Gernot Hoffmann

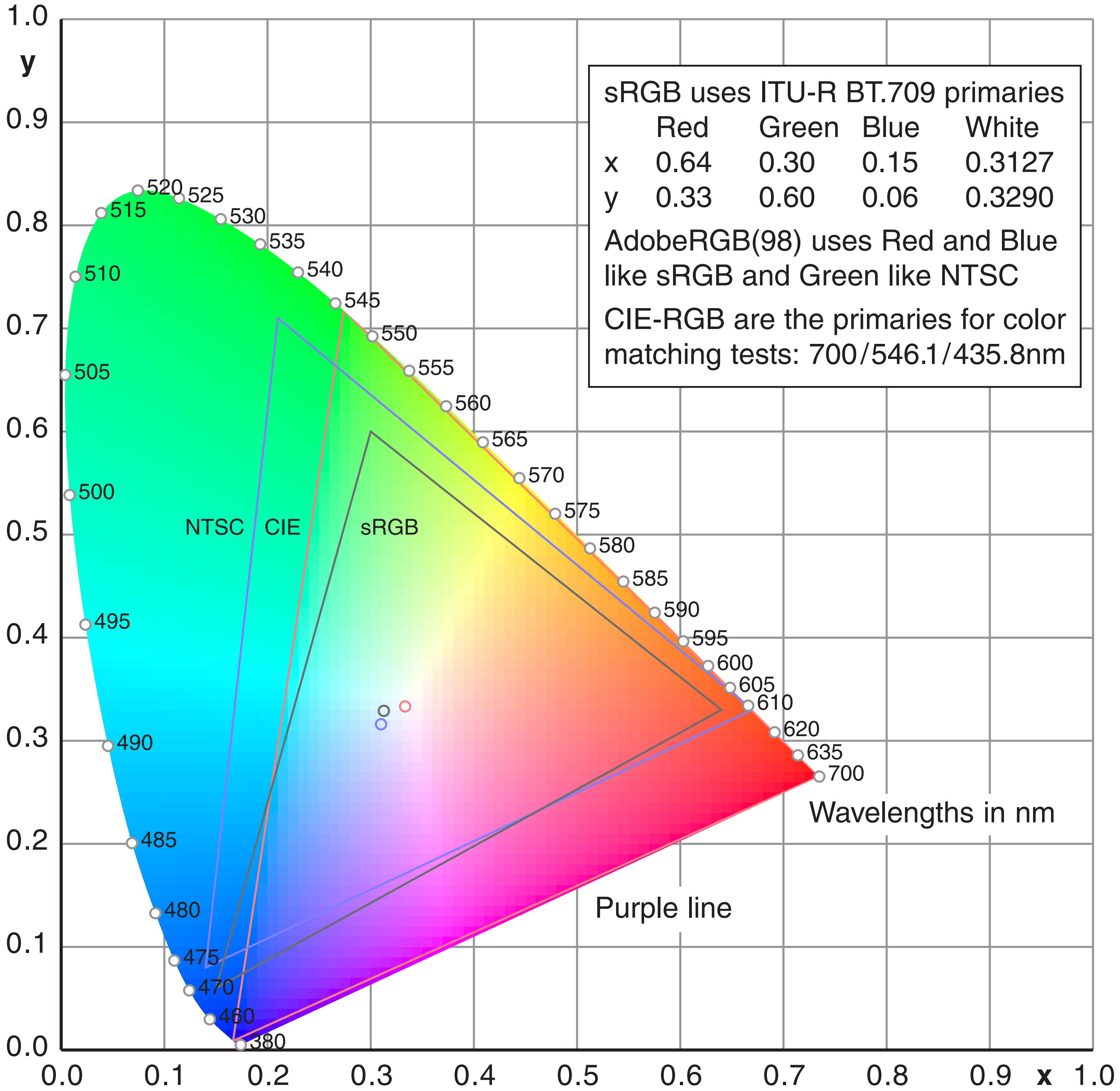
CIE Color Space

Contents

1. CIE Chromaticity Diagram (1931)

The threedimensional color space CIE XYZ is the basis for all color management systems. This color space contains all perceivable colors - the human gamut. Many of them cannot be shown on monitors or printed.

The twodimensional CIE chromaticity diagram xyY (below) shows a special projection of the threedimensional CIE color space XYZ.
Some interpretations are possible in xyY, others require the threedimensional space XYZ or the related threedimensional space CIELab.

