transfer files across Ethernet. Abekas introduced the convention of using an extension "**.**yuv" for these files. But the scale factors – in Abekas equipment, at least – actually correspond to $Y'C_BC_R$. Use of the "**.**yuv" extension reinforces the misleading *YUV* nomenclature.

Chroma components are properly ordered $B'-Y'$ then $R'-Y'$, or $C_B$ then $C_R$. Blue is associated with $U$, and red with $V$. $U$ and $V$ are in alphabetic order.

Subsampling is properly performed only on component digital video, that is, on $Y'C_BC_R$. Subsampling is inappropriate for $Y'UV$. If you see a system described as *Y'UV 4:2:2*, you have a dilemma. Perhaps the person who wrote the description is unfamiliar with the principles of component video, and the scale factors actually implemented in the equipment (or the software) are correct. But you must allow for the possibility that the engineers who designed or implemented the system used the wrong scale factors! If the wrong equations were used, then color accuracy will suffer; however, this can be difficult to diagnose.

Hamilton, Eric, *JPEG File Interchange Format*, Version 1.02 (Milpitas, Calif.: C-Cube Microsystems, 1992).

Proper $Y'C_BC_R$ scaling is usual in Motion-JPEG, and in MPEG. However, the $Y'C_BC_R$ scaling used in stillframe JPEG/JFIF in computer applications usually uses "full-range" luma and chroma excursions, without any headroom or footroom. The chroma excursion is $^{254}/_{255}$ of the luma excursion. The scaling is almost exactly that of $Y'P_BP_R$, but is unfortunately described as $Y'C_BC_R$: Now even $Y'C_BC_R$ is ambiguous! I am hopeful that proper $Y'C_BC_R$ scaling will be incorporated into the next revision of JFIF, so that compressed stillframe and motion imagery in computing can be combined without suffering a conversion process.

Except for very limited use in the encoding and decoding of composite $4f_{SC}$ (or loosely, "D-2") studio video, $Y'IQ$ coding is obsolete.

## Conclusion: A plea

Using the term *luminance* for video $Y'$ is tantamount to using the word *cement* instead of *concrete* to describe the main construction material of a bridge. Lay people don't care, and experts can live with it, but people in the middle – in this case, the programmers and engineers who are reimplementing video technology in the computer domain – are liable to draw the wrong conclusions from careless use of terms. Users suffer from this, because the exchange of images is compromised.

I urge video engineers and computer graphics specialists to avoid *YUV* and *luminance*, and to use the correct terms, $Y'C_BC_R$ and *luma*.

**Discreet Logic**

10, Rue Duke
Montréal, Québec
Canada H3C 2L7
Tel: (514) 393-1616
Fax: (514) 393-0110

www.discreet.com

# Merging computing with studio video: Converting between *R′G′B′* and 4:2:2 C

Charles Poynton
www.poynton.com

**Abstract**

In this paper, I explain the *R′G′B′* and $Y'C_BC_R$ 4:2:2 representations, and explain the technical aspects of conversion between the two. I conclude by suggesting steps that can be taken during production and post-production to avoid difficulty with the conversion.

Film, video, and computer-generated imagery (CGI) all start with red, green, and blue (*RGB*) tristimulus components proportional to intensity – "linear light." A nonlinear transfer function is applied to *RGB* to give *gamma corrected R′G′B′*. This is the native color representation of video cameras, computer monitors, video monitors, and television.

The human visual system has poor color acuity. If *R′G′B′* is transformed into luma and chroma, then color detail can be discarded without the viewer noticing. This enables a substantial saving in data capacity – in "bandwidth," or in storage space. Because studio video equipment has historically operated near the limit of realtime capture, recording, processing, and transmission capabilities, the subsampled $Y'C_BC_R$ 4:2:2 format has been the workhorse of studio video for more than a decade.

The disadvantage of 4:2:2 is its lossy compression. Upon "matrixing" from 8-bit *R′G′B′* to 8-bit $Y'C_BC_R$, three-quarters of the available colors are lost. Upon 4:2:2 subsampling, half the color detail is discarded. However, production staff are facing increasing demands for quality, and increasing demands to integrate video production with film and CGI. The lossy compression of 4:2:2 is becoming a major disadvantage.

