**Cukurova University**

**Computer Engineering Department**

**Image Processing Course Project**

# COLOR SPACES

**Prepared by:**

METİN YORĞUN

YUSUF KUVVETLİ

**Advisor**

Assoc. Prof. Mustafa ORAL

25.11.2018

# Content

# 1. Color Definition

Color is the aspect of things that is caused by differing qualities of light being reflected or emitted by them.

To see color, you have to have light. When light shines on an object some colors bounce off the object and others are absorbed by it. Our eyes only see the colors that are bounced off or reflected.

The sun's rays contain all the colors of the rainbow mixed together. This mixture is known as white light. When white light strikes a white crayon or marker barrel, it appears white to us because it absorbs no color and reflects all color equally. A black crayon or marker cap absorbs all colors equally and reflects none, so it looks black to us. While artists consider black a color, scientists do not because black is the absence of all color.


All light rays contain color. Light is made of electromagnetic waves. These waves spread out from any light source, such as the sun. Light waves travel at tremendous speed (186,000 miles or 300,000 kilometers per second). Different colors have different wavelengths, which is the distance between corresponding parts of two of the waves. The longest wavelength of light that humans can see is red. The shortest is violet. Ultraviolet has an even shorter wavelength, but humans cannot see it. Some birds and bees can see ultraviolet light. Infrared has a longer wavelength than red light, and humans can not see this light but can feel the heat infrared generates.

# 2. Color Spaces Definition

## 2.1. Overview

A range of colors can be created by the primary colors of pigment and these colors then define a specific color space. **Color space**, also known as the color model (or color system), is an abstract mathematical model which simply describes the range of colors as tuples of numbers, typically as 3 or 4 values or color components (e.g. **RGB**). Basically speaking, color space is an elaboration of the coordinate system and sub-space. Each color in the system is represented by a single dot.

A color space is a useful method for users to understand the color capabilities of a particular digital device or file. It represents what a camera can see, a monitor can display or a printer can print, and etc. There are a variety of color spaces, such as RGB, CMY, HSV, HIS. We will talk about RGB color space in this article.

## 2.2. Color Spaces Type

### 2.2.1. Additive Color

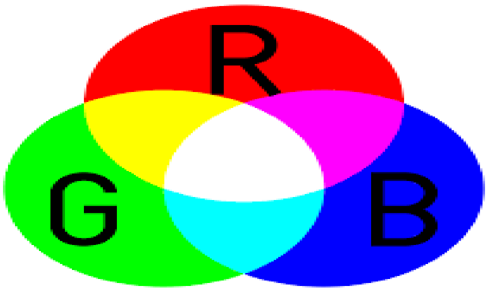
To form a color with RGB, three light beams (one red, one green, and one blue) must be superimposed (for example by emission from a black screen or by reflection from a white screen). Each of the three beams is called a *component* of that color, and each of them can have an arbitrary intensity, from fully off to fully on, in the mixture.

The RGB color model is *additive* in the sense that the three light beams are added together, and their light spectra add, wavelength for wavelength, to make the final color's spectrum.[1][2] This is essentially opposite to the subtractive color model that applies to paints, inks, dyes, and other substances whose color depends on *reflecting* the light under which we see them. Because of properties, these three colours create white, this is in stark contrast to physical colours, such as dyes which create black when mixed.

Zero intensity for each component gives the darkest color (no light, considered the *black*), and full intensity of each gives a white; the *quality* of this white depends on the nature of the primary light sources, but if they are properly balanced, the result is a neutral white matching the system's white point. When the intensities for all the components are the same, the result is a shade of gray, darker or lighter depending on the intensity. When the intensities are different, the result is a colorized hue, more or less saturated depending on the difference of the strongest and weakest of the intensities of the primary colors employed.

When one of the components has the strongest intensity, the color is a hue near this primary color (reddish, greenish or bluish), and when two components have the same strongest intensity, then the color is a hue of a secondary color (a shade of cyan, magenta or yellow). A secondary color is formed by the sum of two primary colors of equal intensity: cyan is green+blue, magenta is red+blue, and yellow is red+green. Every secondary color is the *complement* of one primary color; when a primary and its complementary secondary color are added together, the result is white: cyan complements red, magenta complements green, and yellow complements blue.

The RGB color model itself does not define what is meant by *red*, *green* and *blue* colorimetrically, and so the results of mixing them are not specified as absolute, but relative to the primary colors. When the exact chromaticities of the red, green, and blue primaries are defined, the color model then becomes an absolute color space, such as sRGB or Adobe RGB; see RGB color spaces for more details.

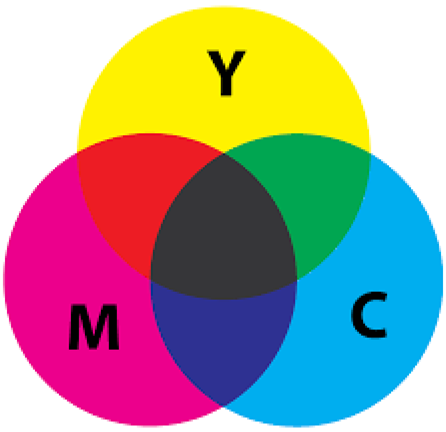



### 2.2.2. Subtractive Color

A **subtractive color** model explains the mixing of a limited set of dyes, inks, paint pigments or natural colorants to create a wider range of colors, each the result of partially or completely subtracting (that is, absorbing) some wavelengths of light and not others. The color that a surface displays depends on which parts of the visible spectrum are not absorbed and therefore remain visible.

Subtractive color systems start with light, presumably white light. Colored inks, paints, or filters between the watchers and the light source or reflective surface *subtract* wavelengths from the light, giving it color. If the incident light is other than white, our visual mechanisms are able to compensate well, but not perfectly, often giving a flawed impression of the "true" color of the surface.

Conversely, additive color systems start with darkness. Light sources of various wavelengths are added in various proportions to produce a range of colors. Usually, three primary colors are combined to stimulate humans' trichromatic color vision, sensed by the three types of cone cells in the eye, giving an apparently full range.



### 2.2.3. RGB Color Spaces

#### 2.2.3.1. RGB

The **RGB color model** is an additive color model in which red, green and blue light are added together in various ways to reproduce a broad array of colors. The name of the model comes from the initials of the three additive primary colors, red, green, and blue.

The main purpose of the RGB color model is for the sensing, representation and display of images in electronic systems, such as televisions and computers, though it has also been used in conventional photography. Before the electronic age, the RGB color model already had a solid theory behind it, based in human perception of colors.

RGB is a *device-dependent* color model: different devices detect or reproduce a given RGB value differently, since the color elements (such as phosphors or dyes) and their response to the individual R, G, and B levels vary from manufacturer to manufacturer, or even in the same device over time. Thus an RGB value does not define the same *color* across devices without some kind of color management.

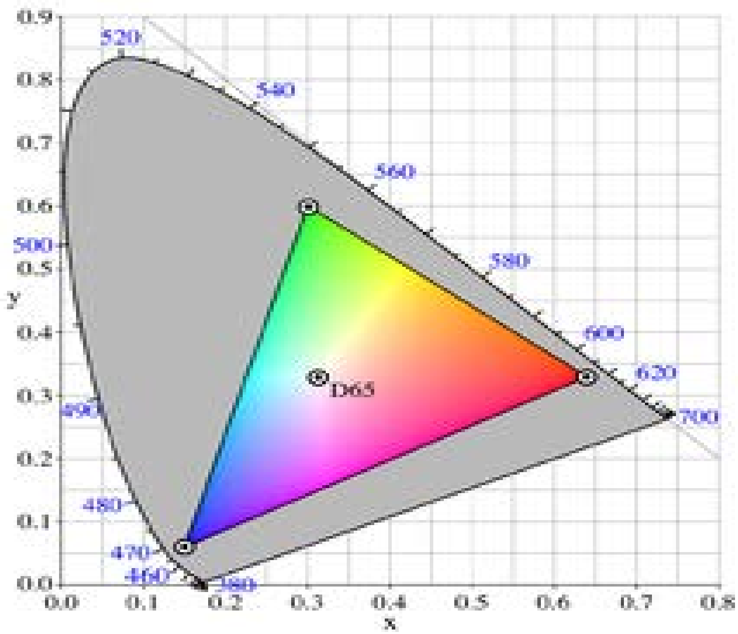
Typical RGB input devices are color TV and video cameras, image scanners, and digital cameras. Typical RGB output devices are TV sets of various technologies (CRT, LCD, plasma, OLED, quantum dots, etc.), computer and mobile phone displays, video projectors, multicolor LED displays and large screens such as JumboTron. Color printers, on the other hand are not RGB devices, but subtractive color devices (typically CMYK color model).

This article discusses concepts common to all the different color spaces that use the RGB color model, which are used in one implementation or another in color image-producing technology.

**Physical principles for the choice of Red,Green and Blue**

The choice of primary colors is related to the physiology of the human eye; good primaries are stimuli that maximize the difference between the responses of the cone cells of the human retina to light of different wavelengths, and that thereby make a large color triangle.[3]

The normal three kinds of light-sensitive photoreceptor cells in the human eye (cone cells) respond most to yellow (long wavelength or L), green (medium or M), and violet (short or S) light (peak wavelengths near 570 nm, 540 nm and 440 nm, respectively[3]). The difference in the signals received from the three kinds allows the brain to differentiate a wide gamut of different colors, while being most sensitive (overall) to yellowish-green light and to differences between hues in the green-to-orange region




As an example, suppose that light in the orange range of wavelengths (approximately 577 nm to 597 nm) enters the eye and strikes the retina. Light of these wavelengths would activate both the medium and long wavelength cones of the retina, but not equally—the long-wavelength cells will respond more. The difference in the response can be detected by the brain, and this difference is the basis of our perception of orange. Thus, the orange appearance of an object results from light from the object entering our eye and stimulating the different cones simultaneously but to different degrees.

Use of the three primary colors is not sufficient to reproduce *all* colors; only colors within the color triangle defined by the chromaticities of the primaries can be reproduced by additive mixing of non-negative amounts of those colors of light

#### 2.2.3.2. SRGB

**sRGB** (**standard Red Green Blue**) is an RGB color space that HP and Microsoft created cooperatively in 1996 to use on monitors, printers, and the Internet. It was subsequently standardized by the IEC as IEC 61966-2-1:1999.[1] It is often the "default" color space for images that contain no color space information, especially if the images' pixels are stored in 8-bit integers per color channel.

sRGB uses the ITU-R BT.709 primaries, the same as in studio monitors and HDTV,[2] a transfer function (gamma curve) typical of CRTs, and a viewing environment designed to match typical home and office viewing conditions. This specification allowed sRGB to be directly displayed on typical CRT monitors of the time, which greatly aided its acceptance.

**The sRGB Gamut**

sRGB defines the chromaticities of the red, green, and blue primaries, the colors where one of the three channels is nonzero and the other two are zero. The gamut of chromaticities that can be represented in sRGB is the color triangle defined by these primaries. As with any RGB color space, for non-negative values of R, G, and B it is not possible to represent colors outside this triangle, which is well inside the range of colors visible to a human with normal trichromatic vision.

sRGB is sometimes avoided by high-end print publishing professionals because its color gamut is not big enough, especially in the blue-green colors, to include all the colors that can be reproduced in CMYK printing.

**The sRGB Transfer Function(gamma)**

sRGB also defines a nonlinear transformation between the intensity of these primaries and the actual number stored. The curve is similar to the gamma response of a CRT display. This nonlinear conversion means that sRGB is a reasonably efficient use of the values in an integer-based image file to display human-discernible light levels.

Unlike most other RGB color spaces, the sRGB gamma cannot be expressed as a single numerical value. The overall gamma is approximately 2.2, consisting of a linear (gamma 1.0) section near black, and a non-linear section elsewhere involving a 2.4 exponent and a gamma (slope of log output versus log input) changing from 1.0 through about 2.3. The purpose of the linear section is so the curve does not have an infinite slope at zero, which could cause numerical problems.

**Usage**

Due to the standardization of sRGB on the Internet, on computers, and on printers, many low- to medium-end consumer digital cameras and scanners use sRGB as the default (or only available) working color space. As the sRGB gamut meets or exceeds the gamut of a low-end inkjet printer, an sRGB image is often regarded as satisfactory for home use. However, consumer-level CCDs are typically uncalibrated, meaning that even though the image is being labeled as sRGB, one can't conclude that the image is color-accurate sRGB.

If the color space of an image is unknown and it is an 8- to 16-bit image format, assuming it is in the sRGB color space is a safe choice. This allows a program to identify a color space for all images, which may be much easier and more reliable than trying to track the "unknown" color space. An ICC profile may be used; the ICC distributes three such profiles:[8] two profiles conforming to version 4 of the ICC specification, which they recommend, and one profile conforming to version 2, which is still commonly used.

Images intended for professional printing via a fully color-managed workflow, e.g. prepress output, sometimes use another color space such as Adobe RGB (1998), which accommodates a wider gamut. Such images used on the Internet may be converted to sRGB using color management tools that are usually included with software that works in these other color spaces.

The two dominant programming interfaces for 3D graphics, OpenGL and Direct3D, have both incorporated support for the sRGB gamma curve. OpenGL supports textureswith sRGB gamma encoded color components (first introduced with EXT_texture_sRGB extension, added to the core in OpenGL 2.1) and rendering into sRGB gamma encoded framebuffers (first introduced with EXT_framebuffer_sRGB extension, added to the core in OpenGL 3.0). Direct3D supports sRGB gamma textures and rendering into sRGB gamma surfaces starting with DirectX 9. Correct mipmapping and interpolation of sRGB gamma textures has direct hardware support in texturing units of most modern GPUs (for example nVidia GeForce 8 performs conversion from 8-bit texture to linear values before interpolating those values), and does not have any performance penalty.[9]

#### 2.2.3.3. Adobe RGB

The **Adobe RGB (1998) color space** is an RGB color space developed by Adobe Systems, Inc. in 1998. It was designed to encompass most of the colors achievable on CMYK color printers, but by using RGB primary colorson a device such as a computer display. The Adobe RGB (1998) color space encompasses roughly 50% of the visible colors specified by the CIELAB color space – improving upon the gamut of the sRGB color space, primarily in cyan-green hues.

**History**

Beginning in 1997, Adobe Systems was looking into creating ICC profiles that its consumers could use in conjunction with Photoshop's new color management features. Since not many applications at the time had any ICC color management, most operating systems did not ship with useful profiles.

Lead developer of Photoshop, Thomas Knoll decided to build an ICC profile around specifications he found in the documentation for the SMPTE 240M standard, the precursor to Rec. 709. SMPTE 240M's gamut was wider than that of the sRGB color space, but not by much. However, with the release of Photoshop 5.0 nearing, Adobe made the decision to include the profile within the software.

Although users loved the wider range of reproducible colors, those familiar with the SMPTE 240M specifications contacted Adobe, informing the company that it had copied the values that described idealized primaries, not actual standard ones. The real values were much closer to sRGB's, which avid Photoshop consumers did not enjoy as a working environment. To make matters worse, an engineer had made an error when copying the red primary chromaticity coordinates, resulting in an even more inaccurate representation of the SMPTE standard.

Adobe tried numerous tactics to correct the profile, such as correcting the red primary and changing the white point to match that of the CIE Standard Illuminant D50, yet all of the adjustments made CMYK conversion worse than before. In the end, Adobe decided to keep the "incorrect" profile, but changed the name to *Adobe RGB (1998)* in order to avoid a trademark search or infringement.

**Properties**

In Adobe RGB (1998), colors are specified as [*R*,*G*,*B*] triplets, where each of the *R*, *G*, and *B* components have values ranging between 0 and 1. When displayed on a monitor, the exact chromaticities of the reference white point [1,1,1], the reference black point [0,0,0], and the primaries ([1,0,0], [0,1,0], and [0,0,1]) are specified. To meet the color appearance requirements of the color space, the luminance of the monitor must be 160.00 cd/m$^2$ at the white point, and 0.5557 cd/m$^2$ at the black point, which implies a contrast ratio of 287.9. Moreover, the black point shall have the same chromaticity as the white point, yet with a luminance equal to 0.34731% of the white point luminance.[2] The ambient illumination level at the monitor faceplate when the monitor is turned off must be 32 lx.

As with sRGB, the *RGB* component values in Adobe RGB (1998) are not proportional to the luminances. Rather, a gamma of 2.2 is assumed, without the linear segment near zero that is present in sRGB. The precise gamma value is 563/256, or 2.19921875. In coverage of the CIE 1931 color space the Adobe RGB (1998) color space covers 52.1%.

**Comparison to sRGB**

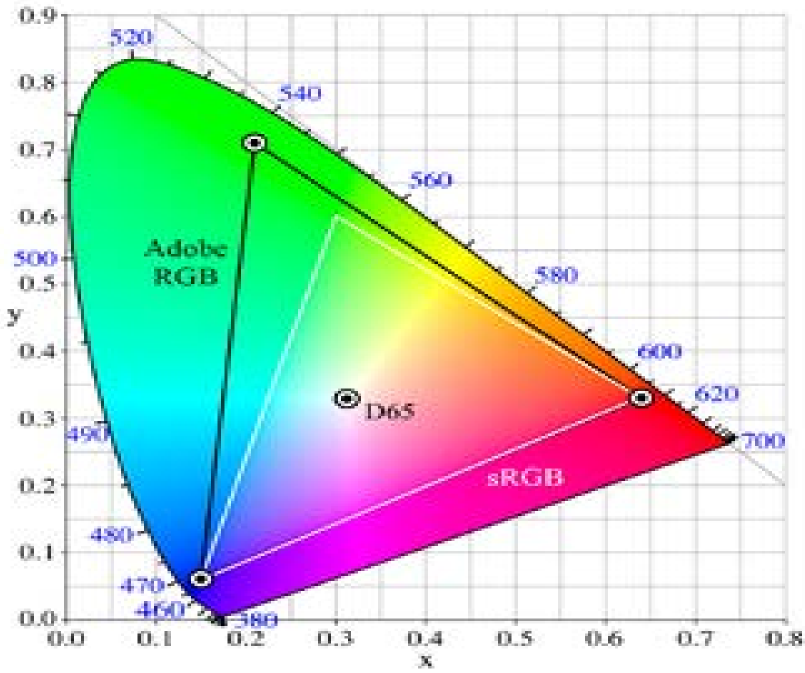
sRGB is an RGB color space proposed by HP and Microsoft in 1996 to approximate the color gamut of the most common computer display devices. Since sRGB serves as a "best guess" metric for how another person's monitor produces color, it has become the standard color space for displaying images on the Internet. sRGB's color gamut encompasses just 35% of the visible colors specified by CIE, whereas Adobe RGB (1998) encompasses slightly more than 50% of all visible colors. Adobe RGB (1998) extends into richer cyans and greens than does sRGB – for all levels of luminance. The two gamuts are often compared in mid-tone values (~50% luminance), but clear differences are evident in shadows (~25% luminance) and highlights (~75% luminance) as well. In fact, Adobe RGB (1998) expands its advantages to areas of intense orange, yellow, and magenta regions.[4]

Although there is a significant difference between gamut ranges in the CIE *xy* chromaticity diagram, if the coordinates were to be transformed to fit on the CIE *u′v′* chromaticity diagram, which illustrates the eye's perceived variance in hue more closely, the difference in the green region is far less exaggerated. Also, although Adobe RGB (1998) can *theoretically* represent a wider gamut of colors, the color space requires special software and a complex workflow in order to utilize its full range. Otherwise, the produced colors would be squeezed into a smaller range (making them appear duller) in order to match sRGB's more widely used gamut.