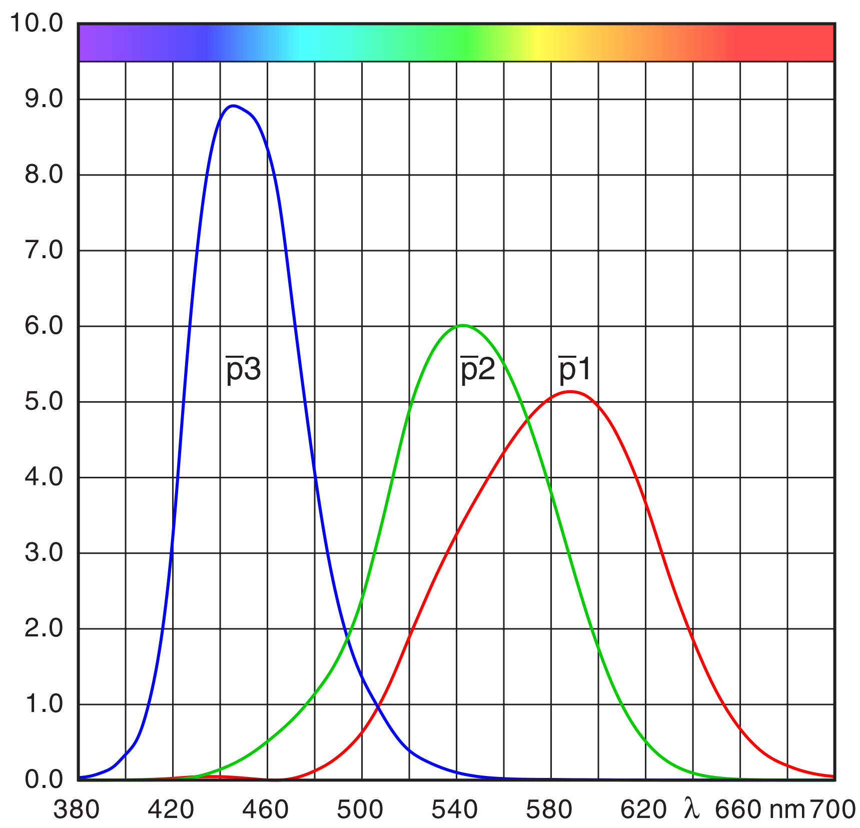


2. Color Perception by Eye and Brain

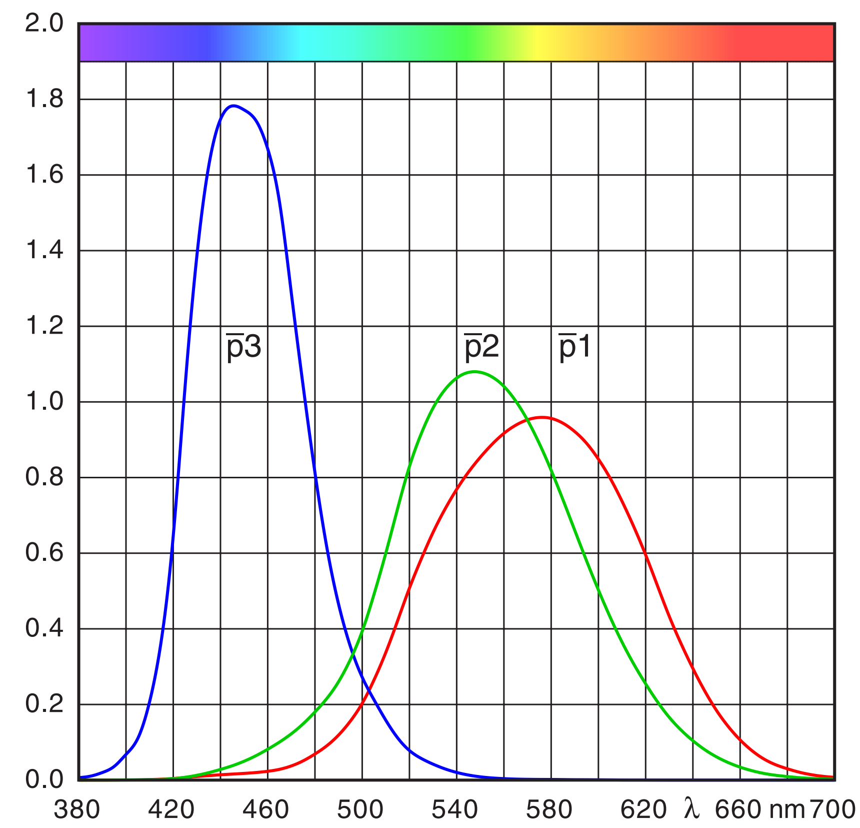
The retina contains two groups of sensors, the rods and the cones. In each eye are about 100 millions of rods responsible for the luminance. About 6 millions of cones measure color. The sensors are already 'wired' in the retina - only 1 million nerve fibres carry the information to the brain.The perception of colors by cones requires an absolute luminance of at least some cd/m^2 (candela per squaremeter). A monitor delivers about 100 cd/m^2 for white and 1 cd/m^2 for black.

Three types of cones (together with the rods) form a tristimulus measuring system. Spectral information is lost and only three color informations are left. We may call these colors blue, green and red but the red sensor is in fact an orange sensor.
The optical system is not color corrected. It would be impossible to focus simultaneously for three different wavelengths. The overlapping sensitivities of the green and the red sensor may indicate that the focussing happens mainly in the overlapping range whereas blue is generally out of focus. This sounds strange, but the gap for image parts on the blind spot is corrected as well - another example for the surprising features of eye and brain.

These diagrams show two of several models for the cone sensitivities. These and similar functions cannot be measured directly - they are mathematical interpretations of color matching experiments.
The sensitivity between 700nm and 800nm is very low, therefore all the diagrams are drawn for the range 380nm to 700nm.



Cone sensitivities [3]



Cone sensitivities [1]

3. RGB Color-Matching

The color matching experiment was invented by Hermann Graßmann (1809 - 1877) about 1853.

Three lamps with spectral distributions ***R***,***G***,***B*** and weight factors R,G,B = 0..100 generate the color impression ***C*** = R***R*** + G***G*** + B***B***.

The three lamps must have linearly independent spectra, without any other special specification.
A fourth lamp generates the color impression ***D***.

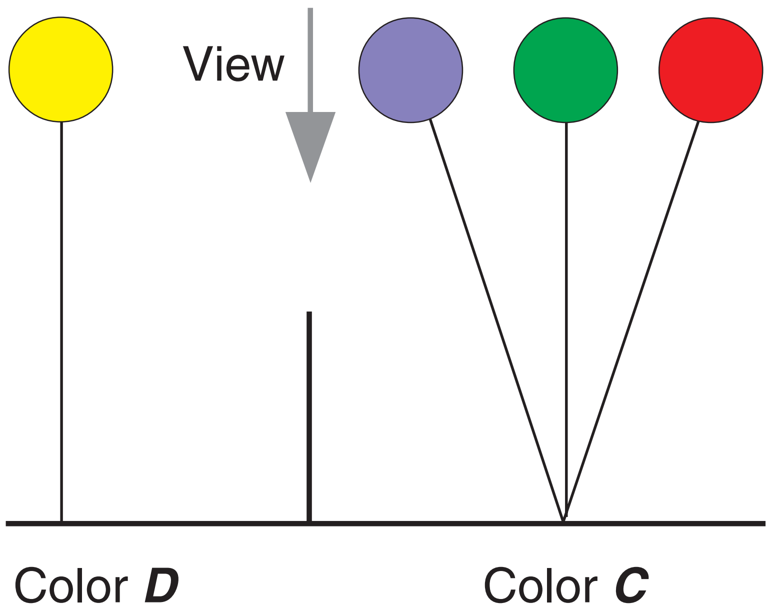
Can we match the color impressions ***C*** and ***D*** by adjusting R,G,B ? In many cases we can:

BlueGreen = 7***R*** + 33***G*** + 39***B***

In other cases we have to move one of the three lamps to the left side and match indirectly:

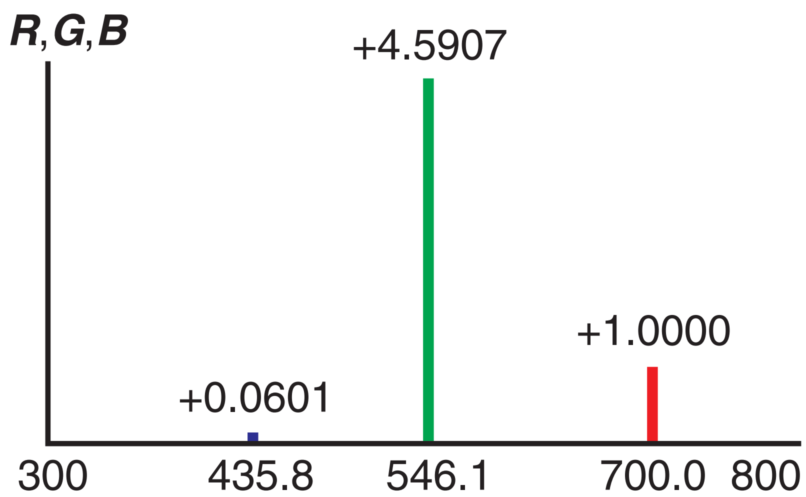
Vibrant BlueGreen + 38***R*** = 42***G*** + 91***B***
Vibrant BlueGreen = - 38***R*** + 42***G*** + 91***B***

This is the introduction of 'negative' colors. The equal sign means 'matched by'. It is generally possible to match a color by three weight factors, but one or even two can be negative (only one for CIE-RGB) .
Data for the example are shown in Appendix A.



Color matching experiment

The CIE Standard Primaries (1931) are narrow band light sources (monochromats, line spectra or delta functions) ***R*** (700 nm), ***G*** (546.1nm) and ***B*** (435.8 nm). They replace the red, green and blue lamps in the drawing above. In fact these sources were actually *not* used - all results were *calculated* for these primaries after tests with other sources.



CIE Standard Primaries

The normalized weight factors are called CIE Color-Matching Functions $\bar{r}(\lambda), \bar{g}(\lambda), \bar{b}(\lambda)$.