Owing to the enormous computing and storage capacity of general-purpose workstations, it is now practical to do production directly in *R′G′B′* (or as it is known in studio video terminology, 4:4:4). To integrate traditional studio video equipment into the new digital studio, conversion between *R′G′B′* and 4:2:2 is necessary.

## Introduction

Linear light *RGB* is the native color coding of CGI. In computing, the gamut of colors comprises the volume bounded by the unit *RGB* cube: See Figure C.1 opposite. In video and computer graphics, a nonlinear transfer function is applied to *RGB* tristimulus signals to give *gamma corrected R'G'B'*, often in 8 bits each. See Figure C.2, on page 74.

If *R'G'B'* is transformed into luma and color difference components, $Y'C_BC_R$, then color detail can be subsampled (lowpass filtered) without the viewer noticing. This leads to a substantial saving in data capacity – in "bandwidth," or in storage space. Subsampling in $Y'C_BC_R$ involves a "visually lossless" lossy compression system. The 4:2:2 scheme has a compression ratio of 1.5:1, and the 4:2:0 and 4:1:1 schemes have compression ratios of 2:1. The subsampled $Y'C_BC_R$ 4:2:2 representation of Rec. 601 is standard in studio digital video. However, $Y'C_BC_R$ has several problems in the digital studio:

ITU-R Rec. BT.601, *Studio encoding parameters of digital television for standard 4:3 and wide-screen 16:9 aspect ratios* (Geneva: ITU).

**Codeword utilization** in $Y'C_BC_R$ is very poor. *R'G'B'* coding with 8 bits per component allows every one of the $2^{24}$ combinations, or 16 million codewords, to represent a color. Theoretically, $\frac{3}{4}$ or more of the "legal" $Y'C_BC_R$ code combinations do not represent colors! In 8-bit Rec. 601 standard $Y'C_BC_R$, only 17% of the codewords represent colors. $Y'C_BC_R$ has fewer colors – or equivalently, more quantization noise, or poorer signal-to-noise ratio (SNR) – than *R'G'B'*.

The designation *D-1* is sometimes loosely applied to 4:2:2. However, *D-1* properly refers to a particular DVTR format, not to an interface standard.

**Filtering and subsampling** operations that form the 4:2:2 signal remove chroma detail. If subsampling is accomplished by simply dropping or averaging alternate $C_B$ and $C_R$ samples, then filtering artifacts (such as aliasing) will be introduced. Artifacts can accumulate if filtering is repeated many times. Subsampling using a sophisticated filter gives much better results than simply dropping or averaging samples. However, even sophisticated filters can exhibit fringing on certain color edges, if conversion between *R'G'B'* and 4:2:2 is repeated many times.

**Loss of color detail** makes it more difficult to pull bluescreen or greenscreen mattes from 4:2:2 than from *R'G'B'*.

**Test signals** characterize the electrical performance of a video system. Standard video test signals include elements that are synthesized electronically as sine waves, and injected onto the signal. Many of these elements have no legitimate *R'G'B'* representation. Since these signals can be conveyed through $Y'C_BC_R$ without incident, some people claim $Y'C_BC_R$ to have an advantage. However, in my opinion, it is more important to allocate bits to picture information than to signals that cannot possibly represent picture information.

In general, $Y'C_BC_R$ is optimized for realtime video, at the expense of more difficult interface with film, CGI, and general-purpose computer tools. *R'G'B'* does not exploit chroma subsampling, so it has somewhat higher data capacity requirements than $Y'C_BC_R$.