**Bit depth distribution**

Although the Adobe RGB (1998) working space clearly provides more colors to utilize, another factor to consider when choosing between color spaces is how each space influences the distribution of the image's bit depth. Color spaces with larger gamuts "stretch" the bits over a broader region of colors, whereas smaller gamuts concentrate these bits within a narrow region.

A similar, yet not as dramatic concentration of bit depth occurs with Adobe RGB (1998) versus sRGB, except in three dimensions rather than one. The Adobe RGB (1998) color space occupies roughly 40% more volume than the sRGB color space, which concludes that one would only be exploiting 70% of the available bit depth if the colors in Adobe RGB (1998) are unnecessary.[4] On the contrary, one may have plenty of "spare" bits if using a 16-bit image, thus negating any reduction due to the choice of working space.




#### 2.2.3.4. Adobe RGB wide gamut

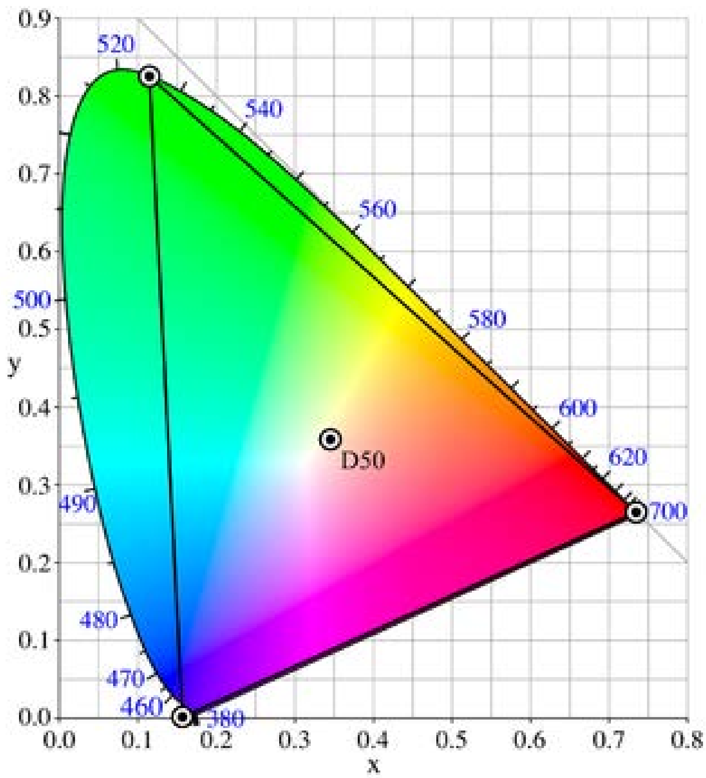
The **wide-gamut RGB color space** (or **Adobe Wide Gamut RGB**) is an RGB color space developed by Adobe Systems, that offers a large gamut by using pure spectral primary colors.[1] It is able to store a wider range of color values than sRGB or Adobe RGB color spaces. As a comparison, the wide-gamut RGB color space encompasses 77.6% of the visible colors specified by the CIELAB color space, while the standard Adobe RGB color space covers just 52.1%[2] and sRGB covers only 35.9%.[3]

When working in color spaces with such a large gamut, it is recommended to work in 16-bit per channel color depth to avoid posterization effects. This will occur more frequently in 8-bit per channel modes as the gradient steps are much larger.[4]

As with sRGB, the color component v alues in wide-gamut RGB are not proportional to the luminances. Similar to Adobe RGB, a gamma of 2.2 is assumed, without the linear segment near zero that is present in sRGB. The precise gamma value is 563/256, or 2.19921875.

The white point corresponds to D50. The chromaticities of the primary colors and the white point are as follows:

| Color | CIE x | CIE y | Wavelength |
|---|---|---|---|
| Red | 0.7347 | 0.2653 | 700 nm |
| Green | 0.1152 | 0.8264 | 525 nm |
| Blue | 0.1566 | 0.0177 | 450 nm |
| White point | 0.3457 | 0.3585 | |




**sRGB vs Wide Gamut RGB**

Wide Gamut allows for more colors at the start of your editing if you are going to adjust things quite a bit and want to minimize posterization and color shifting. This Wide-Gamut color-profile likely got set when you saved out of DPP so check those settings. Personally I use the even wider ProPhotoRGB when I convert from RAW, and do all my adjustments, then convert to sRGB just before I save as a JPG.

Many browsers assume sRGB so viewing a JPG as sRGB that is not sRGB will cause the colors to look different than they should, where the wider the gamut of the color space, the duller the colors will look if the browser guesses wrong.

### 2.2.4. CMYK Color Spaces

The **CMYK color model** (**process color**, **four color**) is a subtractive color model, used in color printing, and is also used to describe the printing process itself. **CMYK** refers to the four inks used in some color printing: **c**yan, **m**agenta, **y**ellow, and blac**k**.

The CMYK model works by partially or entirely masking colors on a lighter, usually white, background. The ink reduces the light that would otherwise be reflected. Such a model is called *subtractive* because inks "subtract" the colors red, green and blue from white light. White light minus red leaves cyan, white light minus green leaves magenta, and white light minus blue leaves yellow.

In additive color models, such as RGB, white is the "additive" combination of all primary colored lights, while black is the absence of light. In the CMYK model, it is the opposite: white is the natural color of the paper or other background, while black results from a full combination of colored inks. To save cost on ink, and to produce deeper black tones, unsaturated and dark colors are produced by using black ink instead of the combination of cyan, magenta, and yellow.

**Halftoning**

With CMYK printing, *halftoning* (also called *screening*) allows for less than full saturation of the primary colors; tiny dots of each primary color are printed in a pattern small enough that humans perceive a solid color. Magenta printed with a 20% halftone, for example, produces a pink color, because the eye perceives the tiny magenta dots on the large white paper as lighter and less saturated than the color of pure magenta ink.[*citation needed*]

Without halftoning, the three primary process colors could be printed only as solid blocks of color, and therefore could produce only seven colors: the three primaries themselves, plus three secondary colors produced by layering two of the primaries: cyan and yellow produce green, cyan and magenta produce blue, yellow and magenta produce red (these subtractive secondary colors correspond roughly to the additive primary colors), plus layering all three of them resulting in black. With halftoning, a full continuous range of colors can be produced

To improve print quality and reduce moiré patterns, the screen for each color is set at a different angle. While the angles depend on how many colors are used and the preference of the press operator, typical CMYK process printing uses any of the following screen angles:

| | | | | |
|---|---|---|---|---|
| **C** | 15° | 15° | 105° | 165° |
| **M** | 75° | 45° | 75° | 45° |
| **Y** | 0° | 0° | 90° | 90° |
| **K** | 45° | 75° | 15° | 105° |

The "black" generated by mixing commercially practical cyan, magenta, and yellow inks is unsatisfactory, so four-color printing uses black ink in addition to the subtractive primaries. Common reasons for using black ink include:

- In traditional preparation of color separations, a red keyline on the black line art marked the outline of solid or tint color areas. In some cases a black keyline was used when it served as both a color indicator and an outline to be printed in black. Because usually the black plate contained the keyline, the K in CMYK represents the keyline or black plate, also sometimes called the key plate.
- Text is typically printed in black and includes fine detail (such as serifs), so to reproduce text or other finely detailed outlines, without slight blurring, using three inks would require impractically accurate registration
- A combination of 100% cyan, magenta, and yellow inks soaks the paper with ink, making it slower to dry, causing bleeding, or (especially on cheap paper such as newsprint) weakening the paper so much that it tears.
- Although a combination of 100% cyan, magenta, and yellow inks should, in theory, completely absorb the entire visible spectrum of light and produce a perfect black, practical inks fall short of their ideal characteristics and the result is actually a dark muddy color that does not quite appear black. Adding black ink absorbs more light and yields much better blacks.
- Using black ink is less expensive than using the corresponding amounts of colored inks.

When a very dark area is desirable, a colored or gray CMY "bedding" is applied first, then a full black layer is applied on top, making a rich, deep black; this is called *rich black.* A black made with just CMY inks is sometimes called a *composite black.*

The amount of black to use to replace amounts of the other ink is variable, and the choice depends on the technology, paper and ink in use. Processes called under color removal, under color addition, and gray complement replacement are used to decide on the final mix; different CMYK recipes will be used depending on the printing task.[

### 2.2.5. Luma+ Chrominance Color Spaces

**Luminance (Y)**

Black and white television came first. A black and white image only contains white, black and shades of grey. All these values could (and still can) be obtained merely by varying brightness.

Pure unadulterated sunlight is white (or at least white enough for our purposes). When light isn't reflected (a black body), what we get is black. By varying the intensity of light, we get various shades of grey. Remember, grey is a perception of the human brain.

In scientific terms, the brightness of light is measured in terms of Luminance. The word 'Luminance' is directly defined by CIE in relation to human vision. It is represented by the letter Y as far as video is concerned.

**Luma (Y')**

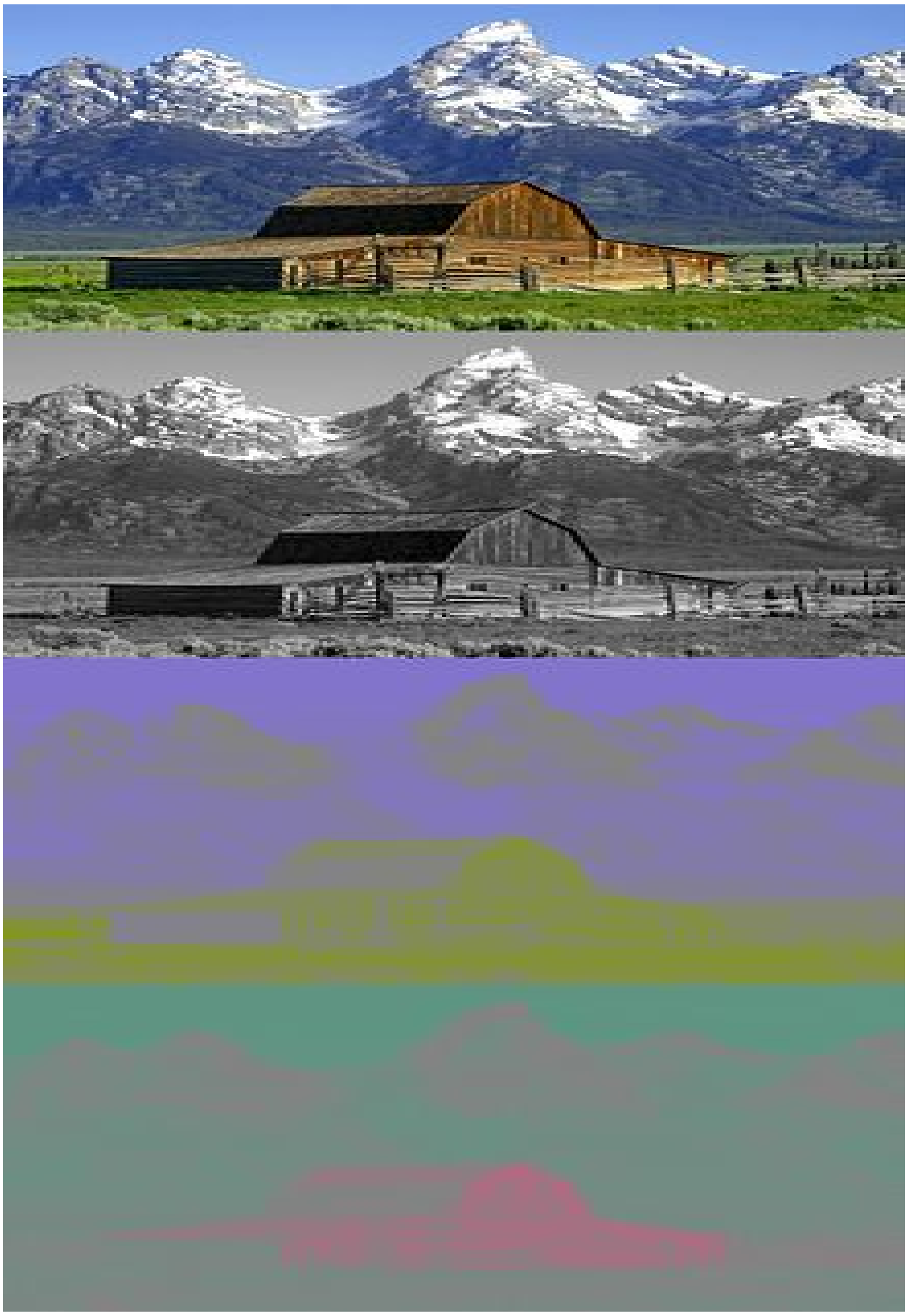
As we have seen in the chapter on Gamma, brightness is 'skewed' for display, a process we call gamma compression. Instead of calling it Twisted Luminance, they decided to call video-encoded luminance 'Luma', represented by Y'.

Whenever we speak of the luminance values with respect to video, we are speaking of Luma (Y'). When a video signal is coming out of an SDI port as 4:2:2 or whatever, the correct notation for it is Y'CbCr. Notice the Y with an apostrophe? The Luminance is already 'gammafied', or has been applied with an encoding gamma.

From this chapter on RGB we know that if Y is the child of RGB, Y' comes from R'G'B'.

Chrominance is the color information in a signal. When color television arrived, instead of starting from scratch they found a way to tack it on to Luma, as we have seen in the difference between CIE XYZ and xyY. From it, we know the two representations of color are named Cb and Cr.

It doesn't mean green is neglected, mind you. From RGB you get XYZ, and from XYZ you derive xyY, which leads to Y'CbCr. This is what it looks like:



The top-most is the full RGB image. If it had to be split into Y’CbCr, Y’ would be the second, Cb the third and Cr the fourth. Don’t assume just because it’s a ‘b’ in Cb and an ‘r’ in Cr it means blue or red – just as if you don’t expect a man called Mr. Red to be red in color.

## Chroma sampling and Sub Sampling

The name of the game is ‘reducing bandwidth’. Luminance becomes Luma, the bit depth is reduced to 8, the color spaces (Rec. 709, PAL, NTSC) are reduced to half or less than half the range of the human eye, data is compressed using compression algorithms, and so on. Can we compress it even further?

Sure we could. Similar to how rods and cones work in the brain, they realized they could split up colors into Luminance and Chrominance – that’s right, our eye exhibits both RGB and Y’CbCr behavior. That’s why it works!

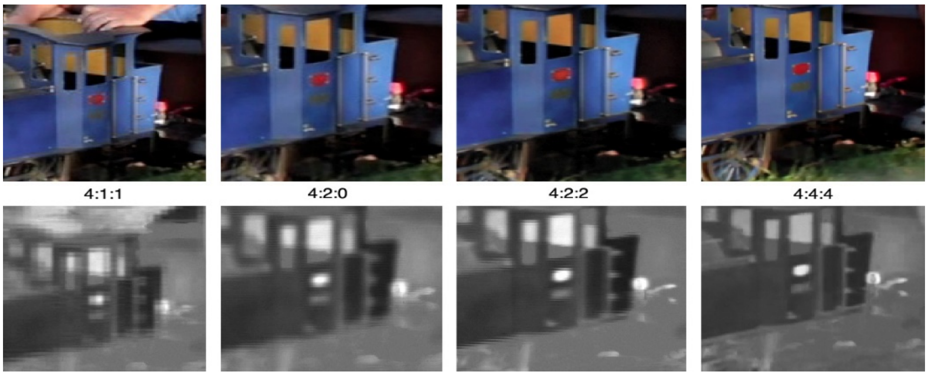
By a convoluted method of trial and error, and lots of testing, they discovered that they could actually throw out half or even three fourths of the color information and most people wouldn’t know it. People consume artificial flavoring every day, too.

It doesn’t work in the way data is compressed, and to really understand how it works, you’ll need to learn what sampling is. When only chrominance information (color) is sampled from analog data (or digital data), we call it Chroma Sampling.

When it is not sampled fully (like potato chips that don’t fill up the whole bag) the process is called Chroma sub-sampling.

It is a type of compression that can be applied at the time of recording in camera, or later in processing. Sub-sampling is what happens when you don’t sample 1:1, which means you throw out data. That’s the objective, to reduce bandwidth.

The most widely used model today is the Y’CbCr model. Take a look at this image:




**How subsampling is works?**

Because the human visual system is less sensitive to the position and motion of color than luminance,[2] bandwidth can be optimized by storing more luminance detail than color detail. At normal viewing distances, there is no perceptible loss incurred by sampling the color detail at a lower rate[*vague*]. In video systems, this is achieved through the use of color difference components. The signal is divided into a luma (Y') component and two color difference components (chroma).

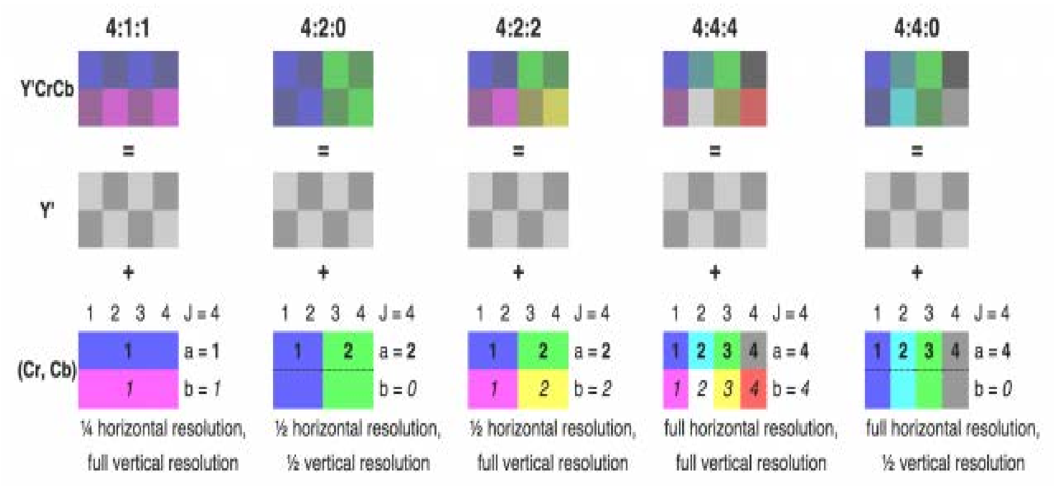
In human vision there are three channels for color detection, and for many color systems, three "channels" is sufficient for representing most colors. For example: red, green, blue or magenta, yellow, cyan. But there are other ways to represent the color. In many video systems, the three channels are luminance and two chroma channels. In video, the luma and chroma components are formed as a weighted sum of *gamma-corrected* (tristimulus) R'G'B' components instead of *linear* (tristimulus) RGB components. As a result, luma must be distinguished from luminance. That there is some "bleeding" of luminance and color information between the luma and chroma components in video, the error being greatest for highly saturated colors and noticeable in between the magenta and green bars of a color-bar test pattern (that has chroma subsampling applied), should not be attributed to this engineering approximation being used. Indeed, similar bleeding can occur also with gamma = 1, whence the reversing of the order of operations between gamma correction and forming the weighted sum can make no difference. The chroma can influence the luma specifically at the pixels where the subsampling put no chroma. Interpolation may then put chroma values there which are incompatible with the luma value there, and further post-processing of that Y'CbCr into R'G'B' for that pixel is what ultimately produces false luminance upon display.