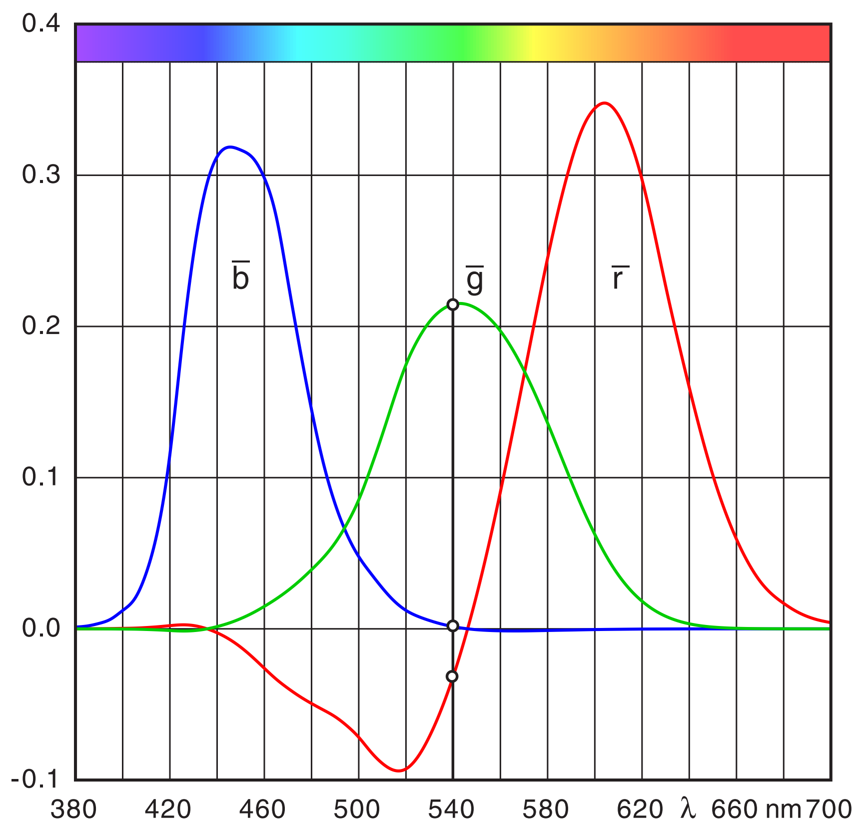
The diagram shows for example the three values for matching a spectral pure color (monochromat) with wavelength λ=540nm. This requires a negative value for red.

RGB colors for a spectrum P(λ) are calculated by these integrals in the range from 380nm to 700nm or 800nm:

$$R = k \int P(\lambda)\, \bar{r}(\lambda)\, d\lambda$$

$$G = k \int P(\lambda)\, \bar{g}(\lambda)\, d\lambda$$

$$B = k \int P(\lambda)\, \bar{b}(\lambda)\, d\lambda$$



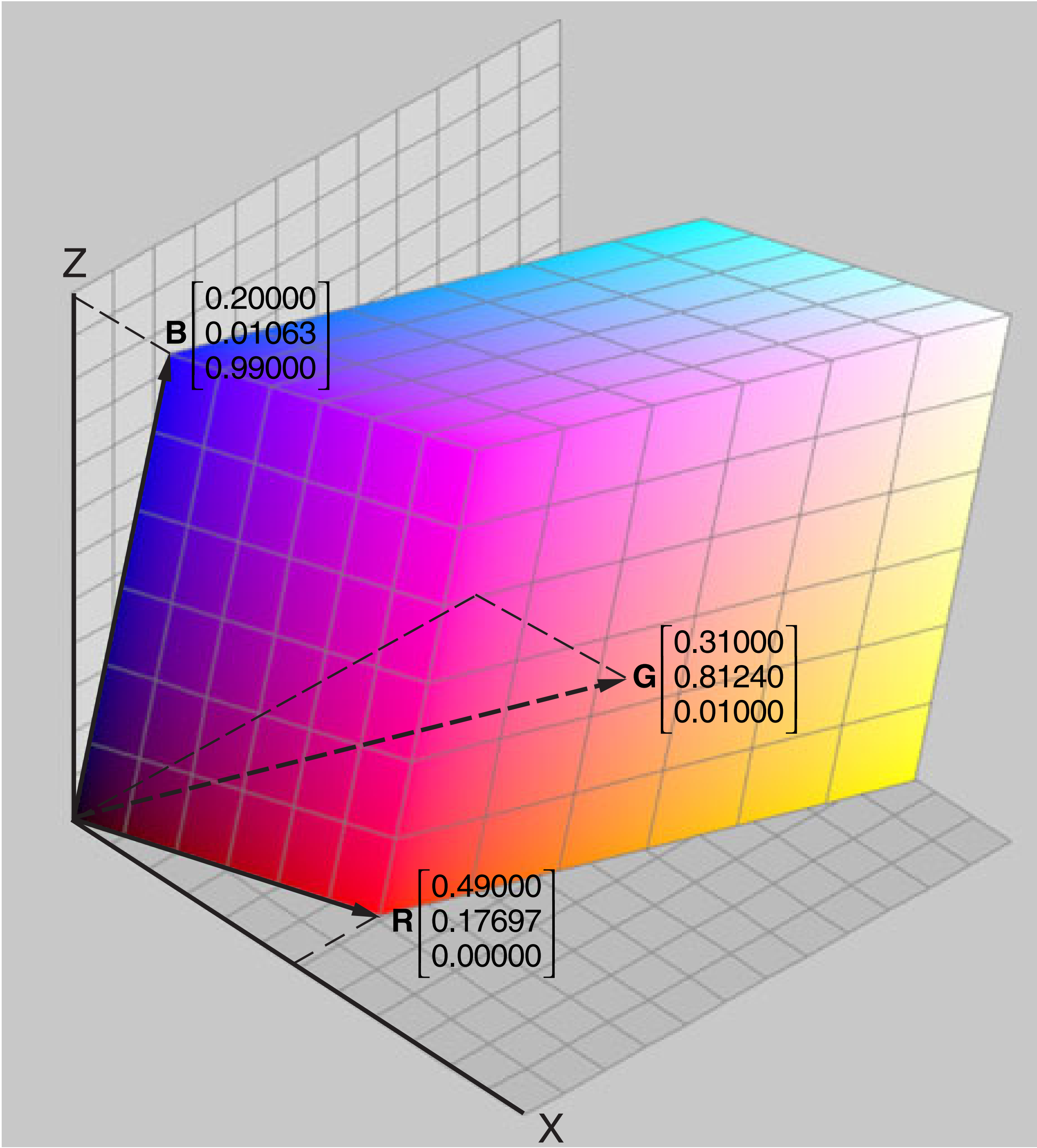
RGB Color-matching functions

4. XYZ Coordinates

In order to avoid negative RGB numbers the CIE consortium had introduced a new coordinate system XYZ.
The RGB system is essentially defined by three non-orthogonal base vectors in XYZ.