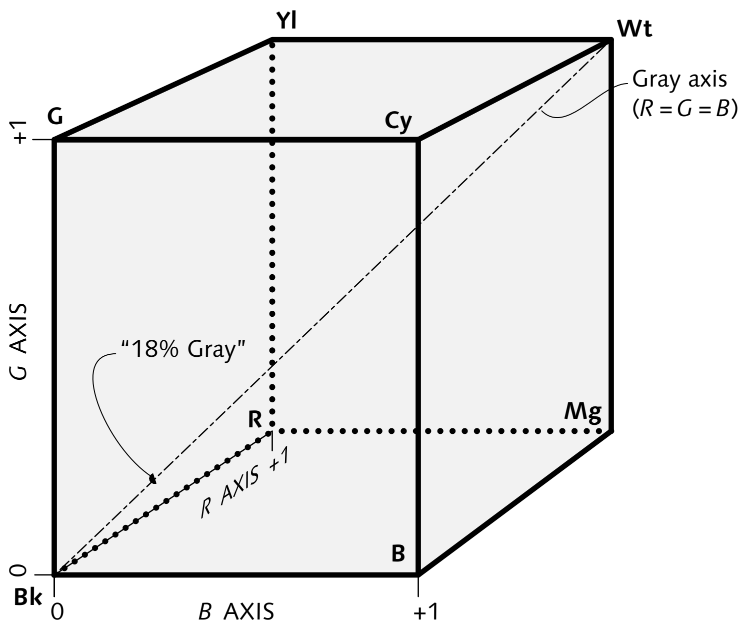
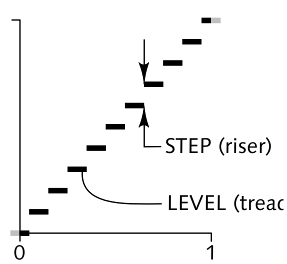



Figure C.1 **RGB unit cube** encompasses linearly-coded *RGB* tristimulus values, each proportional to intensity. This scheme is poorly matched to the lightness sensitivity of vision.

## Computing gamut




Linear light coding is used in CGI, where physical light is simulated. However, linear light coding performs poorly for images to be viewed. The best perceptual use is made of the available bits by using nonlinear coding that mimics the nonlinear lightness response of human vision. In the storing and processing of images, linear light coding is rarely used. In the display of images, linear light coding is never used. In video, computing, and many other domains, a nonlinear transfer function is applied to *RGB* tristimulus signals to give nonlinearly-coded (or *gamma corrected*) components, denoted with prime symbols: *R'G'B'*.

In an 8-bit system with nonlinear coding, each of *R'*, *G'*, and *B'* ranges from 0 through 255, inclusive. Each component has 255 *steps* (risers) and 256 *levels*: A total of $2^{24}$ colors – that is, 16777216 colors – are representable. Not all of them can be distinguished visually; not all are perceptually useful; but they are all colors. See Figure C.2 overleaf.

## R'G'B' in video

Studio video *R'G'B'* standards provide footroom below the black code, and headroom above the white code. The primary purpose of footroom and headroom is to accommodate the transients that result from filtering in either the analog or digital domains. Their secondary purpose is to provide some margin to handle level variations in signals originated in the analog domain. (Additionally, the headroom provides a marginal improvement in highlight handling and exposure latitude.)

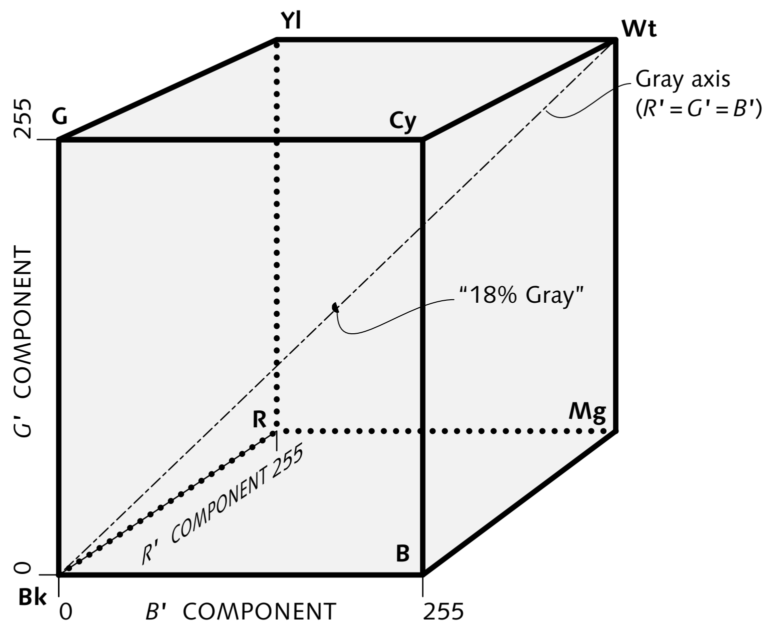
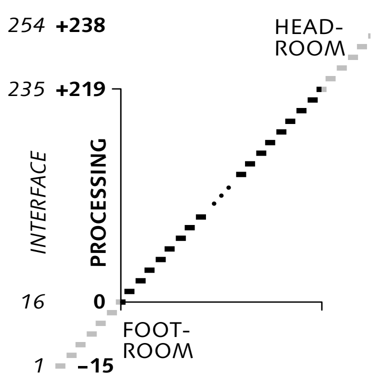