**Sampling systems and ratios**

The subsampling scheme is commonly expressed as a three part ratio *J*:*a*:*b* (e.g. 4:2:2) or four parts if alpha channel is present (e.g. 4:2:2:4), that describe the number of luminance and chrominance samples in a conceptual region that is *J* pixels wide, and 2 pixels high. The parts are (in their respective order):

- *J*: horizontal sampling reference (width of the conceptual region). Usually, 4.
- *a*: number of chrominance samples (Cr, Cb) in the first row of *J* pixels.
- *b*: number of changes of chrominance samples (Cr, Cb) between first and second row of *J* pixels.
- *Alpha*: horizontal factor (relative to first digit). May be omitted if alpha component is not present, and is equal to *J* when present.

This notation is not valid for all combinations and has exceptions, e.g. 4:1:0 (where the height of the region is not 2 pixels but 4 pixels, so if 8 bits/component are used the media would be 9 bits/pixel) and 4:2:1.




**Types of Sampling and Subsampling**

4:4:4

Each of the three Y'CbCr components have the same sample rate, thus there is no chroma subsampling. This scheme is sometimes used in high-end film scanners and cinematic post production.

Note that "4:4:4" may instead be referring to R'G'B' color space, which implicitly also does not have any chroma subsampling. Formats such as HDCAM SR can record 4:4:4 R'G'B' over dual-link HD-SDI.

4:2:2

The two chroma components are sampled at half the sample rate of luma: the horizontal chroma resolution is halved. This reduces the bandwidth of an uncompressed video signal by one-third with little to no visual difference.[*citation needed*]

4:2:1

This sampling mode is not expressible in J:a:b notation. '4:2:1' is an obsolete term from a previous notational scheme, and very few software or hardware codecs use it. **Cb**horizontal resolution is half that of **Cr** (and a quarter of the horizontal resolution of **Y**).

4:1:1

In 4:1:1 chroma subsampling, the horizontal color resolution is quartered, and the bandwidth is halved compared to no chroma subsampling. Initially, 4:1:1 chroma subsampling of the DV format was not considered to be broadcast quality and was only acceptable for low-end and consumer applications.[3][4] However, DV-based formats (some of which use 4:1:1 chroma subsampling) have been used professionally in electronic news gathering and in playout servers. DV has also been sporadically used in feature films and in digital cinematography.

In the NTSC system, if the luma is sampled at 13.5 MHz, then this means that the **Cr** and **Cb** signals will each be sampled at 3.375 MHz, which corresponds to a maximum

Nyquist bandwidth of 1.6875 MHz, whereas traditional "high-end broadcast analog NTSC encoder" would have a Nyquist bandwidth of 1.5 MHz and 0.5 MHz for the I/Q channels. However, in most equipment, especially cheap TV sets and VHS/Betamax VCRs the chroma channels have only the 0.5 MHz bandwidth for both **Cr** and **Cb** (or equivalently for I/Q). Thus the DV system actually provides a superior color bandwidth compared to the best composite analog specifications for NTSC, despite having only 1/4 of the chroma bandwidth of a "full" digital signal.

4:2:0

In 4:2:0, the horizontal sampling is doubled compared to 4:1:1, but as the **Cb** and **Cr** channels are only sampled on each alternate line in this scheme, the vertical resolution is halved. The data rate is thus the same. This fits reasonably well with the PAL color encoding system since this has only half the vertical chrominance resolution of NTSC. It would also fit extremely well with the SECAM color encoding system since like that format, 4:2:0 only stores and transmits one color channel per line (the other channel being recovered from the previous line). However, little equipment has actually been produced that outputs a SECAM analogue video signal. In general SECAM territories either have to use a PAL capable display or a transcoder to convert the PAL signal to SECAM for display.

Cb and Cr are each subsampled at a factor of 2 both horizontally and vertically.

There are three variants of 4:2:0 schemes, having different horizontal and vertical siting.[7]

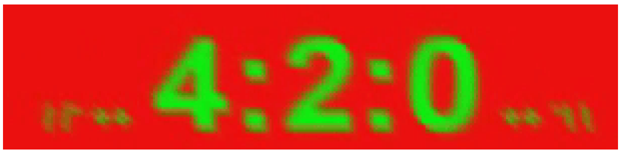
- In MPEG-2, Cb and Cr are cosited horizontally. Cb and Cr are sited between pixels in the vertical direction (sited interstitially).
- In JPEG/JFIF, H.261, and MPEG-1, Cb and Cr are sited interstitially, halfway between alternate luma samples.
- In 4:2:0 DV, Cb and Cr are co-sited in the horizontal direction. In the vertical direction, they are co-sited on alternating lines.

Most digital video formats corresponding to PAL use 4:2:0 chroma subsampling, with the exception of DVCPRO25, which uses 4:1:1 chroma subsampling. Both the 4:1:1 and 4:2:0 schemes halve the bandwidth compared to no chroma subsampling.

With interlaced material, 4:2:0 chroma subsampling can result in motion artifacts if it is implemented the same way as for progressive material. The luma samples are derived from separate time intervals while the chroma samples would be derived from both time intervals. It is this difference that can result in motion artifacts. The MPEG-2 standard allows for an alternate interlaced sampling scheme where 4:2:0 is applied to each field (not both fields at once). This solves the problem of motion artifacts, reduces the vertical chroma resolution by half, and can introduce comb-like artifacts in the image.


Original. *This image shows a single field. The moving text has some motion blur applied to it.

4:2:0 **progressive** sampling applied to moving *interlaced* material. Note that the chroma leads and trails the moving text. *This image shows a single field.




4:2:0 **interlaced** sampling applied to moving *interlaced* material. *This image shows a single field.

In the 4:2:0 interlaced scheme however, vertical resolution of the chroma is roughly halved since the chroma samples effectively describe an area 2 samples wide by 4 samples tall instead of 2X2. As well, the spatial displacement between both fields can result in the appearance of comb-like chroma artifacts.


Original still image.


4:2:0 **progressive** sampling applied to a still image. Both fields are shown.


4:2:0 **interlaced** sampling applied to a still image. Both fields are shown.

If the interlaced material is to be de-interlaced, the comb-like chroma artifacts (from 4:2:0 interlaced sampling) can be removed by blurring the chroma vertically.[8]

4:1:0

This ratio is possible, and some codecs support it, but it is not widely used. This ratio uses half of the vertical and one-fourth the horizontal color resolutions, with only one-eighth of the bandwidth of the maximum color resolutions used. Uncompressed video in this format with 8-bit quantization uses 10 bytes for every macropixel (which is 4 x 2 pixels). It has the equivalent chrominance bandwidth of a PAL I signal decoded with a delay line decoder, and still very much superior to NTSC.

- Some video codecs may operate at 4:1:0.5 or 4:1:0.25 as an option, so as to allow similar to VHS quality.

3:1:1

Used by Sony in their HDCAM High Definition recorders (not HDCAM SR). In the horizontal dimension, luma is sampled horizontally at three quarters of the full HD sampling rate – 1440 samples per row instead of 1920. Chroma is sampled at 480 samples per row, a third of the luma sampling rate.

In the vertical dimension, both

#### 2.2.5.1. YIQ

**YIQ** is the color space used by the NTSC color TV system, employed mainly in North and Central America, and Japan. *I* stands for *in-phase*, while *Q* stands for *quadrature*, referring to the components used in quadrature amplitude modulation. Some forms of NTSC now use the YUV color space, which is also used by other systems such as PAL.

The Y component represents the luma information, and is the only component used by black-and-white television receivers. I and Q represent the chrominance information. In YUV, the U and V components can be thought of as X and Y coordinates within the color space. I and Q can be thought of as a second pair of axes on the same graph, rotated 33°; therefore IQ and UV represent different coordinate systems on the same plane.

The YIQ system is intended to take advantage of human color-response characteristics. The eye is more sensitive to changes in the orange-blue (I) range than in the purple-green range (Q)—therefore less bandwidth is required for Q than for I. Broadcast NTSC limits I to 1.3 MHz and Q to 0.4 MHz. I and Q are frequency interleaved into the 4 MHz Y signal, which keeps the bandwidth of the overall signal down to 4.2 MHz. In YUV systems, since U and V both contain information in the orange-blue range, both components must be given the same amount of bandwidth as I to achieve similar color fidelity.

Very few television sets perform true I and Q decoding, due to the high costs of such an implementation. Compared to the cheaper R-Y and B-Y decoding which requires only one filter, I and Q each requires a different filter to satisfy the bandwidth differences between I and Q. These bandwidth differences also require that the 'I' filter include a time delay to match the longer delay of the 'Q' filter. The Rockwell Modular Digital Radio (MDR) was one I and Q decoding set, which in 1997 could operate in frame-at-a-time mode with a PC or in realtime with the Fast IQ Processor (FIQP). Some RCA "Colortrak" home TV receivers made circa 1985 not only used I/Q decoding, but also advertised its benefits along with its comb filtering benefits as full "100 percent processing" to deliver more of the original color picture content. Earlier, more than one brand of color TV (RCA, Arvin) used I/Q decoding in the 1954 or 1955 model year on models utilizing screens about 13 inches (measured diagonally). The original Advent projection television used I/Q decoding. Around 1990, at least one manufacturer (Ikegami) of professional studio picture monitors advertised I/Q decoding.

The YIQ representation is sometimes employed in color image processing transformations. For example, applying a histogram equalization directly to the channels in an RGB image would alter the color balance of the image. Instead, the histogram equalization is applied to the Y channel of the YIQ or YUV representation of the image, which only normalizes the brightness levels of the image.

**Conversion of RGB/YIQ**

Assume that, R,G,B,Y €[0,1], I €[-0.5957,0.5957], Q€[-0.5226,0.5226

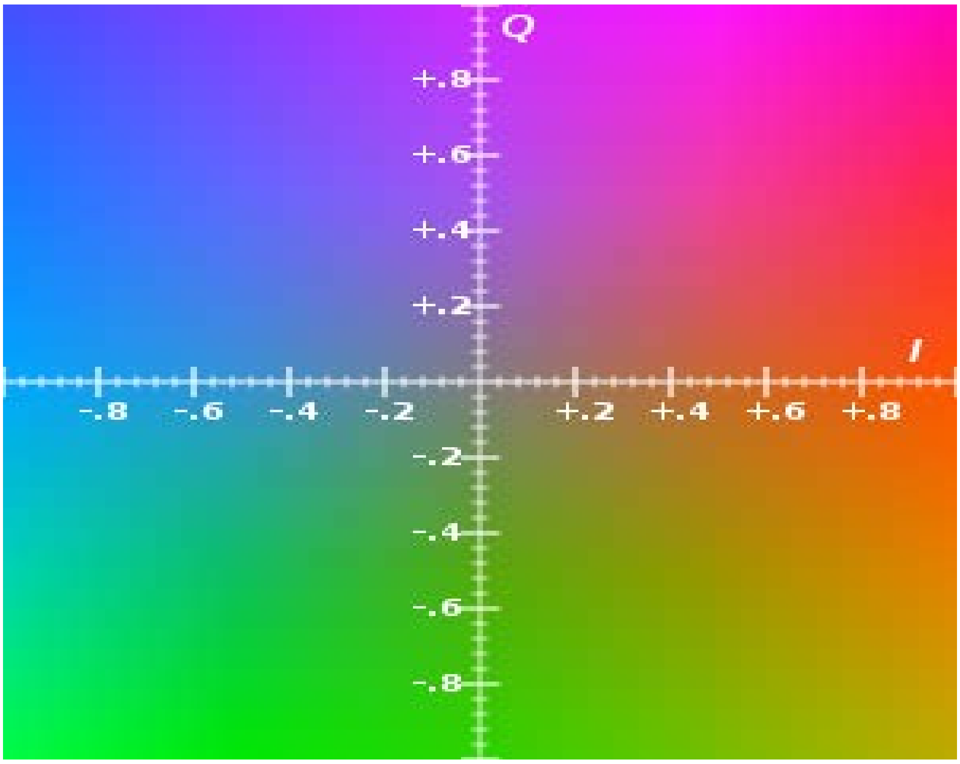
**From RGB to YIQ**

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.274 & -0.322 \\ 0.211 & -0.523 & 0.312 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

**From YIQ to RGB**

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 & 0.956 & 0.621 \\ 1 & -0.272 & -0.647 \\ 1 & -1.106 & 1.703 \end{bmatrix} \begin{bmatrix} Y \\ I \\ Q \end{bmatrix}$$

**Ref:** *https://en.wikipedia.org/wiki/YIQ*




#### 2.2.5.2. YUV

**YUV** is a color encoding system typically used as part of a color image pipeline. It encodes a color image or video taking human perception into account, allowing reduced bandwidth for chrominance components, thereby typically enabling transmission errors or compression artifacts to be more efficiently masked by the human perception than using a "direct" RGB-representation. Other color encodings have similar properties, and the main reason to implement or investigate properties of Y′UV would be for interfacing with analog or digital television or photographic equipment that conforms to certain Y′UV standards.

The scope of the terms Y′UV, YUV, YCbCr, YPbPr, etc., is sometimes ambiguous and overlapping. Historically, the terms YUV and Y′UV were used for a specific *analog encoding* of color information in television systems, while YCbCr was used for *digital encoding* of color information suited for video *and* still-image compression and transmission such as MPEG and JPEG. Today, the term YUV is commonly used in the computer industry to describe *file-formats* that are encoded using YCbCr.

The Y′UV model defines a color space in terms of one luma (Y′) and two chrominance (UV) components. The Y′UV color model is used in the PAL composite color video (excluding PAL-N) standard. Previous black-and-white systems used only luma (Y′) information. Color information (U and V) was added separately via a sub-carrier so that a black-and-white receiver would still be able to receive and display a color picture transmission in the receiver's native black-and-white format.

Y′ stands for the luma component (the brightness) and U and V are the chrominance (color) components; luminance is denoted by Y and lumaby Y′ – the prime symbols (') denote gamma compression,[1] with "luminance" meaning physical linear-space brightness, while "luma" is (nonlinear) perceptual brightness.

The YPbPr color model used in analog component video and its digital version YCbCr used in digital video are more or less derived from it, and are sometimes called Y′UV. ($C_B/P_B$ and $C_R/P_R$ are deviations from grey on blue–yellow and red–cyan axes, whereas U and V are blue–luminance and red–luminance differences respectively.) The Y′IQ color space used in the analog NTSC television broadcasting system is related to it, although in a more complex way. The YDbDr color space used in the analog SECAM and PAL-N television broadcasting systems, are also related.

As for etymology, Y, Y′, U, and V are not abbreviations. The use of the letter Y for luminance can be traced back to the choice of XYZprimaries. This lends itself naturally to the usage of the same letter in luma (Y′), which approximates a perceptually uniform correlate of luminance. Likewise, U and V were chosen to differentiate the U and V axes from those in other spaces, such as the x and y chromaticity space. See the equations below or compare the historical development of the math

**History**

Y′UV was invented when engineers wanted color television in a black-and-white infrastructure.[5] They needed a signal transmission method that was compatible with black-and-white (B&W) TV while being able to add color. The luma component already existed as the black and white signal; they added the UV signal to this as a solution.

The UV representation of chrominance was chosen over straight R and B signals because U and V are color difference signals. In other words, the U and V signals tell the television to shift the color of a certain pixel without altering its brightness. Or the U and V signals tell the monitor to make one color brighter at the cost of the other and by how much it should be shifted. The higher (or the lower when negative) the U and V values are, the more saturated (colorful) the pixel gets. The closer the U and V values get to zero, the lesser it shifts the color meaning that the red, green and blue lights will be more equally bright, producing a greyer pixel. This is the benefit of using color difference signals, i.e. instead of telling how much red there is to a color, it tells by how much it is more red than green or blue. In turn this meant that when the U and V signals would be zero or absent, it would just display a greyscale image. If R and B were to have been used, these would have non-zero values even in a B&W scene, requiring all three data-carrying signals. This was important in the early days of color television, because old black and white TV signals had no U and V signals present, meaning the color TV would just display it as B&W TV out of the box. In addition, black and white receivers could take the Y′ signal and ignore the U- and V-color signals, making Y′UV backward-compatible with all existing black-and-white equipment, input and output. If the color-TV standard wouldn't have used color difference signals, it could mean a color TV would make funny colors out of a B&W broadcast or it would need additional circuitry to translate the B&W signal to color. It was necessary to assign a narrower bandwidth to the chrominance channel because there was no additional bandwidth available. If some of the luminance information arrived via the chrominance channel (as it would have if RB signals were used instead of differential UV signals), B&W resolution would have been compromised.

**RGB/YUV Conversion**

$$\begin{bmatrix} Y' \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.14713 & -0.28886 & 0.436 \\ 0.615 & -0.51499 & -0.10001 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix},$$

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1.13983 \\ 1 & -0.39465 & -0.58060 \\ 1 & 2.03211 & 0 \end{bmatrix} \begin{bmatrix} Y' \\ U \\ V \end{bmatrix}.$$

To get a digital signal, Y′UV images can be sampled in several different ways but most important one is chroma subsampling.We was explained what is subsampling and chroma subsampling above.

### 2.2.5.3. YCbCr and YPbPr

**YCbCr**

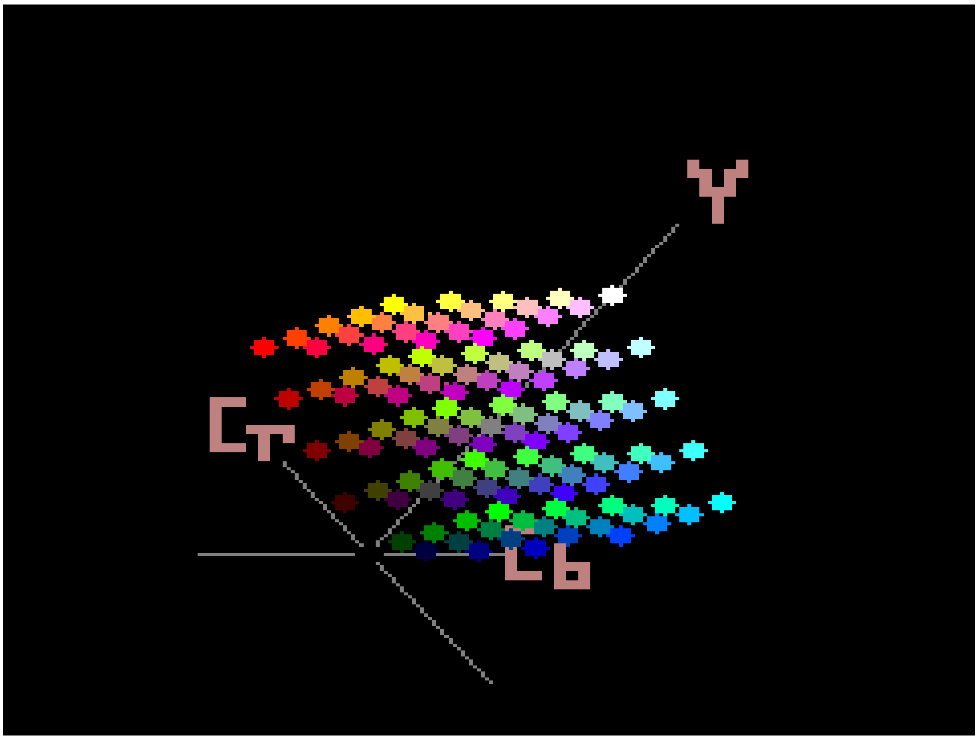
**CbCr**, **Y′CbCr**, or **Y Pb/Cb Pr/Cr**, also written as **$YC_BC_R$** or **$Y'C_BC_R$**, is a family of color spaces used as a part of the color image pipeline in video and digital photography systems. Y′ is the luma component and $C_B$ and $C_R$ are the blue-difference and red-difference chroma components. Y′ (with prime) is distinguished from Y, which is luminance, meaning that light intensity is nonlinearly encoded based on gamma corrected RGB primaries.