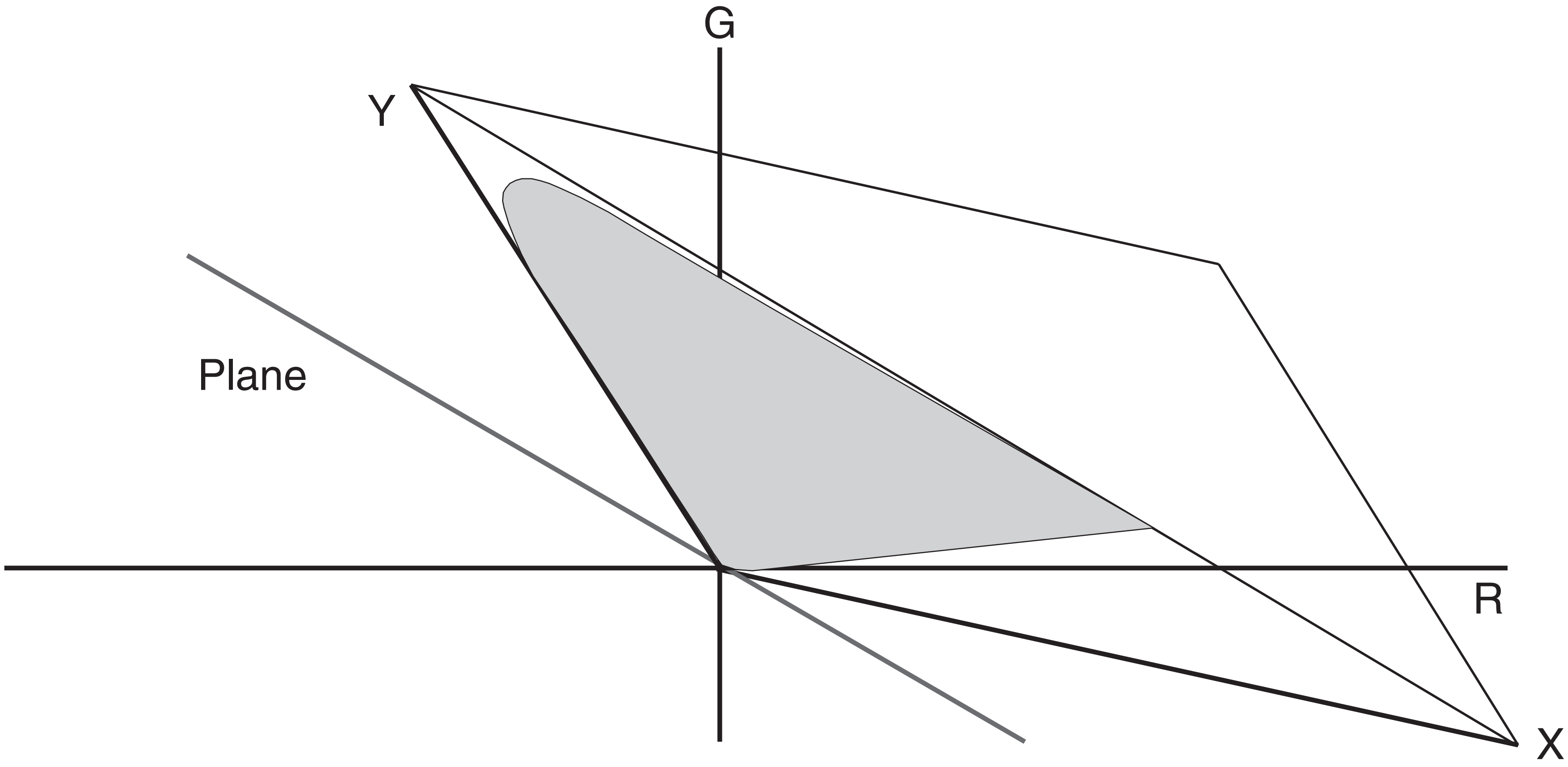
The bottom image explains the sitution for 2D coordinates R,G and X,Y a little simplified.
The shaded area shows the human gamut. A plane divides the space in two half spaces.
The new coordinates X,Y are chosen so that the gamut is entirely accessible for positive values.
This can be generalized for the 3D space.

In the upper image the axes XYZ are drawn orthogonally, in the lower image the axes RGB.



RGB base vectors and color cube in XYZ

The coordinates of the base vectors in XYZ (coordinates of the primaries as shown above) for any RGB system are found as columns of the matrix C_{xr} in chapter 11.



2D visualization for RG and XY

5. XYZ Primaries

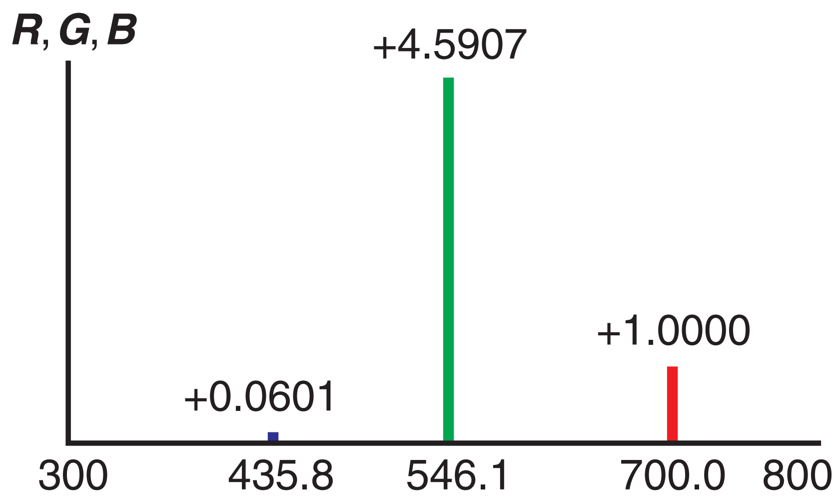
The coordinate systems XYZ and RGB are related to each other by linear equations.

$$\mathbf{X} = \mathbf{C}_{xr}\,\mathbf{R}$$

$$\begin{aligned} X &= +0.49000\,R + 0.31000\,G + 0.20000\,B \\ Y &= +0.17697\,R + 0.81240\,G + 0.01063\,B \qquad (1) \\ Z &= +0.00000\,R + 0.01000\,G + 0.99000\,B \end{aligned}$$

$$\mathbf{R} = \mathbf{C}_{rx}\,\mathbf{X}$$

$$\begin{aligned} R &= +2.36461\,X - 0.89654\,Y - 0.46807\,Z \\ G &= -0.51517\,X + 1.42641\,Y + 0.08876\,Z \qquad (2) \\ B &= +0.00520\,X - 0.01441\,Y + 1.00920\,Z \end{aligned}$$



CIE primaries ***R*,*G*,*B***

Another view is possible by introducing synthetical or 'imaginary' primaries ***X***,***Y***,***Z***.

The Standard Primaries ***R***,***G***,***B*** are monochromatic stimuli. Mathematically they are *single* delta functions with well defined areas.
In the diagram the height represents the contribution to the luminance.
The ratios are 1.0 : 4.5907 : 0.0601.

The spectra ***X***,***Y***,***Z*** are calculated by the application of the matrix operation (2) and the scale factors.

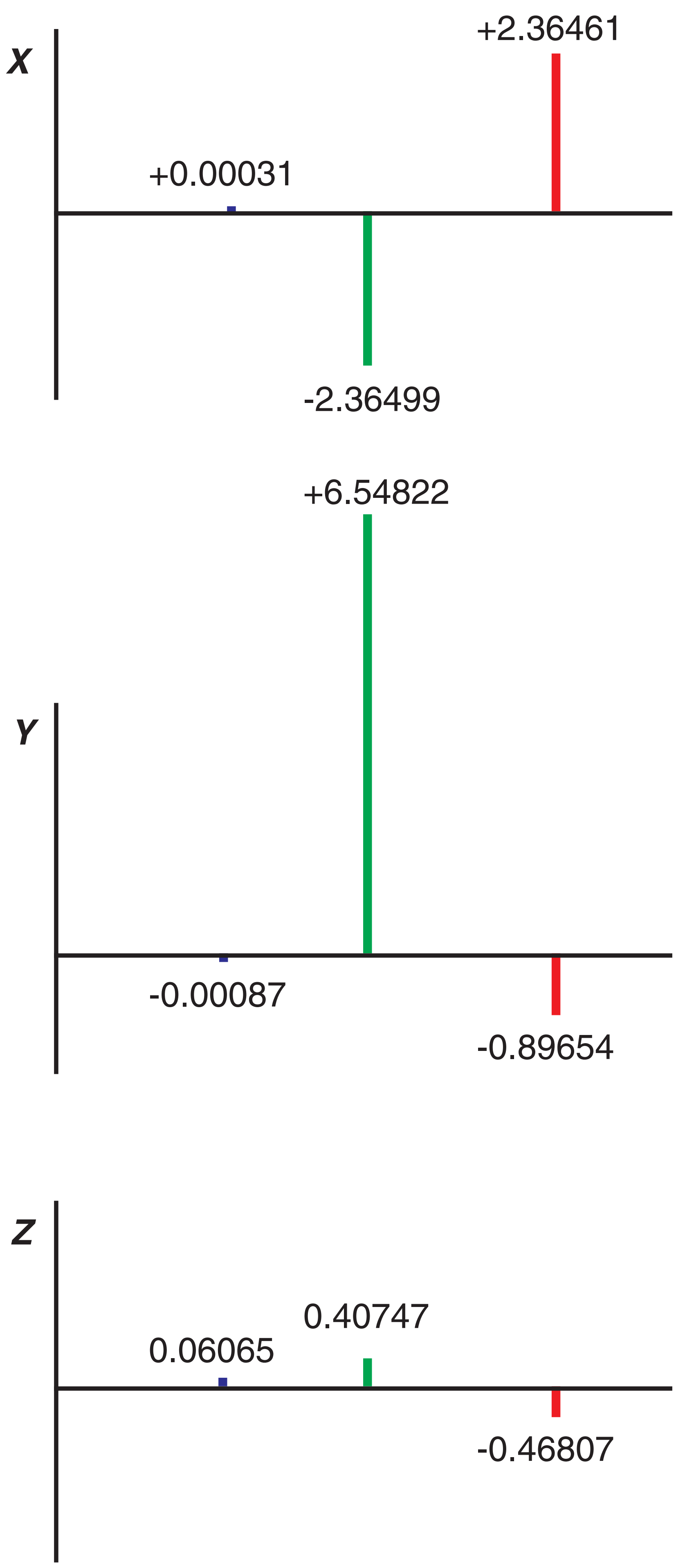
An example:

X=1, Y=0, Z=0 :

$$\begin{aligned} \boldsymbol{X} &= +2.36461 \cdot 1.0000\,\boldsymbol{R} \\ &\quad -0.51517 \cdot 4.5907\,\boldsymbol{G} \\ &\quad +0.00520 \cdot 0.0601\,\boldsymbol{B} \\ \boldsymbol{X} &= +2.36461\,\boldsymbol{R} - 2.36499\,\boldsymbol{G} + 0.00031\,\boldsymbol{B} \end{aligned}$$

The primaries ***X***,***Y***,***Z*** are *sums* of delta functions. ***X*** and ***Z*** do not contribute to the luminance. This is a special trick in the CIE system. The integrals are zero, here represented by the sum of the heights. The luminance is defined by ***Y*** only.

In color matching experiments negative values or weight factors R, G, B are allowed.
Some matchable colors cannot be generated by the Standard Primaries. Other light sources are necessary, especially spectral pure sources (monochromats).



Synthetical primaries ***X***, ***Y***, ***Z***

6. XYZ Color-Matching Functions

The new color-matching functions $\bar{x}(\lambda), \bar{y}(\lambda), \bar{z}(\lambda)$ have non-negative values, as expected. They are calculated from $\bar{r}(\lambda), \bar{g}(\lambda), \bar{b}(\lambda)$ by using the matrix $\mathbf{C}_{xr}$ in chapter 5.

The functions $\bar{x}(\lambda), \bar{y}(\lambda), \bar{z}(\lambda)$ can be understood as weight factors. For a spectral pure color ***C*** with a fixed wavelength λ read in the diagram the three values. Then the color can be mixed by the three Standard Primaries:

$$\boldsymbol{C} = \bar{x}(\lambda)\ \boldsymbol{X} + \bar{y}(\lambda)\ \boldsymbol{Y} + \bar{z}(\lambda)\ \boldsymbol{Z}$$

Generally we write

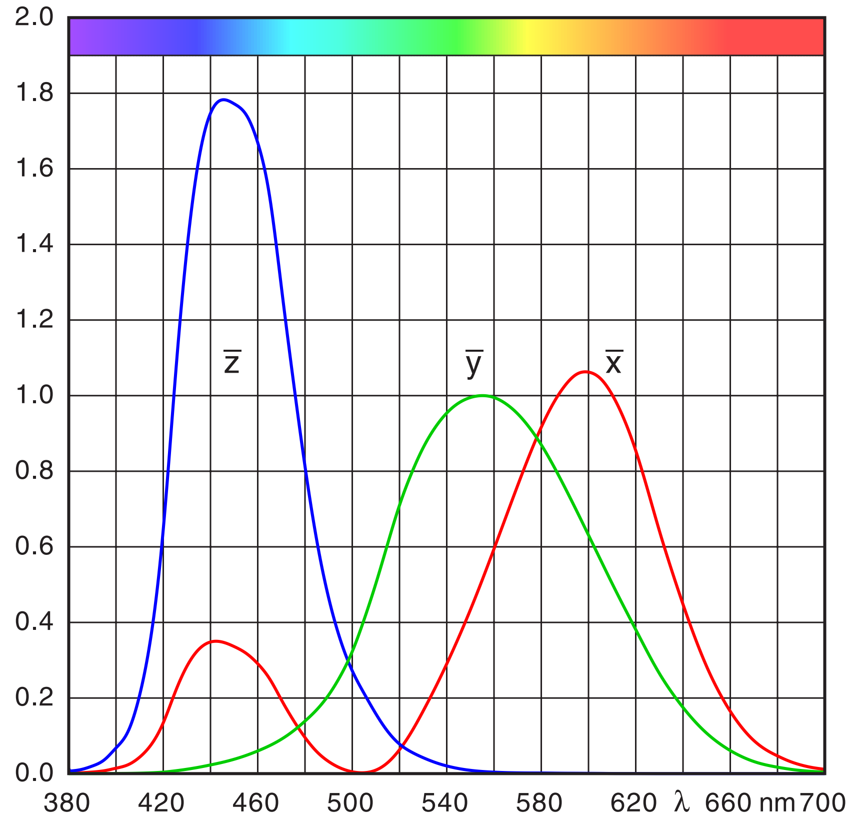
$$\boldsymbol{C} = X\ \boldsymbol{X} + Y\ \boldsymbol{Y} + Z\ \boldsymbol{Z}$$

and a given spectral color distribution $P(\lambda)$ delivers the three coordinates XYZ by these integrals in the range from 380nm to 700nm or 800nm:

$$X = k\int P(\lambda)\,\bar{x}(\lambda)\,d\lambda$$

$$Y = k\int P(\lambda)\,\bar{y}(\lambda)\,d\lambda$$

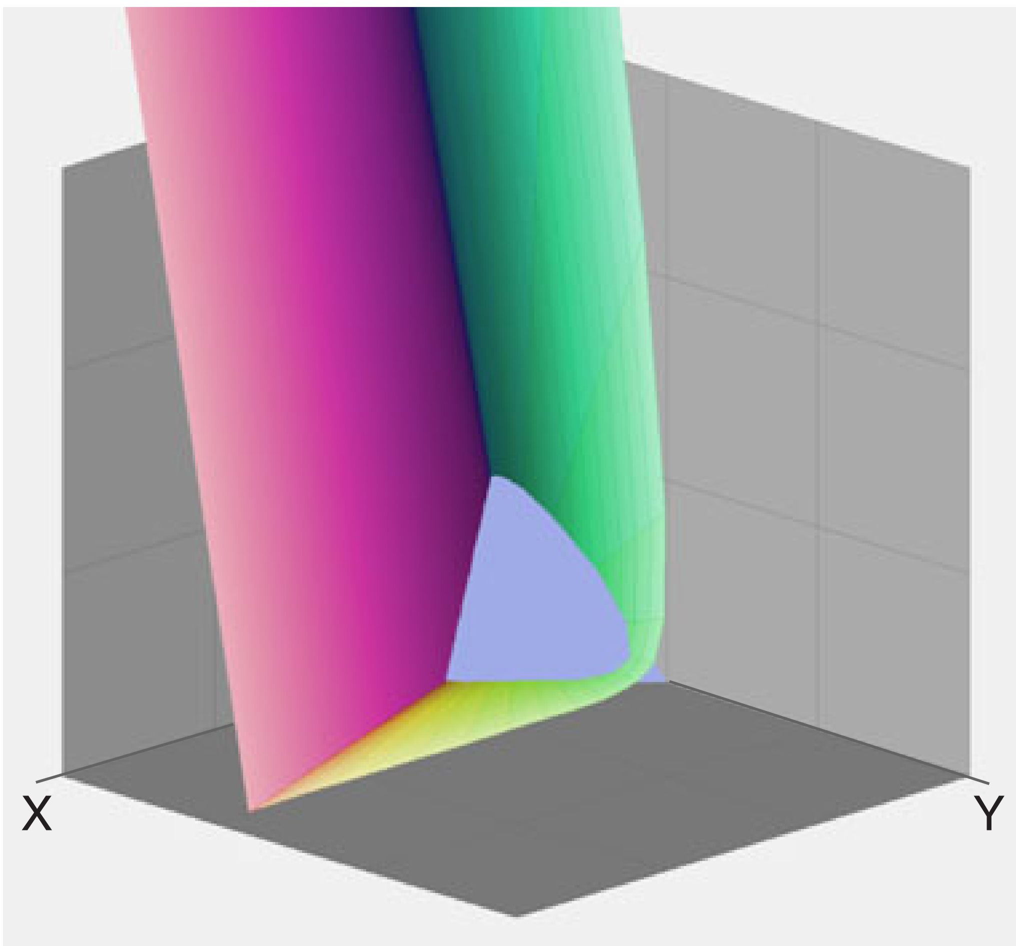
$$Z = k\int P(\lambda)\,\bar{z}(\lambda)\,d\lambda$$



XYZ Color-matching functions

Mostly, the arbitrary factor k is chosen for a normalized value Y=1 or Y=100. Matrix operations are always normalized for R,G,B,Y=0 to 1.

This diagram shows already the human gamut in XYZ. It is an irregularly shaped cone.The intersection with the blue-ish colored plane in the corner will deliver the chromaticity diagram.



Human gamut in XYZ

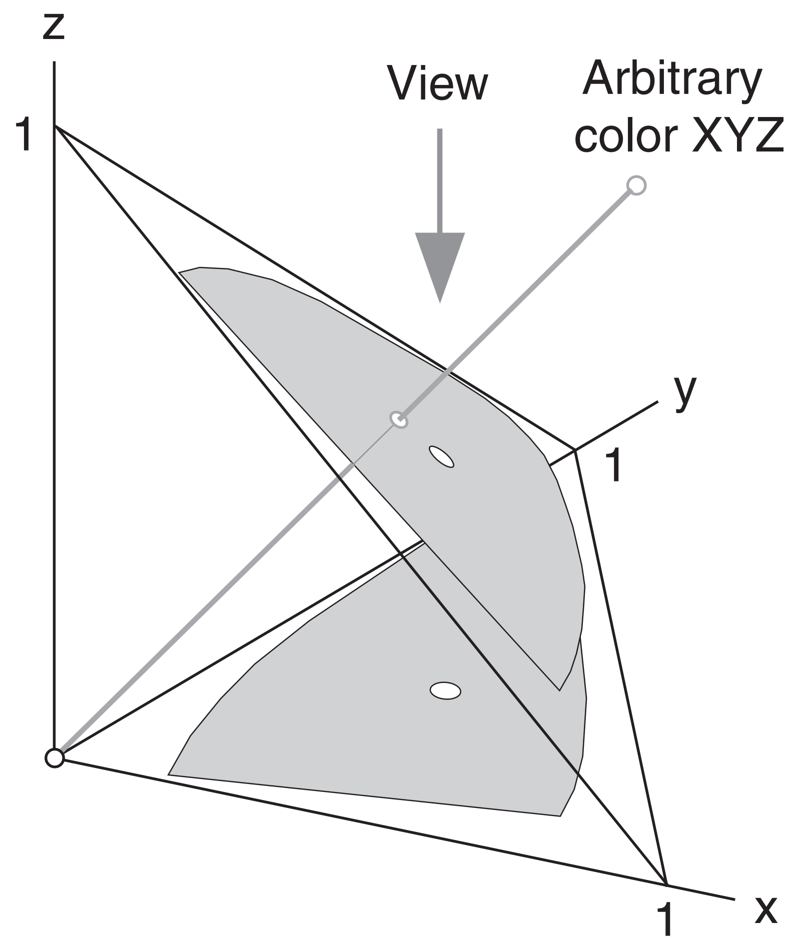
7. Chromaticity Values

The chromaticity values x,y,z depend only on the hue or dominant wavelength and the saturation. They are independend of the luminance:

$$x = \frac{X}{X+Y+Z}$$

$$y = \frac{Y}{X+Y+Z}$$

$$z = \frac{Z}{X+Y+Z}$$



Projection and chromaticity plane

Obviously we have $x + y + z = 1$. All the values are on the triangle plane, projected by a line through the arbitrary color XYZ and the origin, if we draw XYZ and xyz in one diagram.
This is a planar projection. The center of projection is in the origin.

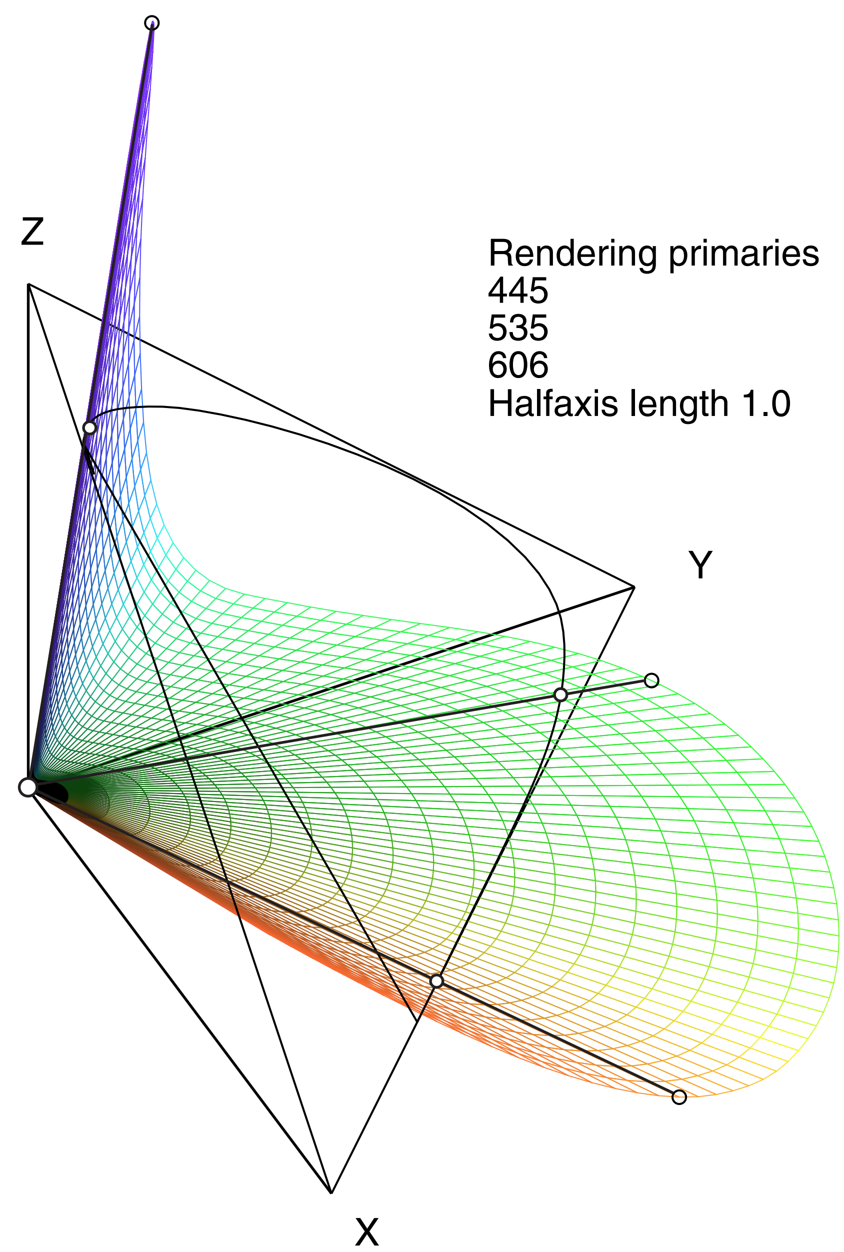
The vertical projection onto the xy-plane is the chromaticity diagram xyY (view direction).
To reconstruct a color triple XYZ from the chromaticity values xy we need an additional information, the luminance Y.