Figure C.2 **R′G′B′ cube** represents nonlinear (gamma corrected) *R′G′B′* typical of computer graphics. Though superficially similar to the *RGB* cube of Figure C.1, it is dramatically different in practice owing to its perceptual coding.




Charles Poynton, *Concerning "legal" and "valid" video signals*, www.poynton.com

Eight-bit Rec. 601 coding has an excursion of 219 codes from black to white. For no good technical reason, footroom and headroom are assigned asymmetrically: Footroom has 15 levels, but headroom has 19. An offset of +16 is added at an 8-bit interface. (Hardware engineers say that black is at code 16, and white is at code 235.) The sketch in the margin shows abstract levels in bold, and hardware levels in italics. Interface codes 0 and 255 are reserved for synchronization purposes, and are prohibited from appearing in video or ancillary data.

The so-called valid colors encompass the volume that is spanned when each *R′G′B′* component ranges from reference black to reference white. In Rec. 601, each component has 219 steps (risers) – that is, 220 levels. That gives 220×220×220, or 10648000 colors: About 64% of the total volume of codewords is valid.

Linear light *RGB* is the basis for color representation in film and CGI, but linear light coding is a poor match to human perception. Greatly improved results are obtained by using nonlinear *R′G′B′* coding that mimics the lightness sensitivity of vision. We can use another more subtle application of the properties of vision to code video signals: Vision has poor acuity to color detail, compared to its acuity for lightness. Providing that lightness detail is maintained, color detail can be discarded. Owing to the nature of the visual system, if subsampling is done correctly, it will not be noticed. Subsampling has two steps: First, a lightness component and two color components are formed. Then, detail is discarded from the two color components.

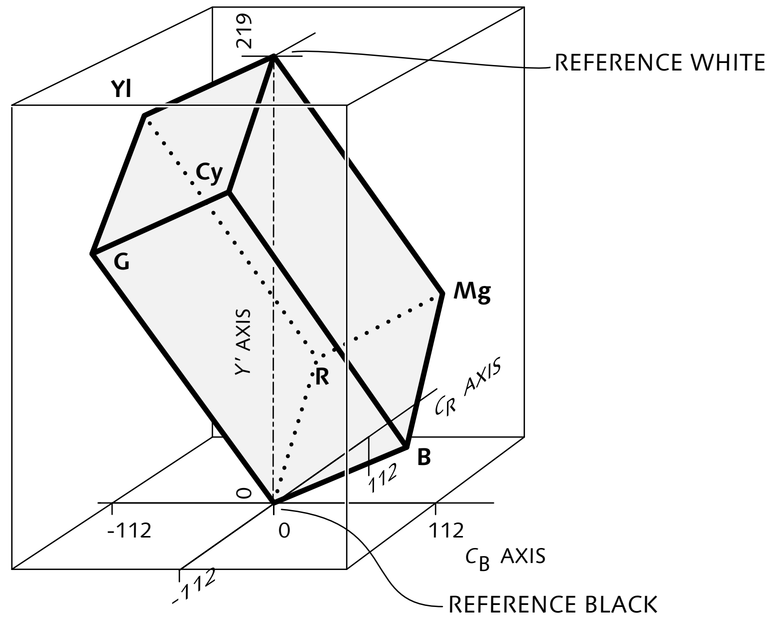



Figure C.3 **Y'$C_B$$C_R$ cube** is formed when gamma-corrected *R'G'B'* are transformed to luma and chroma signals, which are then then scaled. Only about ¼ of the available $Y'C_BC_R$ volume represents colors; the rest is wasted. This transform is performed before 4:2:2, 4:2:0, or 4:1:1 chroma subsampling.