Y′CbCr color spaces are defined by a mathematical coordinate transformation from an associated RGB color space. If the underlying RGB color space is absolute, the Y′CbCr color space is an absolute color space as well; conversely, if the RGB space is ill-defined, so is Y′CbCr.

**Rationale**

Cathode ray tube displays are driven by red, green, and blue voltage signals, but these RGB signals are not efficient as a representation for storage and transmission, since they have a lot of redundancy.

YCbCr and Y′CbCr are a practical approximation to color processing and perceptual uniformity, where the primary colors corresponding roughly to red, green and blue are processed into perceptually meaningful information. By doing this, subsequent image/video processing, transmission and storage can do operations and introduce errors in perceptually meaningful ways. Y′CbCr is used to separate out a luma signal (Y′) that can be stored with high resolution or transmitted at high bandwidth, and two chroma components ($C_B$ and $C_R$) that can be bandwidth-reduced, subsampled, compressed, or otherwise treated separately for improved system efficiency.




**YCbCr** is sometimes abbreviated to **YCC**. Y′CbCr is often called YPbPr when used for analog component video, although the term Y′CbCr is commonly used for both systems, with or without the prime.

Y′CbCr is often confused with the YUV color space, and typically the terms YCbCr and YUV are used interchangeably, leading to some confusion. The main difference is that YUV is analog and YCbCr is digital.

One practical example would be decreasing the bandwidth or resolution allocated to "color" compared to "black and white", since humans are more sensitive to the black-and-white information (see image example to the right). This is called chroma subsampling.

Y′CbCr signals (prior to scaling and offsets to place the signals into digital form) are called YPbPr, and are created from the corresponding gamma-adjusted RGB (red, green and blue) source using three defined constants $K_R$, $K_G$, and $K_B$ as follows:

$$Y' = K_R \cdot R' + K_G \cdot G' + K_B \cdot B'$$
$$P_B = \frac{1}{2} \cdot \frac{B' - Y'}{1 - K_B}$$
$$P_R = \frac{1}{2} \cdot \frac{R' - Y'}{1 - K_R}$$

where $K_R$, $K_G$, and $K_B$ are ordinarily derived from the definition of the corresponding RGB space, and required to satisfy . (The equivalent matrixmanipulation is often referred to as the "color matrix".)

Here, the prime ′ symbols mean gamma correction is being used; thus R′, G′ and B′ nominally range from 0 to 1, with 0 representing the minimum intensity (e.g., for display of the color black) and 1 the maximum (e.g., for display of the color white). The resulting luma (Y) value will then have a nominal range from 0 to 1, and the chroma ($P_B$ and $P_R$) values will have a nominal range from -0.5 to +0.5. The reverse conversion process can be readily derived by inverting the above equations.

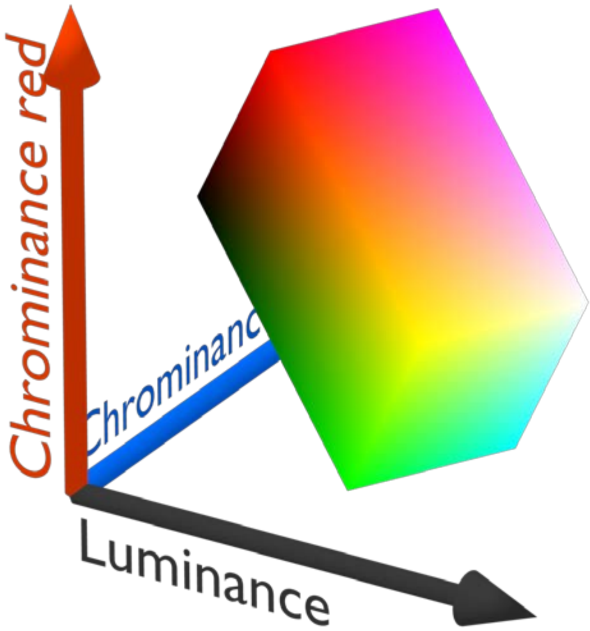
When representing the signals in digital form, the results are scaled and rounded, and offsets are typically added. For example, the scaling and offset applied to the Y′ component per specification (e.g. MPEG-2[1]) results in the value of 16 for black and the value of 235 for white when using an 8-bit representation. The standard has 8-bit digitized versions of $C_B$ and $C_R$ scaled to a different range of 16 to 240. Consequently, rescaling by the fraction (235-16)/(240-16) = 219/224 is sometimes required when doing color matrixing or processing in YCbCr space, resulting in quantization distortions when the subsequent processing is not performed using higher bit depths.

where $K_R$, $K_G$, and $K_B$ are ordinarily derived from the definition of the corresponding RGB space, and required to satisfy . (The equivalent matrixmanipulation is often referred to as the "color matrix".)

Here, the prime ′ symbols mean gamma correction is being used; thus R′, G′ and B′ nominally range from 0 to 1, with 0 representing the minimum intensity (e.g., for display of the

color black) and 1 the maximum (e.g., for display of the color white). The resulting luma (Y) value will then have a nominal range from 0 to 1, and the chroma ($P_B$ and $P_R$) values will have a nominal range from -0.5 to +0.5. The reverse conversion process can be readily derived by inverting the above equations.

When representing the signals in digital form, the results are scaled and rounded, and offsets are typically added. For example, the scaling and offset applied to the Y′ component per specification (e.g. MPEG-2[1]) results in the value of 16 for black and the value of 235 for white when using an 8-bit representation. The standard has 8-bit digitized versions of $C_B$ and $C_R$ scaled to a different range of 16 to 240. Consequently, rescaling by the fraction (235-16)/(240-16) = 219/224 is sometimes required when doing color matrixing or processing in YCbCr space, resulting in quantization distortions when the subsequent processing is not performed using higher bit depths.




**YPbPr**

**YPbPr** or **Y'PbPr**, also written as **$YP_BP_R$**, is a color space used in video electronics, in particular in reference to component videocables. YPbPr is the analog version of the YCbCr color space; the two are numerically equivalent but YPbPr is designed for use in analog systems while YCbCr is intended for digital video.

YPbPr is commonly referred to as *component video* by manufacturers; however, there are many types of component video, most of which are some form of RGB. Some video cards come with video-in video-out (VIVO) ports for connecting to component video devices.

YPbPr is converted from the RGB video signal, which is split into three components: Y, $P_B$, and $P_R$.

- Y carries luma (brightness or *luminance*) and synchronization (sync) information. Y = 0.2126 R + 0.7152 G + 0.0722 B. Before the advent of color television, the Y axis on an oscilloscope display of a video waveform represented the intensity of the scan line. With color, Y still represents intensity but it is a composite of the component colors.
- $P_B$ carries the difference between blue and luma (B − Y).
- $P_R$ carries the difference between red and luma (R − Y).

To send a green signal as a fourth component is redundant, as it can be derived using the blue, red and luma information.

When color signals were first added to the NTSC-encoded black and white video standard, the hue was represented by a phase shift of a color reference sub-carrier. P for phase information or phase shift has carried through to represent color information even in the case where there is no longer a phase shift used to represent hue. Thus, the Y $P_B$ $P_R$ nomenclature derives from engineering metrics developed for the NTSC color standard.[*citation needed*]

The same cables can be used for YPbPr and composite video. This means that the yellow, red, and white RCA connector cables commonly packaged with most audio/visual equipment can be used in place of the YPbPr connectors, provided the end user is careful to connect each cable to corresponding components at both ends. Also, many TVs use the green connection either for luma only or for composite video input. Since YPbPr is backwards compatible with the luminance portion of composite video even with just component video decoding one can still use composite video via this input, but only luma information will be displayed, along with the chroma dots. The same goes the other way around so long as 480i or 576i is used.

**Advantages**

Signals that use YPbPr offer enough separation that no color multiplexing is needed, so the quality of the extracted image is nearly identical to the signal before encoding. S-Video and composite video mix the signals together by means of electronic multiplexing (though S-Video does far better as it gives the whole video bandwith to luma and transmits chroma separately); however, more often than not the signal is degraded at the display end as (unless S-Video is used) the display is not able to separate the signals completely, although newer HDTVs (at least from reputable brands) tend to do a much better job at separation than most CRT TVs. Nevertheless it is possible for their multiplexed counterparts to interfere with each other (see dot crawl).

Among consumer analog interfaces, only YPbPr and analog RGB component video are capable of carrying non-interlaced video and resolutions higher than 480i or 576i, up to 1080i for YPbPr.

### 2.2.6. Cylindirical Color Spaces

**HSL** (**hue, saturation, lightness**) and **HSV** (**hue, saturation, value**) are alternative representations of the RGB color model, designed in the 1970s by computer graphicsresearchers to more closely align with the way human vision perceives color-making attributes. In these models, colors of each *hue* are arranged in a radial slice, around a central axis of neutral colors which ranges from black at the bottom to white at the top. The HSV representation models the way paints of different colors mix together, with the *saturation* dimension resembling various shades of brightly colored paint, and the *value* dimension resembling the mixture of those paints with varying amounts of black or white paint. The HSL model attempts to resemble more perceptual color models such as NCS or Munsell, placing fully saturated colors around a circle at a lightness value of ½, where a lightness value of 0 or 1 is fully black or white, respectively.

#### 2.2.6.1. HSV

**Hue**

Hue is the color portion of the color model, expressed as a number from 0 to 360 degrees:

| **Color** | **Angle** |
|---|---|
| Red | 0–60 |
| Yellow | 60-–20 |
| Green | 120–180 |
| Cyan | 180–240 |
| Blue | 240–300 |
| Magenta | 300–360 |

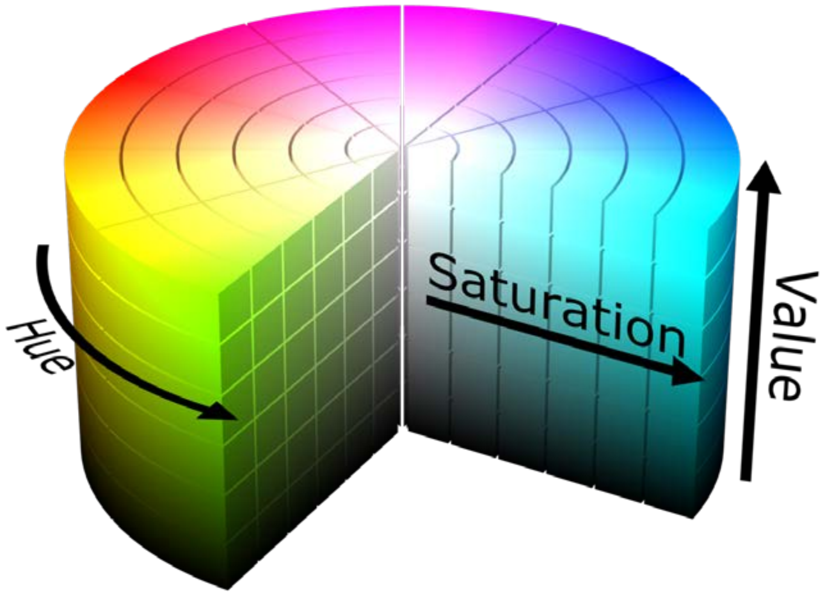
**Saturation**

Saturation is the amount of gray in the color, from 0 to 100 percent. Reducing the saturation toward zero to introduce more gray produces a faded effect.

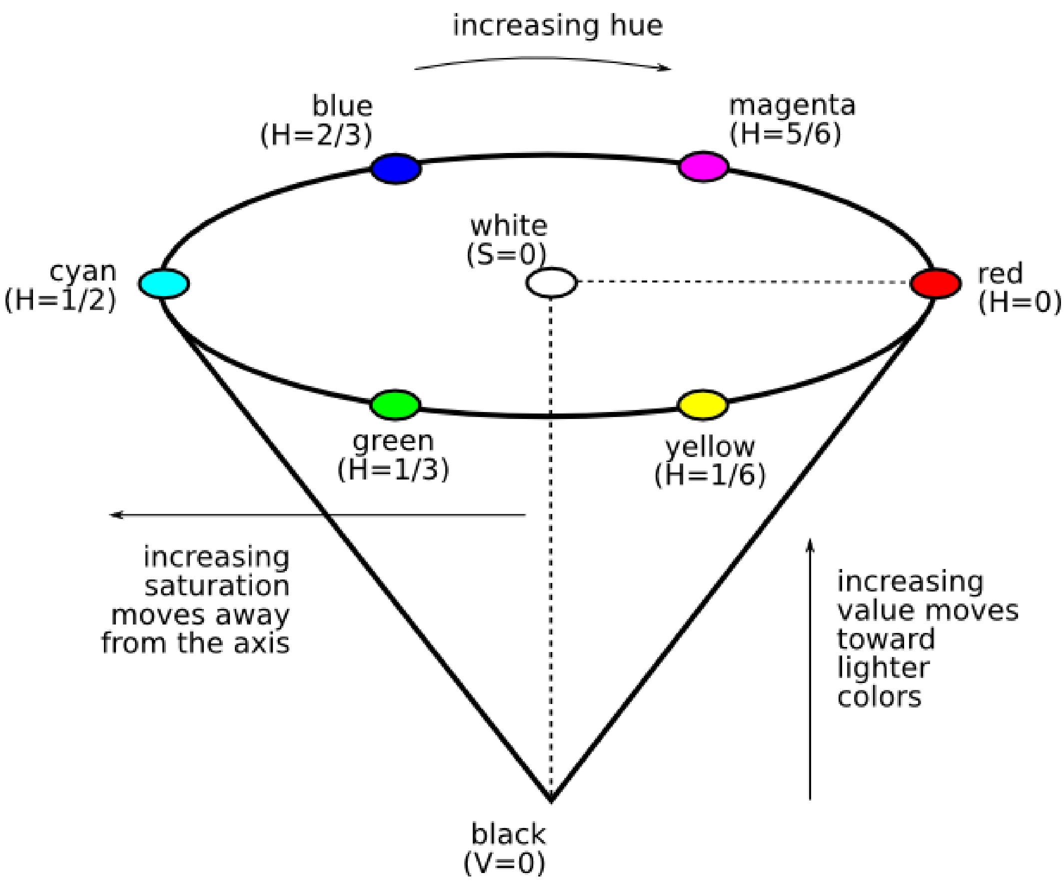
Sometimes, saturation is expressed in a range from just 0–1, where 0 is gray and 1 is a primary color.

**Value (or Brightness)**

Value works in conjunction with saturation and describes the brightness or intensity of the color, from 0–100 percent, where 0 is completely black, and 100 is the brightest and reveals the most color.




This diagram, called the *single-hexcone model of color space*, can help you visualize the meaning of the H, S, and V parameters.




- The outer edge of the top of the cone is the color wheel, with all the pure colors. The H parameter describes the angle around the wheel.
- The S (saturation) is zero for any color on the axis of the cone; the center of the top circle is white. An increase in the value of S corresponds to a movement away from the axis.
- The V (value or lightness) is zero for black. An increase in the value of V corresponds to a movement away from black and toward the top of the cone.

The Ostwald diagram corresponds to a slice of this cone. For example, the triangle between red, white, and black is the Ostwald diagram for the varieties of red.

**2.2.6.2. HSL**

The HSL color space was invented[further explanation needed] in 1938 by Georges Valensi as a method to add color encoding to existing monochrome (i.e. only containing the L signal) broadcasts, allowing existing receivers to receive new color broadcasts (in black and white) without modification as the luminance (black and white) signal is broadcast unmodified. It has been used in all major analog broadcast television encoding including NTSC, PAL and SECAM and all major digital broadcast systems and is the basis for composite video

Most televisions, computer displays, and projectors produce colors by combining red, green, and blue light in varying intensities—the so-called RGB additive primary colors. The resulting mixtures in RGB color space can reproduce a wide variety of colors (called a gamut); however, the relationship between the constituent amounts of red, green, and blue light and the resulting color is unintuitive, especially for inexperienced users, and for users familiar with subtractive color mixing of paints or traditional artists' models based on tints and shades (fig. 4). Furthermore, neither additive nor subtractive color models define color relationships the same way the human eye does.[C]

HSL, HSV, and related models can be derived via geometric strategies, or can be thought of as specific instances of a "generalized LHS model". The HSL and HSV model-builders took an RGB cube—with constituent amounts of red, green, and blue light in a color denoted $R, G, B \in [0, 1]$[E]—and tilted it on its corner, so that black rested at the origin with white directly above it along the vertical axis, then measured the hue of the colors in the cube by their angle around that axis, starting with red at 0°. Then they came up with a characterization of brightness/value/lightness, and defined saturation to range from 0 along the axis to 1 at the most colorful point for each pair of other parameters

**Hue and Chroma**

In each of our models, we calculate both *hue* and what this article will call *chroma*, after Joblove and Greenberg (1978), in the same way—that is, the hue of a color has the same numerical values in all of these models, as does its chroma. If we take our tilted RGB cube, and project it onto the "chromaticity plane" perpendicular to the neutral axis, our projection takes the shape of a hexagon, with red, yellow, green, cyan, blue, and magenta at its corners (fig. 9). *Hue* is roughly the angle of the vector to a point in the projection, with red at 0°, while *chroma* is roughly the distance of the point from the origin.[F][G]

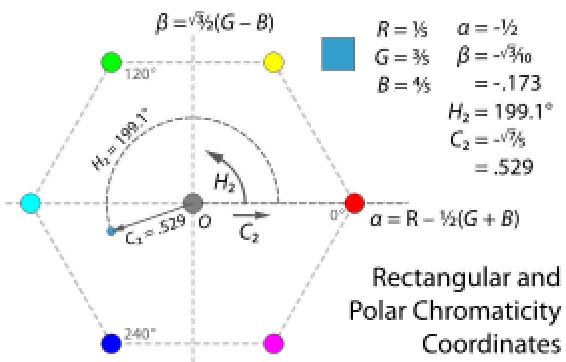
More precisely, both hue and chroma in this model are defined with respect to the hexagonal shape of the projection. The *chroma* is the proportion of the distance from the origin to the edge of the hexagon. In the lower part of the adjacent diagram, this is the ratio of lengths $OP/OP'$, or alternately the ratio of the radii of the two hexagons. This ratio is the difference between the largest and smallest values among $R$, $G$, or $B$ in a color. To make our definitions easier to write, we'll define these maximum, minimum, and chroma component values as $M$, $m$, and $C$, respectively.[H]

To understand why chroma can be written as $M - m$, notice that any neutral color, with $R = G = B$, projects onto the origin and so has 0 chroma. Thus if we add or subtract the same amount from all three of $R$, $G$, and $B$, we move vertically within our tilted cube, and do not change the projection. Therefore, any two colors ($R$, $G$, $B$) and ($R - m$, $G - m$, $B - m$) project on the same point, and have the same chroma. The chroma of a color with one of its components equal to zero ($m = 0$) is simply the maximum of the other two components. This chroma is $M$ in the particular case of a color with a zero component, and $M - m$ in general.

The *hue* is the proportion of the distance around the edge of the hexagon which passes through the projected point, originally measured on the range [0, 1) but now typically measured in degrees [0°, 360°). For points which project onto the origin in the chromaticity plane (i.e., grays), hue is undefined. Mathematically, this definition of hue is written piecewise:[I]

Sometimes, neutral colors (i.e. with $C = 0$) are assigned a hue of 0° for convenience of representation.

These definitions amount to a geometric warping of hexagons into circles: each side of the hexagon is mapped linearly onto a 60° arc of the circle (fig. 10). After such a transformation, hue is precisely the angle around the origin and chroma the distance from the origin: the angle and magnitude of the vector pointing to a color.