$$z = 1 - x - y$$

$$X = \frac{x}{y}Y$$

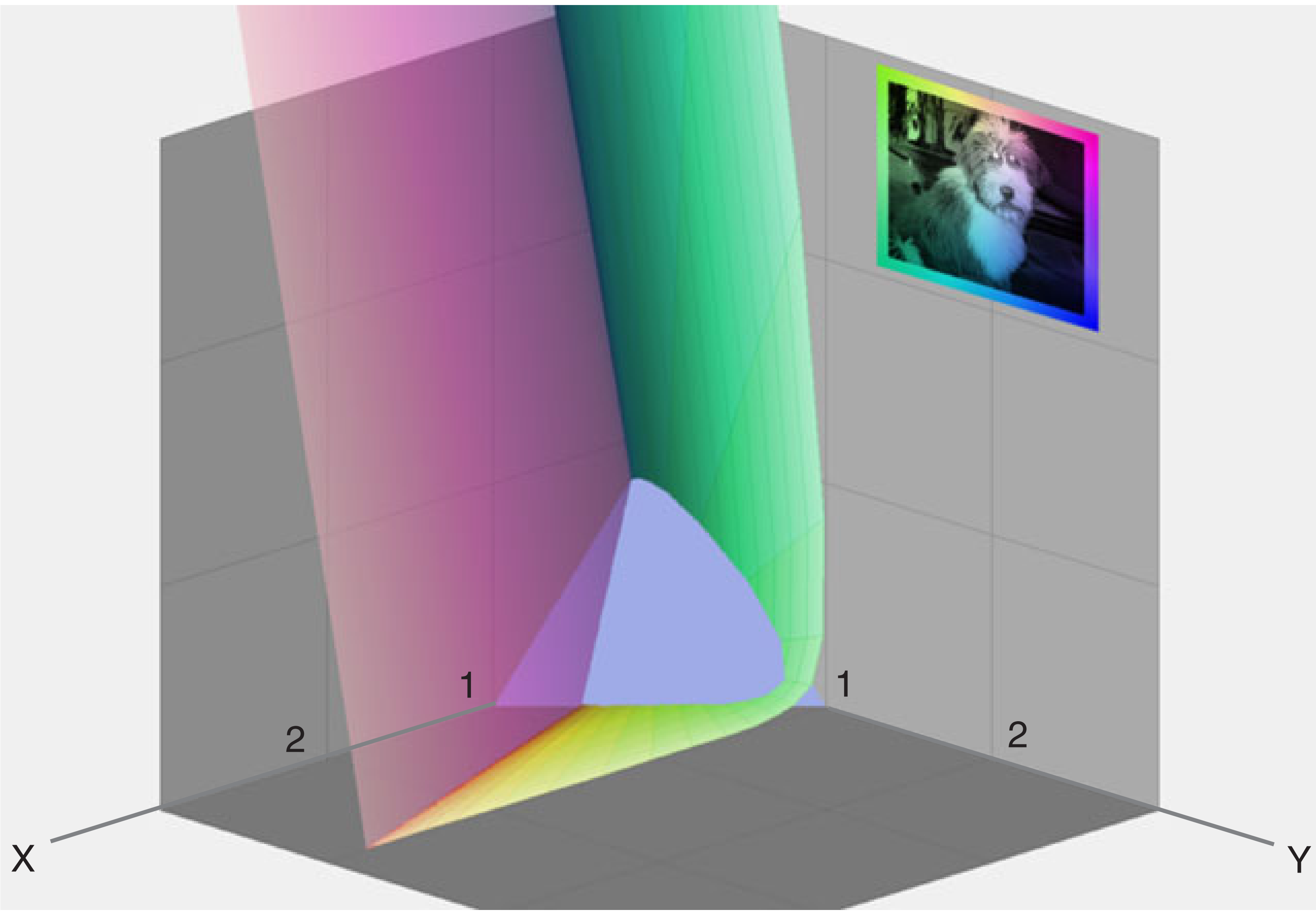
$$Z = \frac{z}{y}Y$$

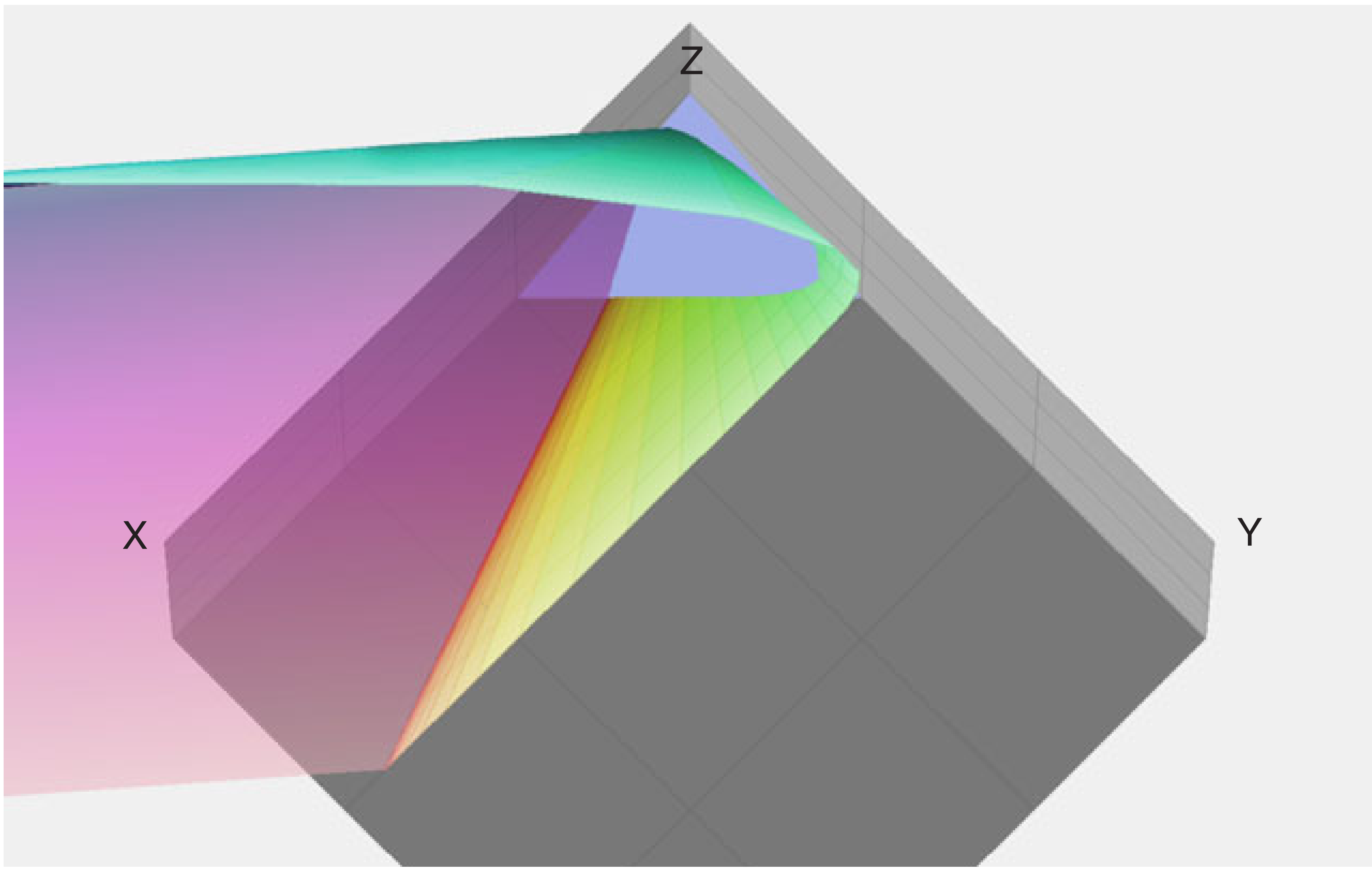
All visible (matchable) colors which differ only by luminance map to the same point in the chromaticity diagram. This is sometimes called 'horseshoe diagram' (page 2).
The right image shows a 3D view of the color-matching functions, connected by rays with the origin. The contour is here called 'locus of unit monochromats' [18]. For spectral colors this is the same as XYZ.
Then the contour is mapped onto the plane as above.
The spectral loci for blue and for red end nearly in the origin: colors with short and long wavelengths appear rather dark, they are almost invisible for a reasonably limited power.
The chromaticity diagram conceals this important fact. The purple line can be considered as a fake. Real purples are inside the horseshoe contour.