## Y'$C_B$$C_R$ video

$$^{601}Y' = 0.299\,R' + 0.587\,G' + 0.114\,B'$$

To exploit the poor color acuity of vision, *luma* is formed as a properly-weighted sum of nonlinear *R'*, *G'*, and *B'*. It is standard to use the coefficients of Rec. 601. Two *color difference* – or *chroma* – components are then formed as *blue minus luma* and *red minus luma*, where blue, red, and luma incorporate gamma correction. (Luma, *B'–Y'*, and *R'–Y'* can be formed simultaneously from *R'*, *G'*, and *B'* through a 3×3 matrix multiplication.)

Charles Poynton, *YUV and luminance considered harmful: A plea for precise terminology in video*, www.poynton.com

Various scale factors, and various notations, are applied to the basic *B'–Y'* and *R'-Y'* color differences. The correct scaling and nomenclature for component digital systems is $Y'C_BC_R$ (not *YUV*). The correct term for the lightness component is *luma* (not *luminance*).

If each of the *Y'*, $C_B$, and $C_R$ components has 8 bits of precision, then obviously the entire $Y'C_BC_R$ cube has the same number of codewords as 8-bit *R'G'B'*. However, it is immediately obvious from the appearance of the transformed *R'G'B'* unit cube in Figure C.3 above that only a small fraction of the total volume of the $Y'C_BC_R$ coordinate space is occupied by colors! The number of colors accommodated is computed as the determinant of the transform matrix. In Rec. 601 $Y'C_BC_R$, only about ¼ of the Rec. 601 studio video *R'G'B'* codes are used.

$$\frac{\frac{1}{4}\cdot 220\cdot 225^2}{220^3} = \frac{2784375}{10648000} = 0.261$$

Of the 16.7 million colors available in studio *R'G'B'*, only about 2.75 million are available in $Y'C_BC_R$. If *R'G'B'* is transcoded to $Y'C_BC_R$,

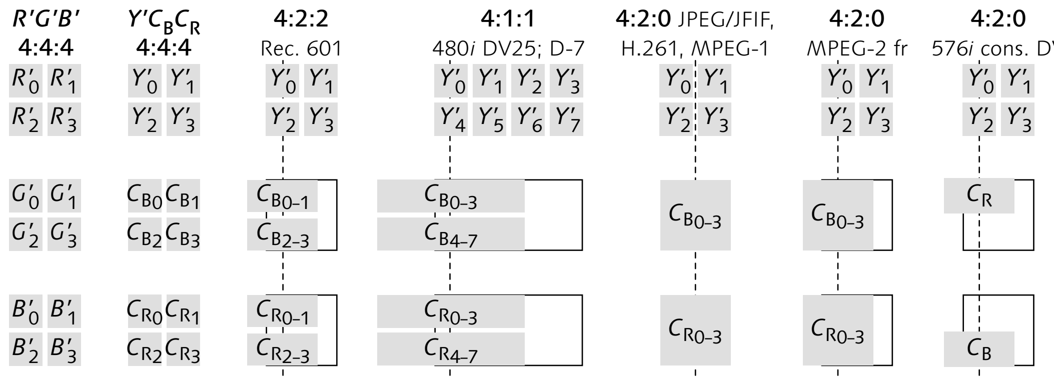



Figure C.4 **Chroma subsampling.** Providing full luma detail is maintained, vision's poor color acuity enables color detail to be reduced by subsampling. A 2×2 array of *R'G'B'* pixels is *matrixed* to a luma component *Y'* and color difference (*chroma*) components $C_B$ and $C_R$. $C_B$ and $C_R$ are then filtered (averaged). Here, $C_B$ and $C_R$ samples are drawn wider or taller than the luma samples to indicate their spatial extent. The horizontal offset of $C_B$ and $C_R$ is due to cositing. (In 4:2:0 in JPEG/JFIF, MPEG-1, and H.261, chroma samples are sited *interstitially*, not cosited.)

then transcoded back to *R'G'B'*, the resulting *R'G'B'* cannot have any more than 2.75 million colors!