Sometimes for image analysis applications, this hexagon-to-circle transformation is skipped, and *hue* and *chroma*(we'll denote these $H_2$ and $C_2$) are defined by the usual cartesian-to-polar coordinate transformations (fig. 11). The easiest way to derive those is via a pair of cartesian chromaticity coordinates which we'll call $\alpha$ and $\beta$:

(The atan2 function, a "two-argument arctangent", computes the angle from a cartesian coordinate pair.)

Notice that these two definitions of hue ($H$ and $H_2$) nearly coincide, with a maximum difference between them for any color of about 1.12°—which occurs at twelve particular hues, for instance $H = 13.38°$, $H_2 = 12.26°$—and with $H = H_2$for every multiple of 30°. The two definitions of chroma ($C$ and $C_2$) differ more substantially: they are equal at the corners of our hexagon, but at points halfway between two corners, such as $H = H_2 = 30°$, we have $C = 1$, but $C_2 = \sqrt{\frac{3}{4}} \approx 0.866$, a difference of about 13.4%.

**Lightness**

While the definition of *hue* is relatively uncontroversial—it roughly satisfies the criterion that colors of the same perceived hue should have the same numerical hue—the definition of a *lightness* or *value* dimension is less obvious: there are several possibilities depending on the purpose and goals of the representation. Here are four of the most common (fig. 12; three of these are also shown in fig. 8):

- The simplest definition is just the average of the three components, in the HSI model called *intensity* (fig. 12a). This is simply the projection of a point onto the neutral axis—the vertical height of a point in our tilted cube. The advantage is that, together with Euclidean-distance calculations of hue and chroma, this representation preserves distances and angles from the geometry of the RGB cube.[23][25]

- In the HSV "hexcone" model, *value* is defined as the largest component of a color, our $M$ above (fig. 12b). This places all three primaries, and also all of the "secondary colors"—cyan, yellow, and magenta—into a plane with white, forming a hexagonal pyramid out of the RGB cube.[10]

- In the HSL "bi-hexcone" model, *lightness* is defined as the average of the largest and smallest color components (fig. 12c). This definition also puts the primary and secondary colors into a plane, but a plane passing halfway between white and black. The resulting color solid is a double-cone similar to Ostwald's, shown above.[11]

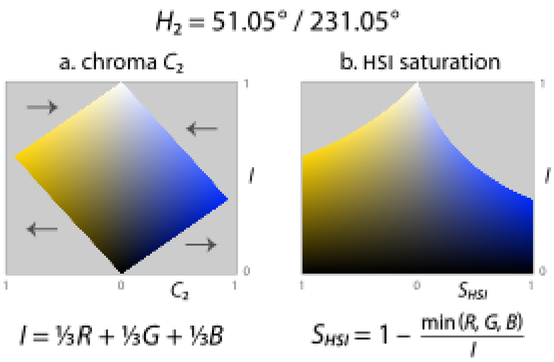
- A more perceptually relevant alternative is to use *luma*, $Y'$, as a lightness dimension (fig. 12d). Luma is the weighted average of gamma-corrected $R$, $G$, and $B$, based on their contribution to perceived lightness, long used as the monochromatic dimension in color television broadcast. For the Rec. 709 primaries used in sRGB, $Y'_{709} = 0.21R + 0.72G + 0.07B$; for the Rec. 601 NTSCprimaries, $Y'_{601} \approx 0.30R + 0.59G + 0.11B$; for other primaries different coefficients should be used.[26][J]

All four of these leave the neutral axis alone. That is, for colors with $R = G = B$, any of the four formulations yields a lightness equal to the value of $R$, $G$, or $B$.

**Saturation**

When encoding colors in a hue/lightness/chroma or hue/value/chroma model (using the definitions from the previous two sections) model, not all combinations of lightness (or value) and chroma are meaningful: that is, half of the colors denotable using $H \in [0°, 360°)$, $C \in [0, 1]$, and $V \in [0, 1]$ fall outside the RGB gamut (the gray parts of the slices in figure 14). The creators of these models considered this a problem for some uses. For example, in a color selection interface with two of the dimensions in a rectangle and the third on a slider, half of that rectangle is made of unused space. Now imagine we have a slider for lightness: the user's intent when adjusting this slider is potentially ambiguous: how should the software deal with out-of-gamut colors? Or conversely, If the user has selected as colorful as possible a dark purple, and then shifts the lightness slider upward, what should be done: would the user prefer to see a lighter purple still as colorful as possible for the given hue and lightness, or a lighter purple of exactly the same chroma as the original color?[11]

To solve problems such as these, the HSL and HSV models scale the chroma so that it always fits into the range [0, 1] for every combination of hue and lightness or value, calling the new attribute *saturation* in both cases (fig. 14). To calculate either, simply divide the chroma by the maximum chroma for that value or lightness.




The HSI model commonly used for computer vision, which takes $H_2$ as a hue dimension and the component average $I$("intensity") as a lightness dimension, does not attempt to "fill" a cylinder by its definition of saturation. Instead of presenting color choice or modification interfaces to end users, the goal of HSI is to facilitate separation of shapes in an image. Saturation is therefore defined in line with the psychometric definition: chroma relative to lightness (fig. 15). See the Use in image analysis section of this article.[28]

Using the same name for these three different definitions of saturation leads to some confusion, as the three attributes describe substantially different color relationships; in HSV and HSI, the term roughly matches the psychometric definition, of a chroma of a color relative to its own lightness, but in HSL it does not come close. Even worse, the word *saturation* is also often used for one of the measurements we call chroma above ($C$ or $C_2$).

**Usage**

The original purpose of HSL and HSV and similar models, and their most common current application, is in color selection tools. At their simplest, some such color pickers provide three sliders, one for each attribute. Most, however, show a two-dimensional slice through the model, along with a slider controlling which particular slice is shown. The latter type of GUI exhibits great variety, because of the choice of cylinders, hexagonal prisms, or cones/bicones that the models suggest (see the diagram near the top of the page). Several color choosers from the 1990s are shown to the right, most of which have remained nearly unchanged in the intervening time: today, nearly every computer color chooser uses HSL or HSV, at least as an option. Some more sophisticated variants are designed for choosing whole sets of colors, basing their suggestions of compatible colors on the HSL or HSV relationships between them.[M]

Most web applications needing color selection also base their tools on HSL or HSV, and pre-packaged open source color choosers exist for most major web front-end frameworks. The CSS 3 specification allows web authors to specify colors for their pages directly with HSL coordinates.

HSL and HSV are sometimes used to define gradients for data visualization, as in maps or medical images. For example, the popular GIS program ArcGIS historically applied customizable HSV-based gradients to numerical geographical data.

Image editing software also commonly includes tools for adjusting colors with reference to HSL or HSV coordinates, or to coordinates in a model based on the "intensity" or luma defined above. In particular, tools with a pair of "hue" and "saturation" sliders are commonplace, dating to at least the late-1980s, but various more complicated color tools have also been implemented. For instance, the Unix image viewer and color editor xvallowed six user-definable hue ($H$) ranges to be rotated and resized, included a dial-like control for saturation ($S_{HSV}$), and a curves-like interface for controlling value ($V$)—see fig. 17. The image editor Picture Window Proincludes a "color correction" tool which affords complex remapping of points in a hue/saturation plane relative to either HSL or HSV space.[P]

Video editors also use these models. For example, both Avid and Final Cut Pro include color tools based on HSL or a similar geometry for use adjusting the color in video. With the Avid tool, users pick a vector by clicking a point within the hue/saturation circle to shift all the colors at some lightness level (shadows, mid-tones, highlights) by that vector.

### Image Analysis

HSL, HSV, HSI, or related models are often used in computer vision and image analysis for feature detection or image segmentation. The applications of such tools include object detection, for instance in robot vision; object recognition, for instance of faces, text, or license plates; content-based image retrieval; and analysis of medical images.[28]

For the most part, computer vision algorithms used on color images are straightforward extensions to algorithms designed for grayscale images, for instance k-means or fuzzy clustering of pixel colors, or canny edge detection. At the simplest, each color component is separately passed through the same algorithm. It is important, therefore, that the features of interest can be distinguished in the color dimensions used. Because the *R*, *G*, and *B* components of an object's color in a digital image are all correlated with the amount of light hitting the object, and therefore with each other, image descriptions in terms of those components make object discrimination difficult. Descriptions in terms of hue/lightness/chroma or hue/lightness/saturation are often more relevant.[28]

Starting in the late 1970s, transformations like HSV or HSI were used as a compromise between effectiveness for segmentation and computational complexity. They can be thought of as similar in approach and intent to the neural processing used by human color vision, without agreeing in particulars: if the goal is object detection, roughly separating hue, lightness, and chroma or saturation is effective, but there is no particular reason to strictly mimic human color response. John Kender's 1976 master's thesis proposed the HSI model. Ohta et al. (1980) instead used a model made up of dimensions similar to those we have called $I$, $\alpha$, and $\beta$. In recent years, such models have continued to see wide use, as their performance compares favorably with more complex models, and their computational simplicity remains compelling.

### Disadvantages

While HSL, HSV, and related spaces serve well enough to, for instance, choose a single color, they ignore much of the complexity of color appearance. Essentially, they trade off perceptual relevance for computation speed, from a time in computing history (high-end 1970s graphics workstations, or mid-1990s consumer desktops) when more sophisticated models would have been too computationally expensive.[S]

HSL and HSV are simple transformations of RGB which preserve symmetries in the RGB cube unrelated to human perception, such that its *R*, *G*, and *B* corners are equidistant from the neutral axis, and equally spaced around it. If we plot the RGB gamut in a more perceptually-uniform space, such as CIELAB (see below), it becomes immediately clear that the red, green, and blue primaries do not have the same lightness or chroma, or evenly spaced hues.

Furthermore, different RGB displays use different primaries, and so have different gamuts. Because HSL and HSV are defined purely with reference to some RGB space, they are not absolute color spaces: to specify a color precisely requires reporting not only HSL or HSV values, but also the characteristics of the RGB space they are based on, including the gamma correction in use.

If we take an image and extract the hue, saturation, and lightness or value components, and then compare these to the components of the same name as defined by color scientists, we can quickly see the difference, perceptually. For example, examine the following images of a fire breather . The original is in the sRGB colorspace. CIELAB $L^*$ is a CIE-defined achromatic lightness quantity (dependent solely on the perceptually achromatic luminance $Y$, but not the mixed-chromatic components $X$ or $Z$, of the CIEXYZ colorspace from which the sRGB colorspace itself is derived), and it is plain that this appears similar in perceptual lightness to the original color image. Luma is roughly similar, but differs somewhat at high chroma, where it deviates most from depending solely on the true achromatic luminance ($Y$, or equivalently $L^*$) and is influenced by the colorimetric chromaticity ($x$,$y$, or equivalently, $a^*$,$b^*$ of CIELAB). HSL $L$ and HSV $V$, by contrast, diverge substantially from perceptual lightness.

Color photograph (sRGB colorspace).

CIELAB $L^*$ (further transformed back to sRGB for consistent display).

Rec. 601 luma *Y′*.

Component average: "intensity" *I*.

HSV value *V*.

HSL lightness *L*.

### ***2.1.7. CIE Color Spaces***

CIE 1931 is a Color Matching System. Color matching does not attempt to describe how colors appear to humans, color matching tells us how to numerically specify a measured color, and then later accurately reproduce that measured color (e.g. in print or digital displays).

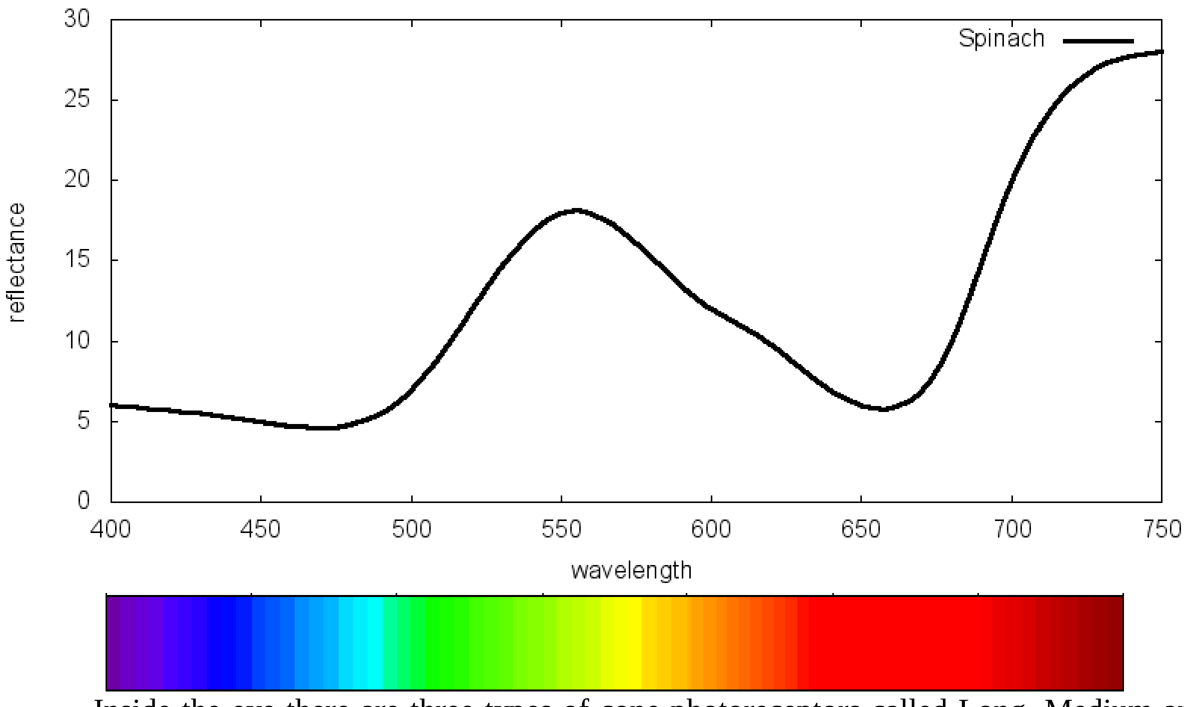
To reiterate that point, Color Matching Systems are not focused on describing color with qualities like hue or saturation, they just tell us what combinations of light appear to be the same color to most people (they "match"). Color matching allows gives us a basic framework for color reproduction.

You can think of CIE 1931 like you would Newtonian physics. It's a mathematical generalization of human color vision, that allows us to define, and accurately reproduce colors in most situations. It does not fully describe the subjective and complicated process of human color vision, it's not without edge cases, but it is relatively easy to work with.

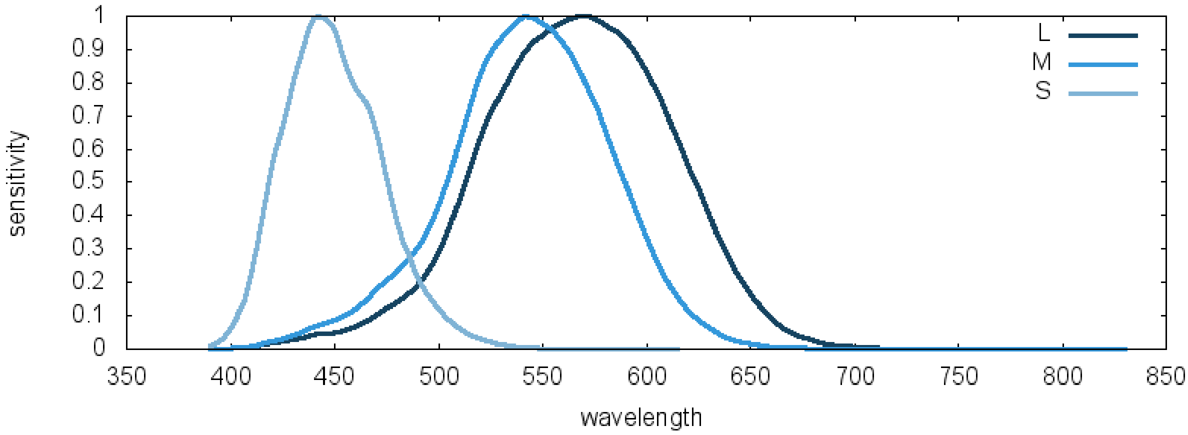
**A. Trichromatic Color Theory**

A principal quality of light is its wavelength. Humans can perceive light with waves as small as ~380nm and as large as ~750nm.

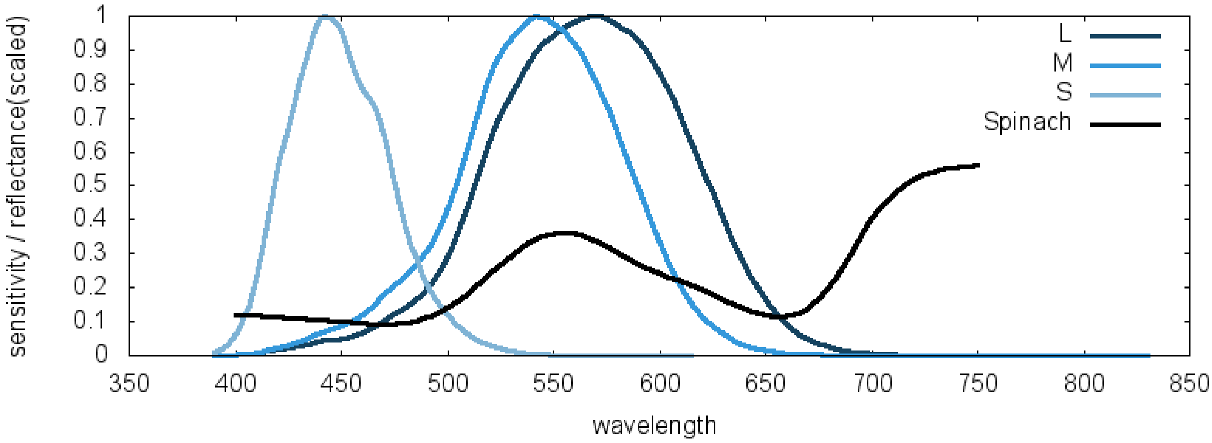
The sun emits light across many wavelengths and objects around us reflect some wavelengths and absorb some wavelengths. Below is the spectral distribution of a spinach leaf illuminated by sunlight. The leaf absorbs most wavelengths as heat, but reflects visible light around 550nm. We see this particular reflected spectral composition as green.




Inside the eye there are three types of cone photoreceptors called Long, Medium and Short that contribute to color discrimination. They are all sensitive to different, yet overlapping, wavelengths of light. They are commonly associated with the color they are most sensitive too, L = red, M = green, S = blue.



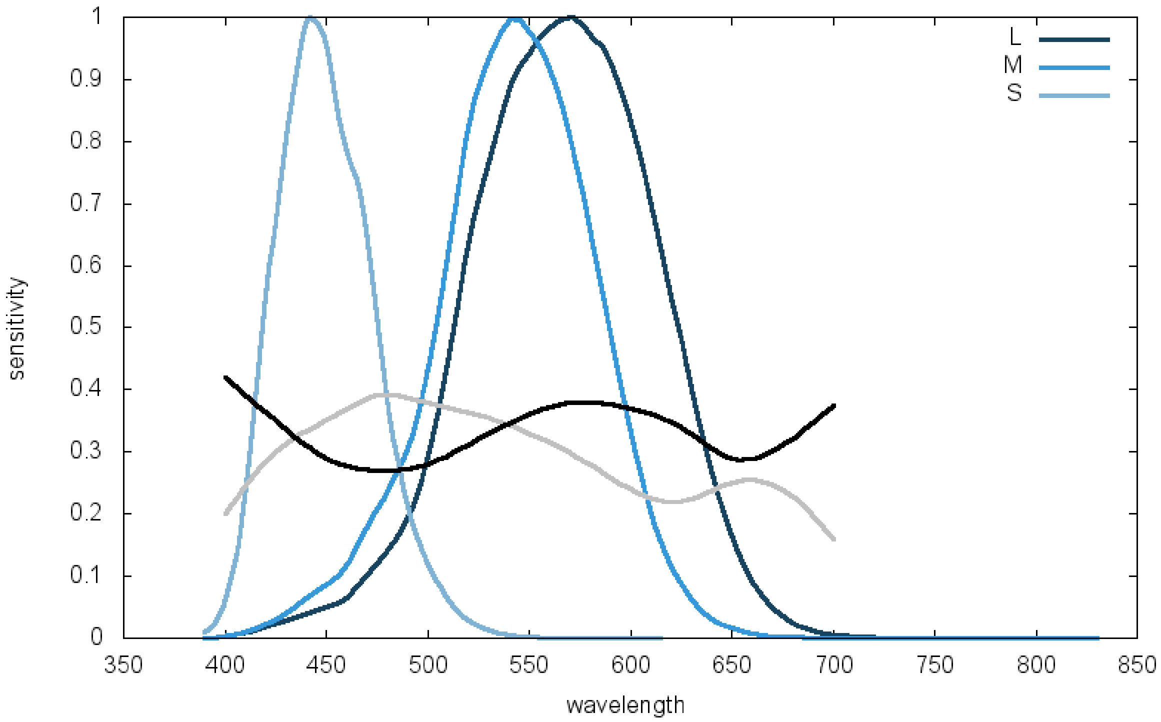

When you view the charts overlaid, you can see that the spinach mostly reflects light outside of the eye's visual range, and inside our range it mostly reflects light centered around our M cone.




The eye's cone system takes a complex spectral distribution (like the spinach) and reduces it down to three numerical values, each representing how much the three cones were stimulated. These cone values are fed into the next part of the brain's visual processing pipeline, which we won't go into.

This is important: The trichromatic process is lossy, you can't go from the output of the cones back to the original spectral distribution. This is because different spectral distributions can stimulate the cones in the exact same way.

Shown here are two different spectral distributions that are indistinguishable to the human eye. You can see that the M cone is stimulated by both distributions equally, just on opposite ends of the cone's sensitivity range.




You could imagine a leaf and a green car that look the same to you, but physically have different reflectance properties. It turns out every color (or, unique cone output) can be created from many different spectral distributions. Color science started to make a lot more sense when I finally understood this.

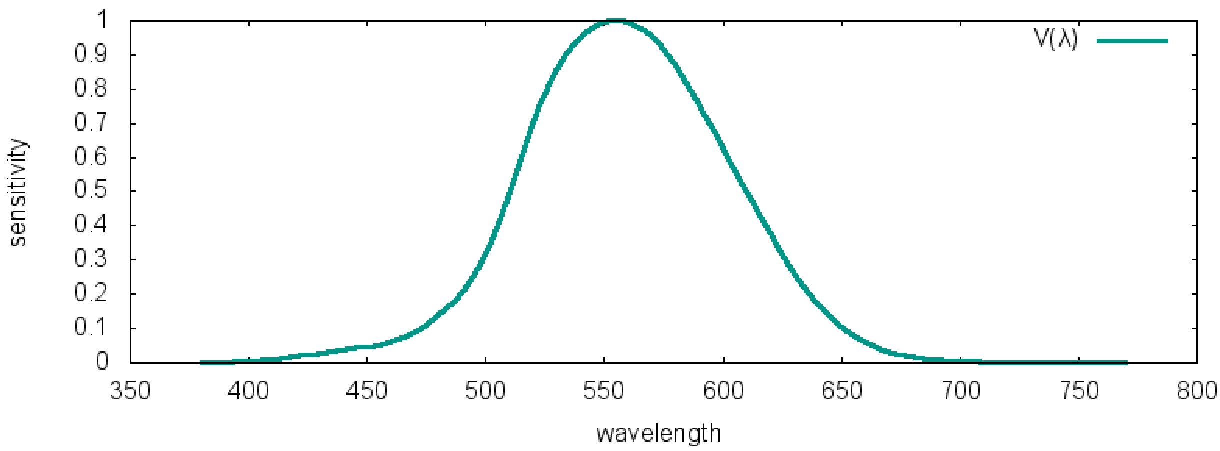
This phenomenon is called metemerism and it has huge ramifications for color reproduction. It means we don't need the original light to reproduce an observed color. We can reproduce a color if we can create a spectral distribution of light with the same cone response as the original distribution.

This is why you can create a color that looks like spectral yellow on a LCD display that has no yellow lights. The LCD can combine red and green light in the right amounts to mimic the human cone response of spectral yellow light.

CIE 1931 is a model that tell us how to create these matches. Excited yet?

**B) 1924 Luminous Efficiency Function, V(λ)**

Before the fundamental work of colorimetry occurred CIE published a function, V(λ), that describes the human eye's sensitivity to light at different wavelengths in daylight. (λ means wavelength)




What this tell us is that two lights, 550nm and 400nm can have the same radiance (an objective unit) but appear to have different brightnesses.

Or to put it another way, if you have a green and blue light that appear to be equally bright, then you know that the blue light is more luminous.

Note: The 1924 V(λ) was eventually shown to under represent our sensitivity at the blue end of the spectrum.

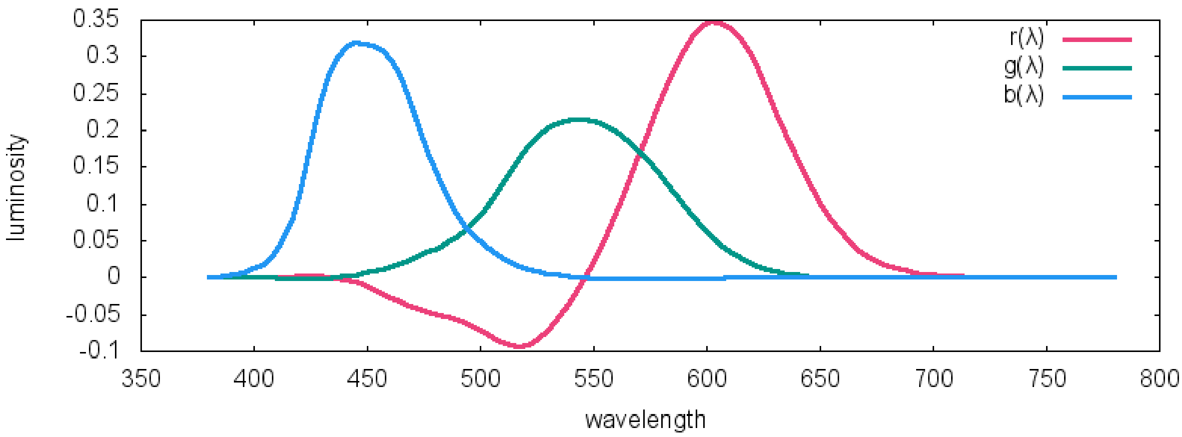
**C) 1931 RGB Color Matching Functions**

CIE 1931 contains 3 functions called the RGB color matching functions.

Let's say you have three lights, a red and green and a blue with precisely known single wavelengths. If you point these lights at the same spot on the wall is it possible to adjust the power of each light until the spot had the exactly same color as 540nm light?

This is exactly the kind of question color matching functions can answer. Color Matching functions have three fixed primary colors and each function outputs the amount of primary needed to create a desired color when all three are mixed.

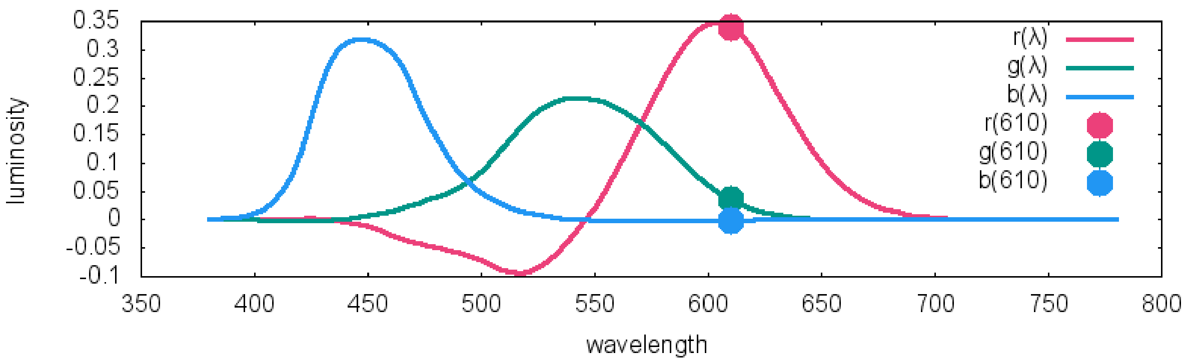
700nm, 546.1nm and 435.8nm are the primaries of the 1931 RGB color matching functions: r(λ), g(λ), b(λ). Plotted below.




Let's use this to make some matches in our hypothetical light setup. Note, this scenario is going to seem very specific and contrived but later we will show how it can be generalized to explain all color matches.

We'll pick a target wavelength of 610nm, the goal is to reproduce the color of this light with our primary lights.

λ = 610nm r(λ) = 0.34756 g(λ) = 0.04776 b(λ) = -0.00038




One quirk about these functions, they are scaled as if all the primary lights are equally bright. As mentioned earlier, a blue light must be more luminous than an equally bright green light. So in order to get absolute luminance needed for our match, we must un-scale the values using a set of relative luminance values provided to us along with the function.

Lr = 1 # r luminance scale

Lg = 4.5907 # g luminance scale

Lb = 0.0601 # b luminance scale

λ = 610nm

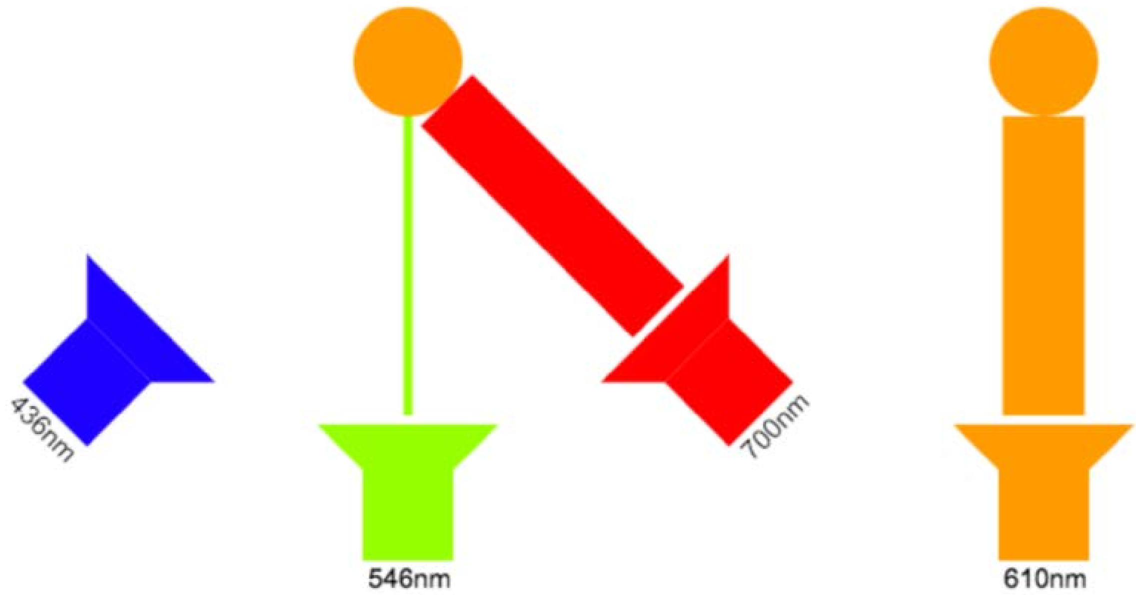
R = r(λ)/Lr = 0.34756

G = g(λ)/Lg = 0.04776

B = b(λ)/Lb = -0.00038

The resulting values R, G, and B are called tristimulus values, they are the required luminosity of the test lights to achieve a color match.

Instead of subjecting you to the colorimetric equations that describes this match, here's a diagram of the scenario.




Since blue is negative, we leave the light off because negative light isn't possible. What this means is that we actually can't create a perfect match of 610nm light with these primaries, but by using the positive amounts of red and green we can create something very close. Negative values are explored more in the next section.

The exercise we've described is not a day-to-day use of color matching functions, but as we will see arranging these lights is similar to how you might vary the intensity of rgb sub-pixels to create an LCD pixel of a desired color.

**D) Wright Guild Color Matching Experiments**

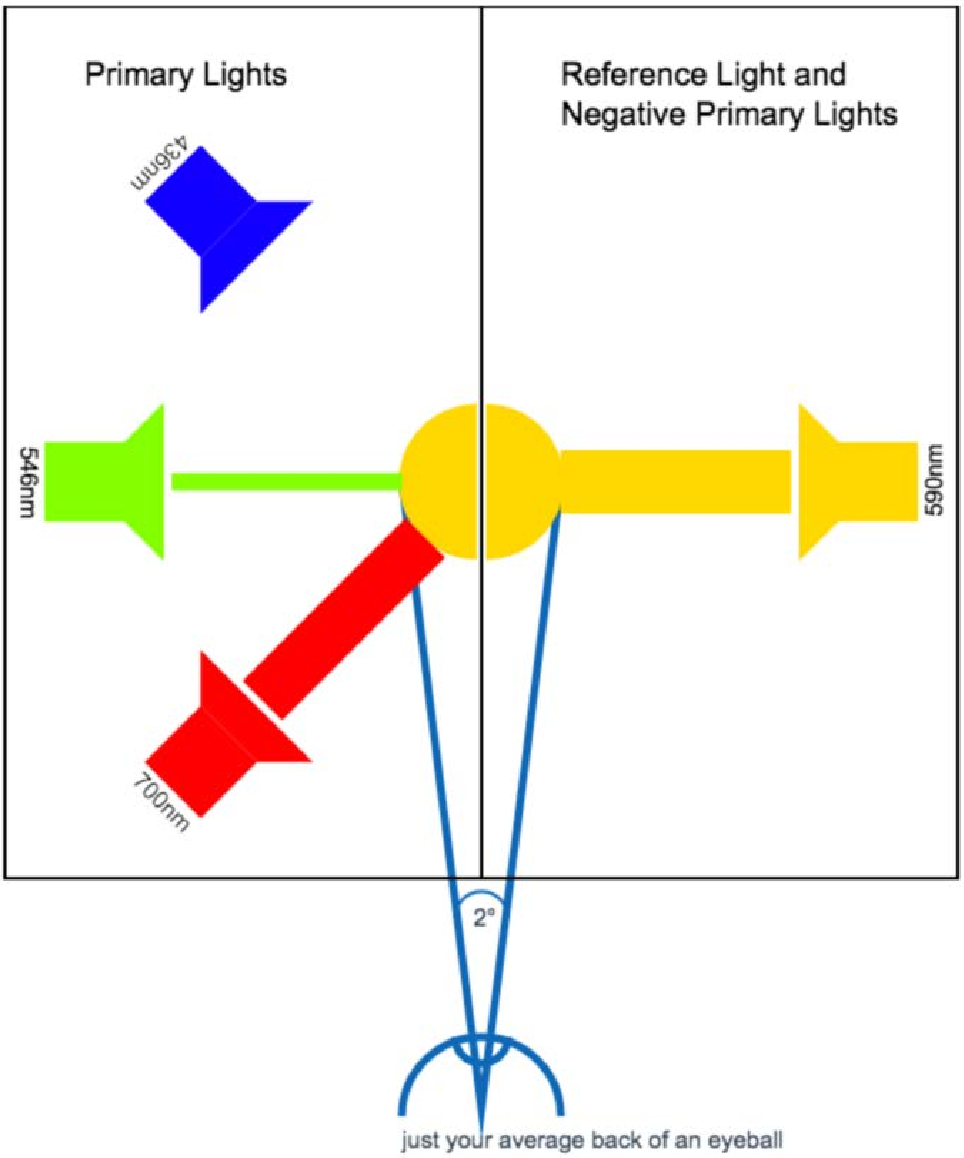
I used the lighting arrangement in the previous section as a demonstration of a physical application of the color matching functions, but it actually has its origins in the creation of those functions.

In the 1920s two color scientists, W. D. Wright and J. Guild, each performed similar color vision experiments. Wright performed his experiment on 10 subjects, Guild used 7. Their results agreed with each other so well that they were combined by CIE to create the RGB color matching functions we've been discussing.

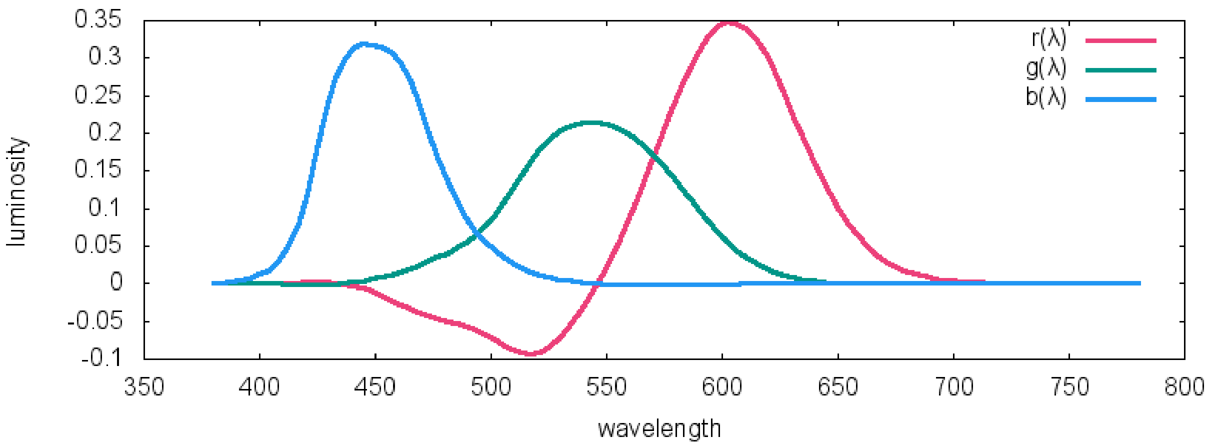
Color vision can sometimes vary, so it was important to find test subjects with no color vision deficiencies or yellowing of the lens (happens with old age) in order to create functions that worked for the average person. The combination of the Wright Guild subject data is known as the 1931 Standard Observer.

The experiment setup asks a subject to adjust the values of 3 primary lights until they produce a color indistinguishable from a reference light. This is repeated using a reference light for every visible wavelength (at some increment). When this is done you have the values of your color matching functions.

Here’s an example. On the right is a monochromatic (single wavelength) yellow, on the left the test subject has adjusted the primary lights so that their mixture is identical to the monochromatic light.