Izraelevitz, David, and Joshua L. Koslov, "Code Utilization for Component-coded Digital Video," in *Tomorrow's Television, Proceedings of 16th Annual SMPTE Television Conference* (White Plains, New York: SMPTE, 1982), 22–30.

The color difference components are bipolar. Unscaled, they range from roughly –1 to +1. For analog engineers, the doubled excursion represents a 6 dB SNR penalty for the chroma components. Digital engineers should consider the sign to consume an extra bit in each of $C_B$ and $C_R$. This codeword utilization issue represents a serious limitation of 8-bit $Y'C_BC_R$ performance. It necessitates techniques such as Quantel's patented *dynamic rounding*®.

In addition to this obvious problem of codeword utilization, transforms between $Y'C_BC_R$ and *R'G'B'* must have carefully-chosen matrix coefficients. If the product of the encoding matrix and the decoding matrix is not very nearly an identity matrix, then roundoff errors will accumulate every time an image is transcoded. High-end manufacturers take great care in choosing these matrix coefficients; however, the entire problem is circumvented by operating in *R'G'B'*.

## Chroma subsampling

Once color difference components have been formed, they can be subsampled (filtered). The data compression that results from subsampling is the justification for using $Y'C_BC_R$ in the first place! To subsample by simply dropping samples leads to aliasing, and consequent poor image quality. It is necessary to perform some sort of averaging operation. The various subsampling schemes in use are sketched in Figure C.4 above.

Some systems implement 4:2:0 subsampling with minimum computation by simply averaging $C_B$ over a 2×2 block, and averaging $C_R$ over the same 2×2 block. Simple averaging causes subsampled chroma to take an effective position centered among a 2×2 block of luma

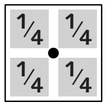



Figure C.6 **Interstitial 4:2:0 filter** for subsampling may be implemented using simple averaging. The rectangular outline indicates the subsampled $Y'C_BC_R$ block; the black dot suggests the effective siting of the computed chroma sample.

samples, what I call *interstitial* siting. Low-end decoders simply replicate the subsampled 4:2:0 $C_B$ and $C_R$ to obtain the missing chroma samples, prior to conversion back to *R′G′B′*. This technique, sketched in Figure C.6 in the margin, is used in JPEG/JFIF stillframes in computing, MPEG-1, and ITU-R Rec. H.261 videoconferencing.

Simple averaging causes subsampled chroma to take an effective position halfway between two luma samples, what I call *interstitial* siting. This approach is inconsistent with standards for studio video and MPEG-2, where $C_B$ and $C_R$ are *cosited* horizontally.

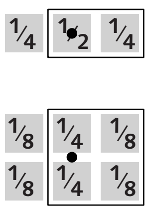



Figure C.7 **Cosited filters** for subsampling use weights that cause each computed chroma sample to be horizontally aligned with a luma sample.

Weights of [$\frac{1}{4}$, $\frac{1}{2}$, $\frac{1}{4}$] can be used to achieve horizontal cositing as required by Rec. 601, while still using simple computation, as sketched at the top of Figure C.7 in the margin. A [$\frac{1}{4}$, $\frac{1}{2}$, $\frac{1}{4}$] filter can be combined with [$\frac{1}{2}$, $\frac{1}{2}$] vertical averaging, so as to be extended to 4:2:0 used in MPEG-2, as sketched at the bottom of Figure C.7.

Simple averaging filters exhibit poor image quality. Providing the weights are carefully chosen, a filter combining a large number of samples – that is, a filter with a larger number of *taps* – will always perform better than a filter with a smaller number of taps. (This fact is not intuitive, because high frequency information is only apparent across a small scale.) High-end digital video and film equipment uses sophisticated subsampling filters, where the subsampled $C_B$ and $C_R$ of a 2×1 pair in 4:2:2, or 2×2 quad of 4:2:0, take contributions from many surrounding samples.