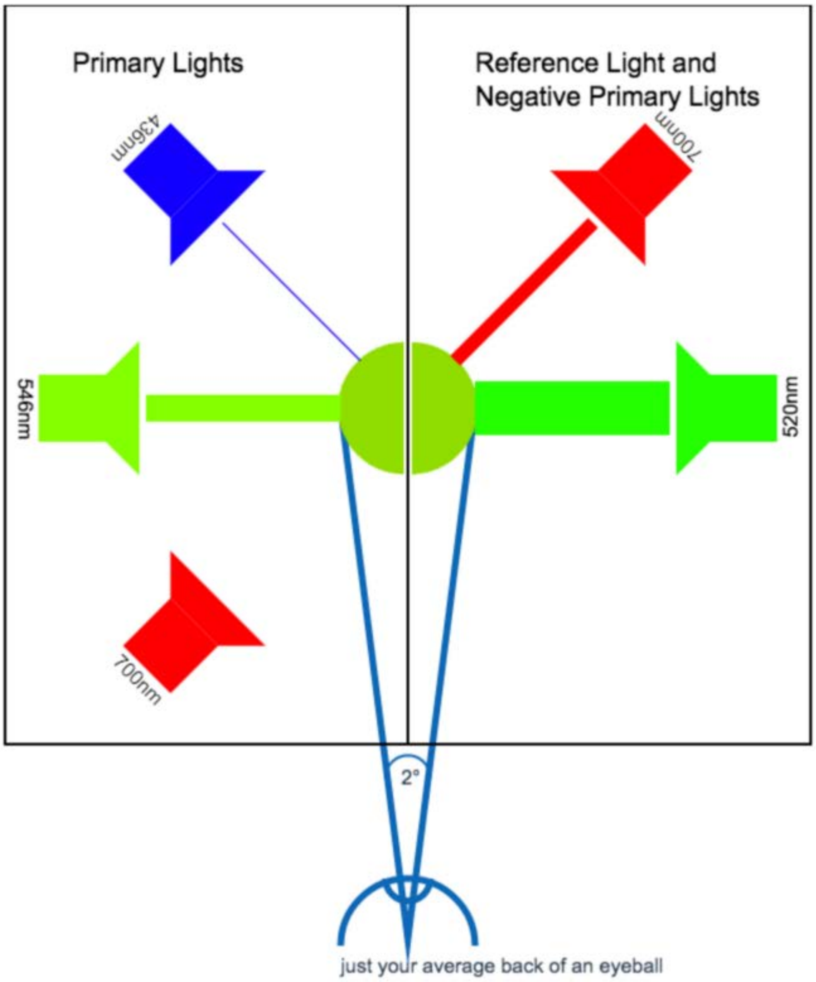

**About those negative values**




You might have noticed in the color matching functions that sometimes they call for a negative amount of light to achieve a match, such as 520nm. What this means is that the test subject was unable to achieve a match using positive values of the primary lights. To address this, some of the primary lights were mixed in on the opposite side of the screen with the reference light, until a color match was able to be made.

As it turns out, no 3 primaries can create every spectral color. By using the negative light trick researchers were able to quantify a color match in spectral ranges not matchable by their chosen primaries.

520nm is an example of a bright green that wasn't achievable with the test primaries unless a negative amount of red was used.



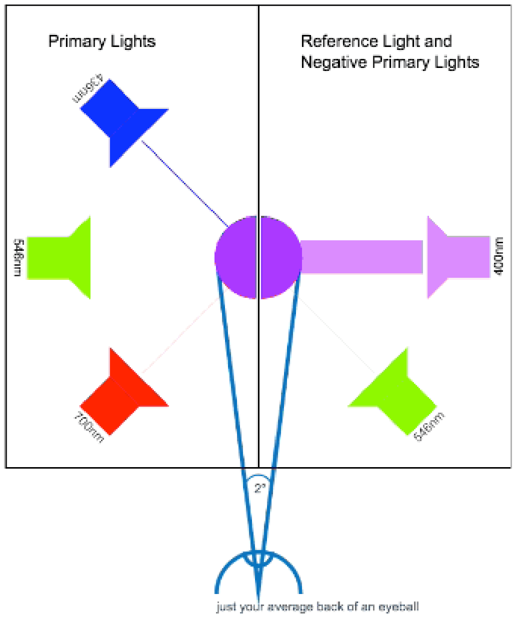

Restrictions on viewing conditions

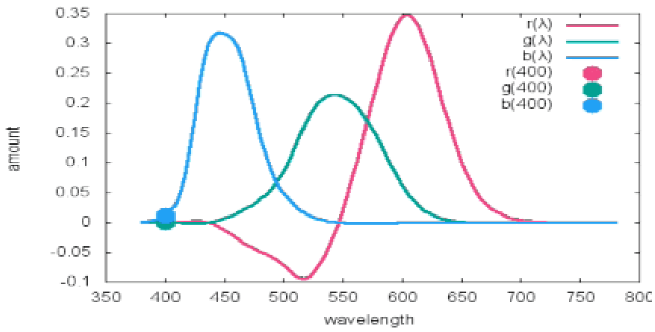
If you hold out your thumb in front of you, it takes up a 2° field of view. If you make a fist, that's 10°.

The 1931 Standard Observer is only valid for colors viewed at a 2° field of view. This places the light on a spot on the back of your eye called the Fovea. This is a spot with high cone density, giving you maximum color discrimination and limited rod interference.

Color researchers realized that a 10° color matching function would be more representative of day to day color perception so color matching experiments were repeated at 10° and published as the 1964 10° Supplementary Standard Observer.

This might be more fun than useful but here is an animation cycling through all of the configurations of the 1931 2° tests.




**E) RGB Chromaticity Coordinates**

Colors specified as three tristimulus values, like in the RGB, are difficult to visualize. It's not easy to imagine what a color looks like at a specific coordinate inside a three dimensional space.

One way we can simplify this problem is to remove the intensity dimension from the data, so that dark red and light red are the same value. This concept of color minus intensity is called Chromaticity.

This is what's happening when you adjust the brightness on your monitor, you are changing the colors of the pixels but the chromaticity of pixel remains the same. This shows why we don't always care about the original intensity information, and why Chromaticity is a useful concept.

This is the equation to convert tristimulus values into chromaticity coordinates.

RGB are tristimulus values

$r = R/(R+G+B)$

$g = G/(R+G+B)$

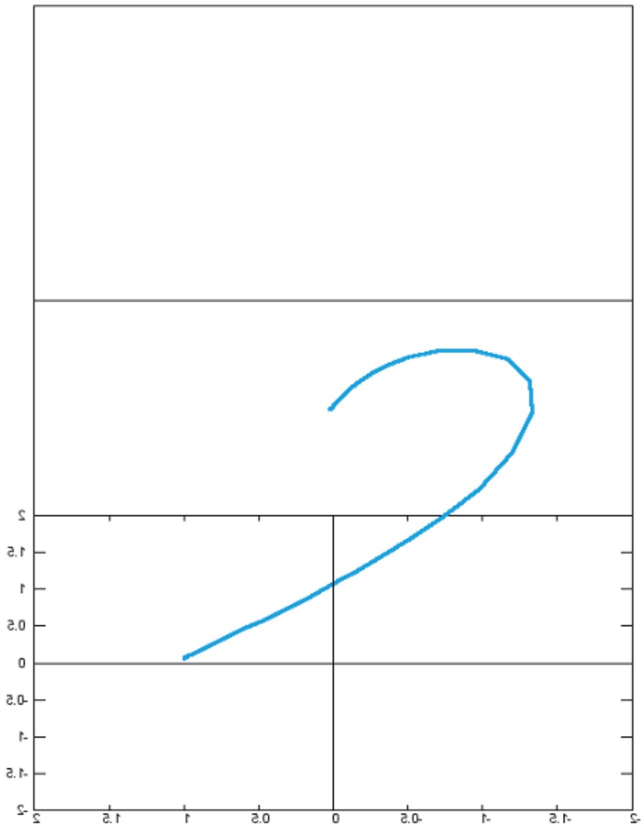
$b = B/(R+G+B)$

rgb are chromaticity coordinates and $r + g + b = 1$

rgb is about the ratio, it tells you "To match 400nm light, much more blue is needed than red or green"

RGB is about absolute amounts, it tells you "To match 400nm light, very little of any primary light is needed"

We can plot every value of rgb as a point in three dimensions. This gives us a curve that represents the chromaticity of each spectral wavelength of light, also called a spectral locus.



Even though we still have three values (rgb), they all sum to one, which means we can always determine the third value from the first two. Like so:

r = -1.5 #known

g = 2 #known

b = ? #unknown

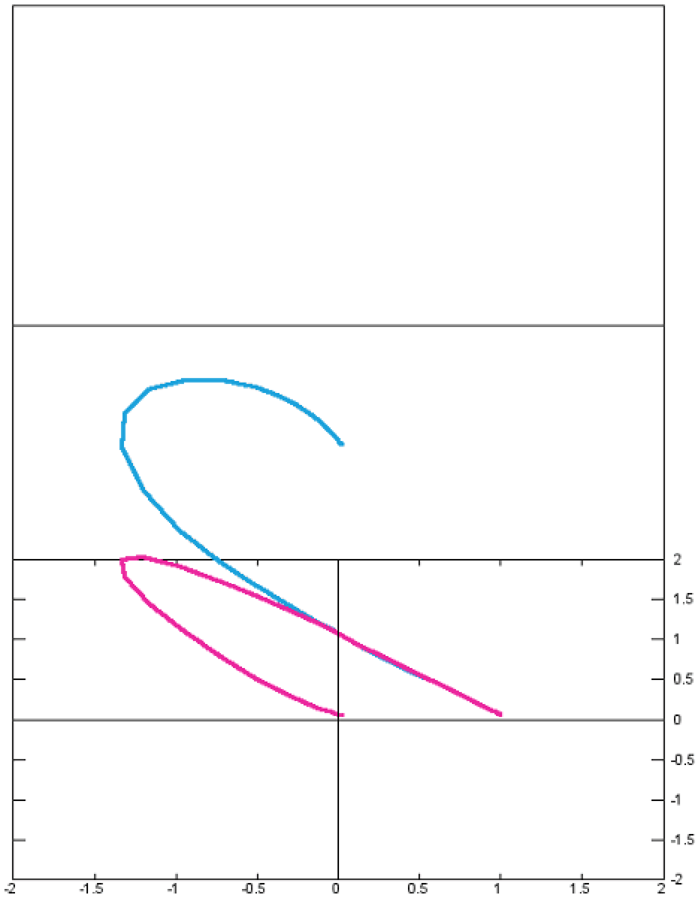
if r + g + b must equal 1

then b = 1 - g - r

so b must be 0.5

By removing intensity information from the data (in the conversion from RGB to rgb), we have actually converted a three dimensional space into two dimensions, which we can take advantage of for simplified visualizations.

Since we can safely drop a dimension without losing information, let's convert the b axis to all zeroes and just plot rg in two dimensions. This is called a projection to the rg plane.

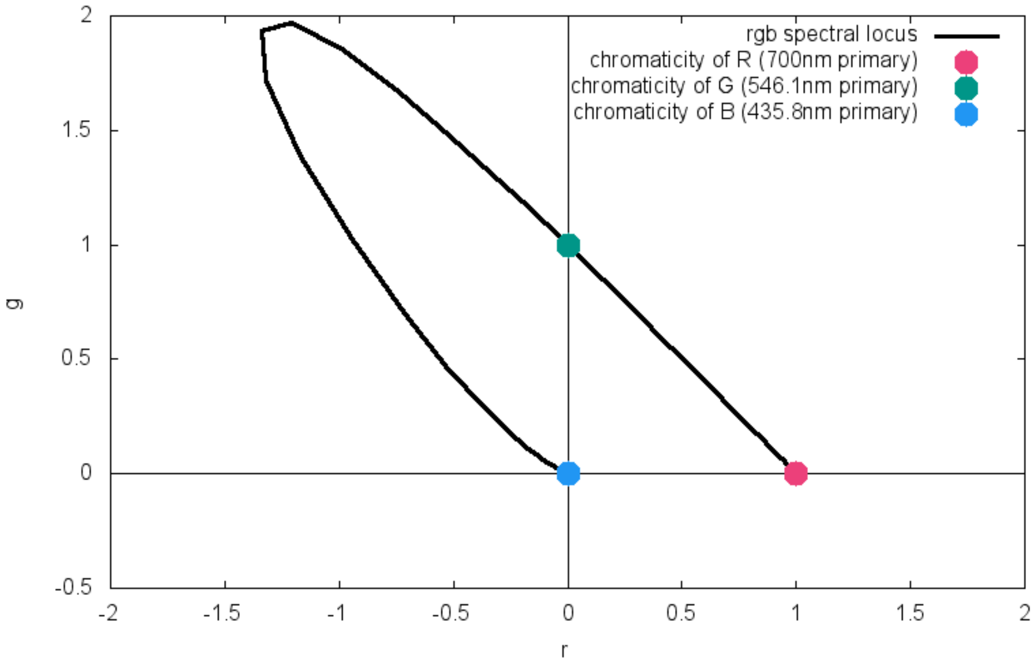



Dropping the 3D plot we end up with our 2D projection of the rg chromaticity space, our first chromaticity diagram!

Let's break down what we're looking at

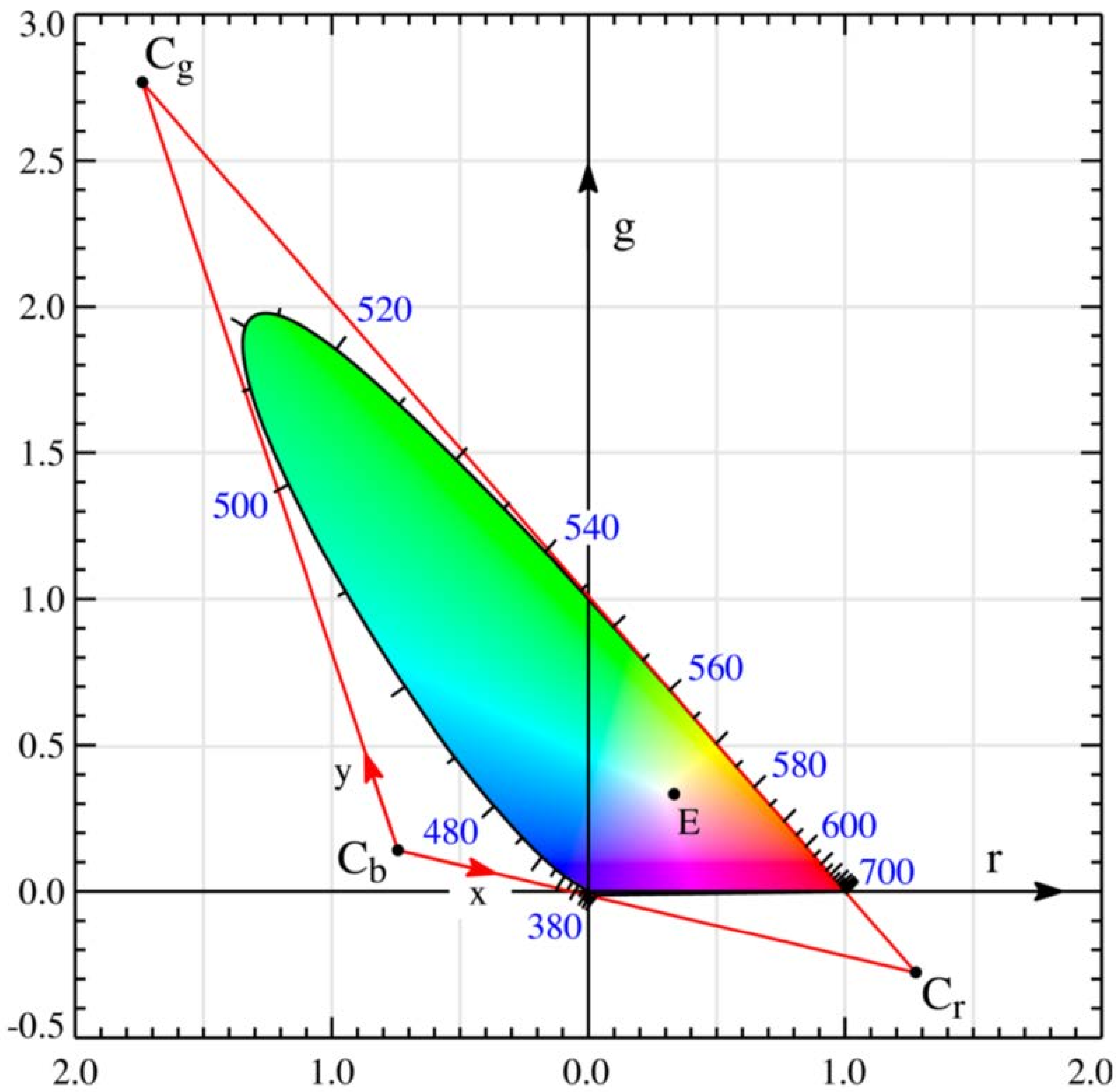
- Every point on the outer curve is the chromaticity coordinate of a spectral color.
- Every point inside the curve is a chromaticity of a non-spectral color.
- Every point outside the curve is an imaginary chromaticity that is meaningless and has no realizable color.
- Our original RGB primaries, are spectral colors, so they fall on the curve.

- Every point inside the triangle formed by the primaries is a chromaticity that can be created with those lights.
- Points inside the curve but not the triangle are real chromaticities but you would need different primary lights to create them.




My plotting abilities are not quite up to the task of rendering a colored chromaticity diagram so here is the rg diagram from wikipedia, ignore the red Cr, Cg and Cb triangle for now. When looking at chromaticity renderings remember that real colors have luminosity, chromaticities do not.

This diagram shows colors that have the appropriate chromaticities rendered with a visually interesting luminosity.

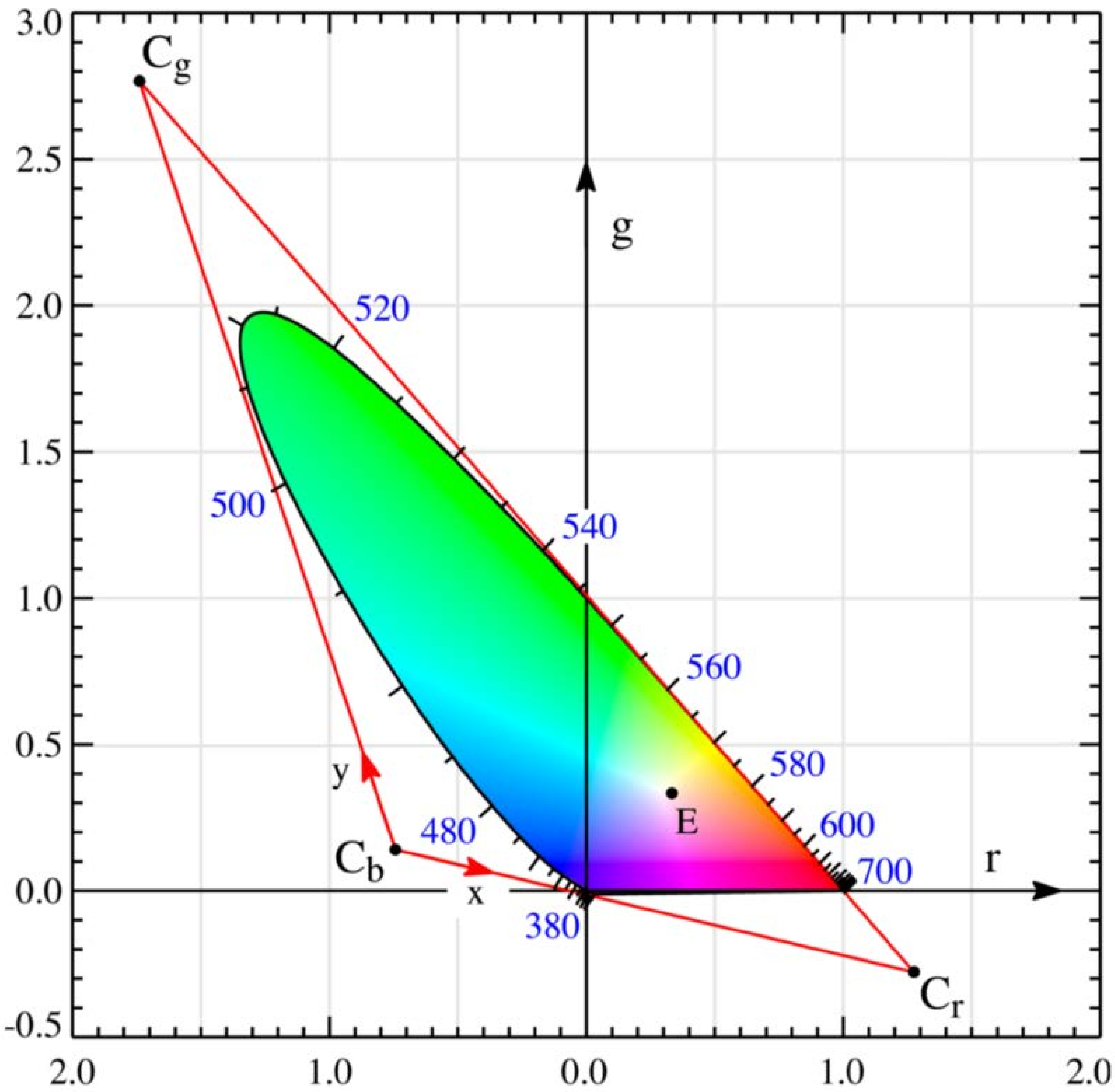



**G) Non Spectral Colors**

We know the chromaticities of the spectrum, so it makes sense that we could color the boundary of the curve, but what does it mean to color the inside of the curve?

The space inside the curve turns out to represent non spectral chromaticities, chromaticities that only can be created from mixtures of light. If you draw a line between any two points on the curve, the midpoint is the chromaticities of mixing those two spectral chromaticities.

The spectral colors are featured prominently in colorimetry as the spectral locus but you almost never run into these colors them in the real world. Narrow band spectral colors can be generated with lasers but the colors of every day objects are wide broadband spectral power distributions that well inside the boundary of the chromaticity diagram, like the spinach leaf at the start of this post.




**H) CIE XYZ Color Space**

The 1931 XYZ Color Matching Functions are a linear transformation of the 1931 RGB Color Matching Functions in order to give them some mathematically convenient properties. In fact, the RGB CMFs we've been discussing are mostly unused in practical colorimetry. Modern color spaces are defined in terms of XYZ.

Keep in mind that XYZ and RGB are different transformations of the same data, anything you can do with one you can do with the other, and colors can be converted back and forth between them.

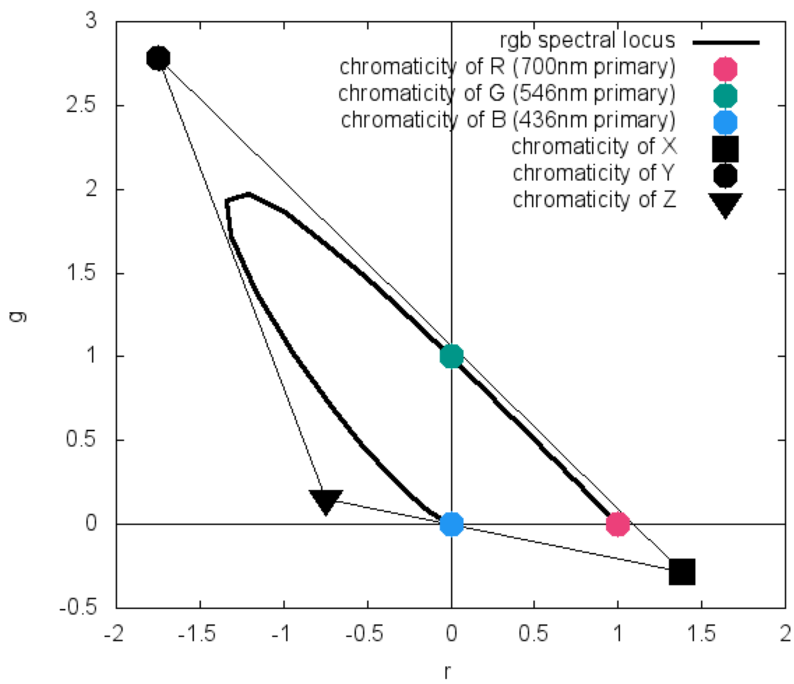
The big idea of XYZ is that one of the three functions could be transformed so that it aligned very closely to 1924 Luminosity Function, V(λ). This would mean that luminosity of a color could be determined entirely from looking at one of that color's primary values. Additionally, before computers the manual math involved in using the CMFs was difficult and there was a desire to have CMFs that didn't contain any negative values.

There is a slight problem however, in order to create an all positive transformation of the RGB CMFs you have to use imaginary primaries that don't correspond to any real color.

You can see this in the rg chromaticity diagram from the previous section, the only way to form a triangle around all of the real colors is to choose primary coordinates outside of the real colors. This is exactly what XYZ does.

Three new primaries, X Y and Z were picked such that when the RGB data was transformed to have XYZ as the new axes the Y primary mapped to the V curve and all values were positive.

Before, in rg chromaticity space:




Matrix coefficents that convert the RGB space into one where the carefully chosen XYZ primaries exist at coordinates (1,0) (0,1) and (0,0) in the new chromaticity space.

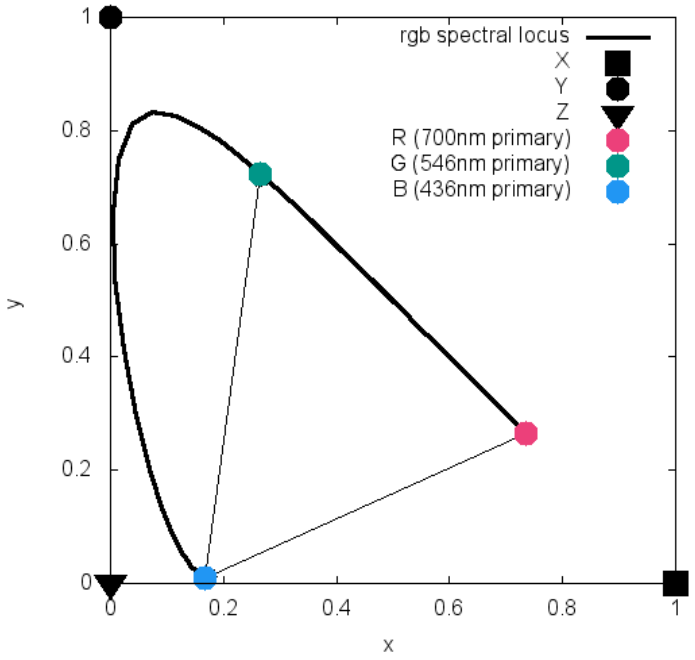
[X]   | 2.768 1.751 1.130|   [R]

[Y] = | 1.000 4.590 0.060| * [G]

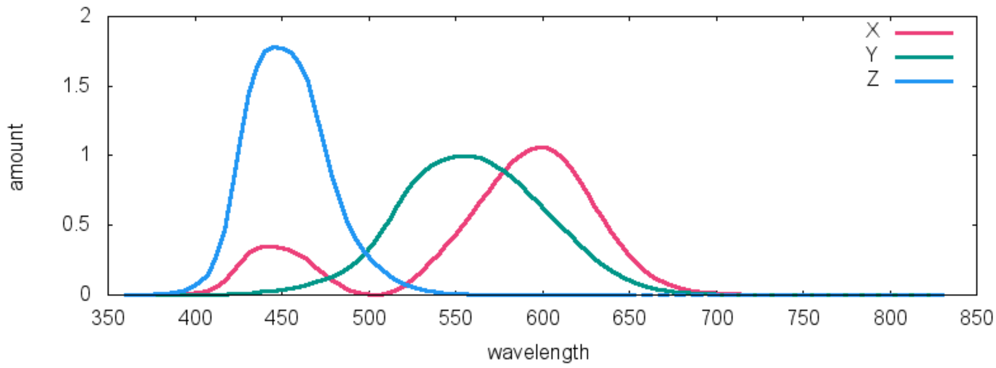
[Z]   | 0     0.056 5.594|   [B]

~ waves linear algebra wand ~

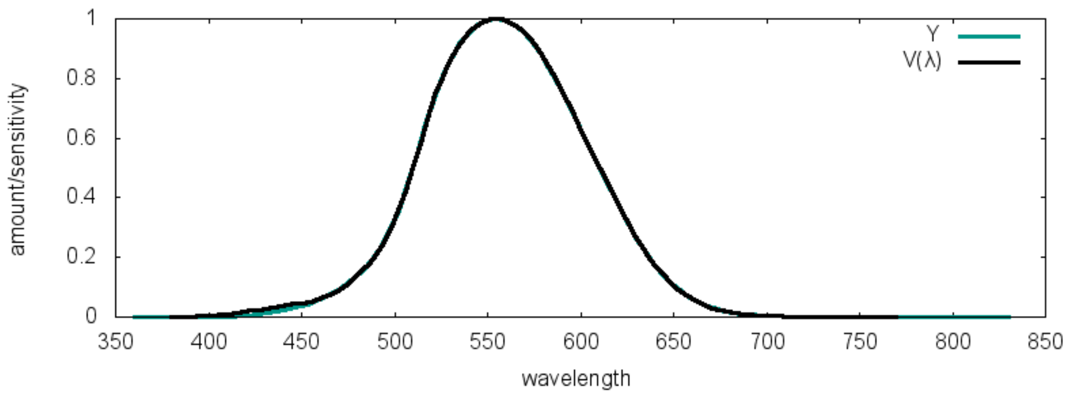
After, in xy chromaticity space. Look, all positive values!




Similarly, our old friend the RGB Color Matching Functions becomes the all positive XYZ Color Matching Functions:




Plotting the Y CMF against the luminosity from earlier shows the near perfect alignment.




The spectral locus still defines the coordinates of every spectral color, only now it's in terms of primaries that don't exist.

Coordinates specified in terms of imaginary colors might not seem very useful, but when it comes time to create physical colors you will basically do another linear transformation to convert the coordinates into your desired color space. If your XYZ color endd up having all positive values in the new color space, and if your new color space has non-imaginary primaries, then you have a recipe for real color creation. This is the subject of this 1996 blogpost by the founder of Autodesk, John Walker

There are probably thousands of color spaces specific to individual digital displays, but each one hypothetically only needs to know two. They need to know about XYZ and they need to know how to convert XYZ into their supported color space. XYZ is the common language used to communicate color.

**I) Real World Color Spaces**

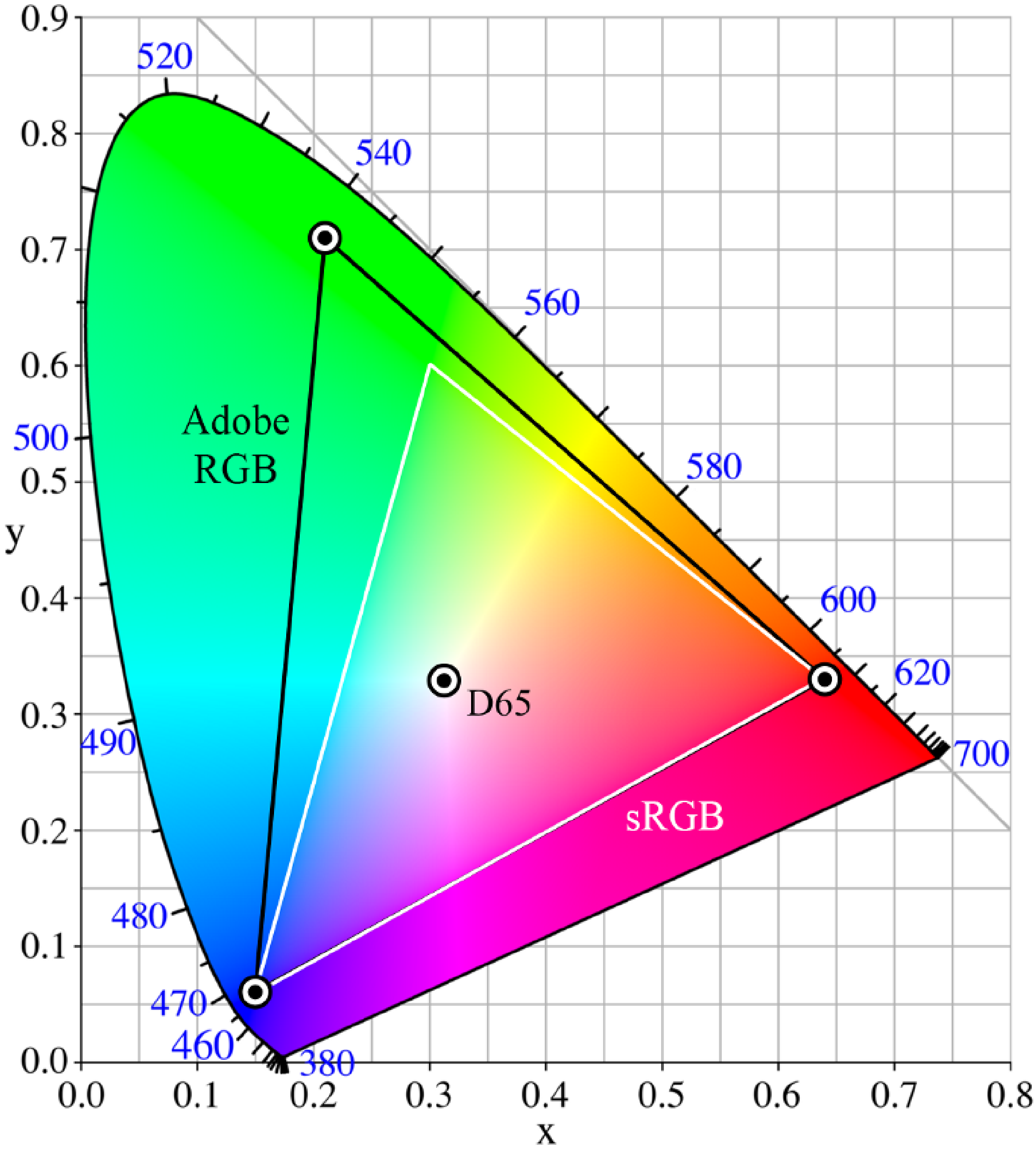
The release of the 1931 XYZ CMFs kick started the practical applications of accurate color reproduction in science and media.

The first request that I received to specify a colour on the CIE system was from our Ministry of Agriculture and Fisheries, who wished to define the colour of forced rhubarb in connection with a National Mark scheme they were introducing. This showed surprising enterprise on the part of our civil servants and it also provoked an amusing article in our humorous mazagine, Punch.

If you have ever dealt with colors in web development, you've probably seen colors specified as rgb values like 'rgb(31,157,167)'. These aren't referring to the 1931 RGB color space. They mostly commonly refer to the sRGB color space, created by Microsoft in 1996.

sRGB, and another popular web color space Adobe RGB are each defined by their own three primary colors. Those primaries are defined by specifying their XYZ values.

You can plot the primaries of other color spaces onto the XYZ chromaticity diagram to see the triangles they form. Every chromaticity inside the triangle is a chromaticity that can be specified in this that color space. The area inside a primary triangle is known as a color gamut.




You can see that neither sRGB or Adobe RGB can express the entire gamut of human color perception. If you photograph something with a color outside the range of the color space used by your camera, it will get usually get clipped to the closest expressible color.

Armed with the knowledge of an sRGB color, like rgb(184, 44, 161). You know how much you would need to vary the three sRGB primaries to create that color. If you had a digital display whose sub pixels were exactly the color of sRGB primaries you would be set! But that's usually not the case.

This is an array of 9 pixels, each made up of 3 sub pixels. The three colors of the sub pixels are the primary colors of this display.

If you have an sRGB value, you can do a linear transformation back into XYZ space.

Then you can do another linear transformation to make the value in terms of the primaries of the display device.

Now you can display that color in real life, by varying the intensities of the sub pixels of the display accordingly.

**J) Beyond Color Matching**

As we mentioned in the introduction, Color Matching deals with the accurate reproducibility of color, it tells us if one color looks like another but not what they look like. In fact colors that match can retain that match even as their appearance changes across different lighting conditions.

Color Appearance Models seek to solve the much more difficult problem of quantifying colors in dimensions that are meaningful and intuitive to humans. Dimensions like, hue, saturation and brightness.

Another advanced topic is Color Difference Models, quantifying how different two colors seem from each other so that you can create colors spaces with perceptual uniformity. In these, the distance between any two colors in the coordinate space mirrors the perceived visual difference in the colors.

Stability of matching also means that a color specification cannot take account of the changes in appearance of a color introduced by adaptation, and it is for this reason that we must regard a tristimulus specification as a measure of the quality of the stimulus rather than of the quality of the sensation.

This may be regarded as a limitation of colour measurement and it must be admitted that we are at present a long way from establishing a system on which colour appearance can be specified exactly, yet experience shows that colorimetry as at present practised has many applications and is, indeed, an esssential state towards the specficiation of colour apperance.

#### *2.1.7.1. CIELUV*

This is based on CIE Yu’v’ (1976) and is a further attempt to linearise the perceptibility of unit vector colour differences. It is a non-linear colour space, but the conversions are reversible. Colouring information is centered on the colour of the white point of the system, subscript n, (D65 in most TV systems). The non-linear relationship for Y* is intended to mimic the logarithmic response of the eye.

$$L^* = \begin{cases} 116(\frac{Y}{Y_n})^{\frac{1}{3}} - 16 & \text{if } \frac{Y}{Yn} > 0.008856 \\ 903.3(\frac{Y}{Y_n}) & \text{if } \frac{Y}{Yn} \leq 0.008856 \end{cases}$$

$$u^* = 13(L^*)(u' - u'_n)$$
$$v^* = 13(L^*)(v' - v'_n)$$

L* scales from 0 to 100 for relative luminance (Y/Yn) scaling 0 to 1.

There are three other, more meaningful polar parameters which more closely match the human visual experience than do the Cartesian parameters. Chroma C*, Hue $h_{uv}$, and psychometric saturation $s_{uv}$.

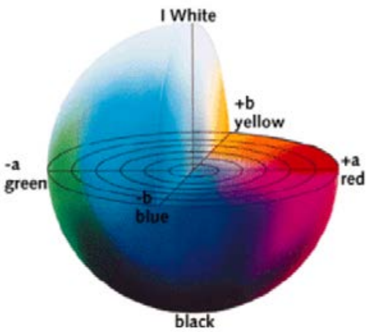
$$\begin{aligned} C^* &= (u^{*2} + v^{*2})^{0.5} \\ h_{uv} &= arctan(\frac{v^*}{u^*}) \\ s_{uv} &= \frac{C^*}{L^*} \end{aligned}$$

#### 2.1.7.2. ***CIELAB***

Color space defined by the CIE, based on one channel for Luminance (lightness) (**L**) and two color channels (**a** and **b**).

One problem with the XYZ color system, is that colorimetric distances between the individual colors do not correspond to perceived color differences. For example, in the figure above, a difference between green and greenish-yellow is relatively large, whereas the distance distinguishing blue and red is quite small. The CIE solved this problem in 1976 with the development of the three-dimensional Lab color space (or CIELAB color space).

In this model, the color differences which you perceive correspond to distances when measured colorimetrically. The *a* axis extends from green (-a) to red (+a) and the *b* axis from blue (-b) to yellow (+b). The *brightness* (L) increases from the bottom to the top of the three-dimensional model.



This color space is better suited to many digital image manipulations than the RGB space, which is typically used in image editing programs. For example, the Lab space is useful for sharpening images and the removing artifacts in JPEG images or in images from digital cameras and scanners.