## Sample aspect ratio, “square sampling”

In computing, it is a *de facto* standard to have samples equally-spaced horizontally and vertically (“square sampling”). In conventional video, various sample aspect ratios are in use: Sample aspect ratios differ between 525/59.94 and 625/50, and neither has equally-spaced samples. In high-definition television (HDTV), thankfully, square sampling has been adopted.

In certain adaptations of $Y'C_BC_R$ for film, the nonsquare sample aspect ratio of conventional 625/50 video has been maintained. This forces a resampling operation when that imagery is imported into the CGI environment, and another resampling operation when it is exported. If resampling is done well, it is intrinsically expensive. If resampling is done poorly, or done often (in tandem), it introduces artifacts.

## R′G′B′ and $Y'C_BC_R$ characterization

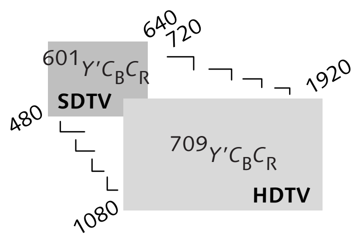
Charles Poynton, “The rehabilitation of *gamma*,” in *Human Vision and Electronic Imaging III*, Proc. SPIE/IS&T Conf. 3299, ed. B.E. Rogowitz and T.N. Pappas (Bellingham, Wash.: SPIE, 1998).

*R′G′B′* is completely characterized by four technical parameters: white point, primary chromaticities, transfer function, and coding range. (A fifth *rendering intent* parameter is implicit; see my SPIE/IS&T paper.)

White point, primary chromaticities, and transfer function are all standardized by Rec. 709. The parameters of Rec. 709 closely represent current practice in video and in computing. We have, in effect, reached worldwide consensus on *R′G′B′* coding. This is highly significant.

Coding range in computing has a *de facto* standard excursion, 0 to 255. Studio video accommodates footroom and headroom; its range is standardized from 16 to 235. (In ITU-R Rec. BT.1361, the coding range of Rec. 709 is extended to achieve a wider gamut.)

$$^{709}Y' = 0.2126\,R' + 0.7122\,G' + 0.0722\,B'$$




$Y'C_BC_R$ is characterized by all of the parameters of $R'G'B'$, plus a set of luma coefficients. The coefficients of Rec. 601 are ubiquitous in conventional 525/59.94 video, 625/50 video, and computing. But according to recently-adopted SMPTE and Advanced Television Systems Committee (ATSC) standards, HDTV will use a new, different set: the luma coefficients of Rec. 709. This introduces a huge problem: There will be one flavor of $Y'C_BC_R$ for small, standard-definition television (SDTV) pictures, and another for big (HDTV) pictures. $Y'C_BC_R$ data cannot be accurately exchanged between these flavors of coding without undergoing a mathematical transform of comparable complexity – and comparable susceptibility to artifacts – as resampling for the correction of pixel aspect ratio. (If the mathematical transform is not performed, then dramatic color errors result.)

## Practical suggestions

To maximize performance at the interface of computing and video, I recommend that you take these steps:

**Acquire R′G′B′ 4:4:4 images** wherever possible, instead of acquiring images already subjected to the $Y'C_BC_R$ transform and 4:2:2 subsampling. For realtime transfer, use the dual SDI link.

**Stay in R′G′B′** if your production situation permits. The first conversion to $Y'C_BC_R$ will cause an unrecoverable loss of 75% of the available $R'G'B'$ codewords, and the first subsampling to 4:2:2 will cause an unrecoverable loss of half the color detail.

**Avoid repeated conversions** back and forth between $R'G'B'$ and 4:2:2. Conversions after the first are liable to accumulate rounding errors, and are liable to accumulate filtering artifacts such as aliasing.

**Retain intermediates in R′G′B′ 4:4:4 format** where possible. Use DLT or Exabyte computer media, instead of videotape. Where intermediate or archival work must be recorded on video equipment, use 10-bit D-5 recording, instead of 8-bit D-1.

**Minimize resampling.** To the extent possible, avoid changing from one sample structure to another – for example, from square sampling to nonsquare, or from nonsquare to square.

**Establish and maintain accurate black levels.** Establish the correct black level for a scene or an element upon entry to the digital domain. When possible, perform this adjustment using video playback equipment. (Establishing and maintaining white level is not quite so important.)