8. Color Space Visualization

These images are computer graphics. Accurate transformations and a few applications of image processing.The contour of the horseshoe is mapped to XYZ for luminances Y= 0..1 . The purple plane is shown transparent. All colors were selected for readabilty. The colors are not correct, this is anyway impossible. More important is here the geometry. The gamut volume is confined by the color surface (pure spectral colors), the purple plane and the plane Y= 1. The regions with small values Y appear extremely distorted - near to a singularity.
For blue very high values Z are necessary to match a color with specified luminance Y= 1.



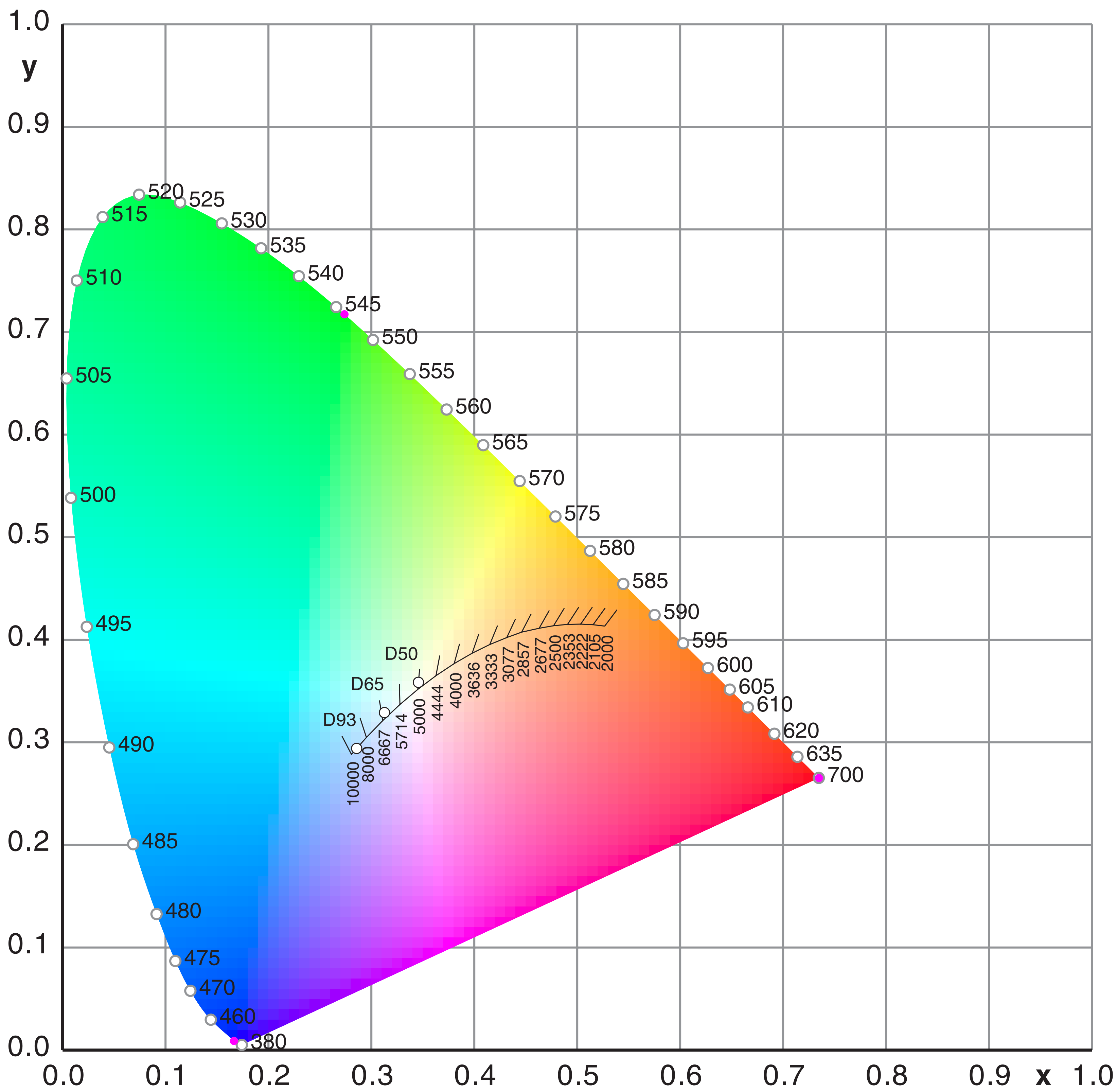


9. Color Temperature and White Points

The graphic shows the color temperature for the *Planck* radiator from 2000K to 10000K, the directions of correlated color temperatures and the white points for daylight D50 and D65. Uncalibrated monitors have about 9300K which is here simply called D93.

Data by [3]. EPS graphic available here [15].

```
  T/K     x        y      Dir y/x
 2000 0.52669 0.41331    1.33101
 2105 0.51541 0.41465    1.39021
 2222 0.50338 0.41525    1.45962
 2353 0.49059 0.41498    1.54240
 2500 0.47701 0.41368    1.64291
 2677 0.463   0.41121    1.76811 % error in table [3], estimated values
 2857 0.446   0.40742    1.92863
 3077 0.43156 0.40216    2.14300
 3333 0.41502 0.39535    2.44455
 3636 0.39792 0.38690    2.90309
 4000 0.38045 0.37676    3.68730
 4444 0.36276 0.36496    5.34398
 5000 0.34510 0.35162   11.17883
 5714 0.32775 0.33690  -39.34888
 6667 0.31101 0.32116   -6.18336
 8000 0.29518 0.30477   -3.08425
10000 0.28063 0.28828   -1.93507
```

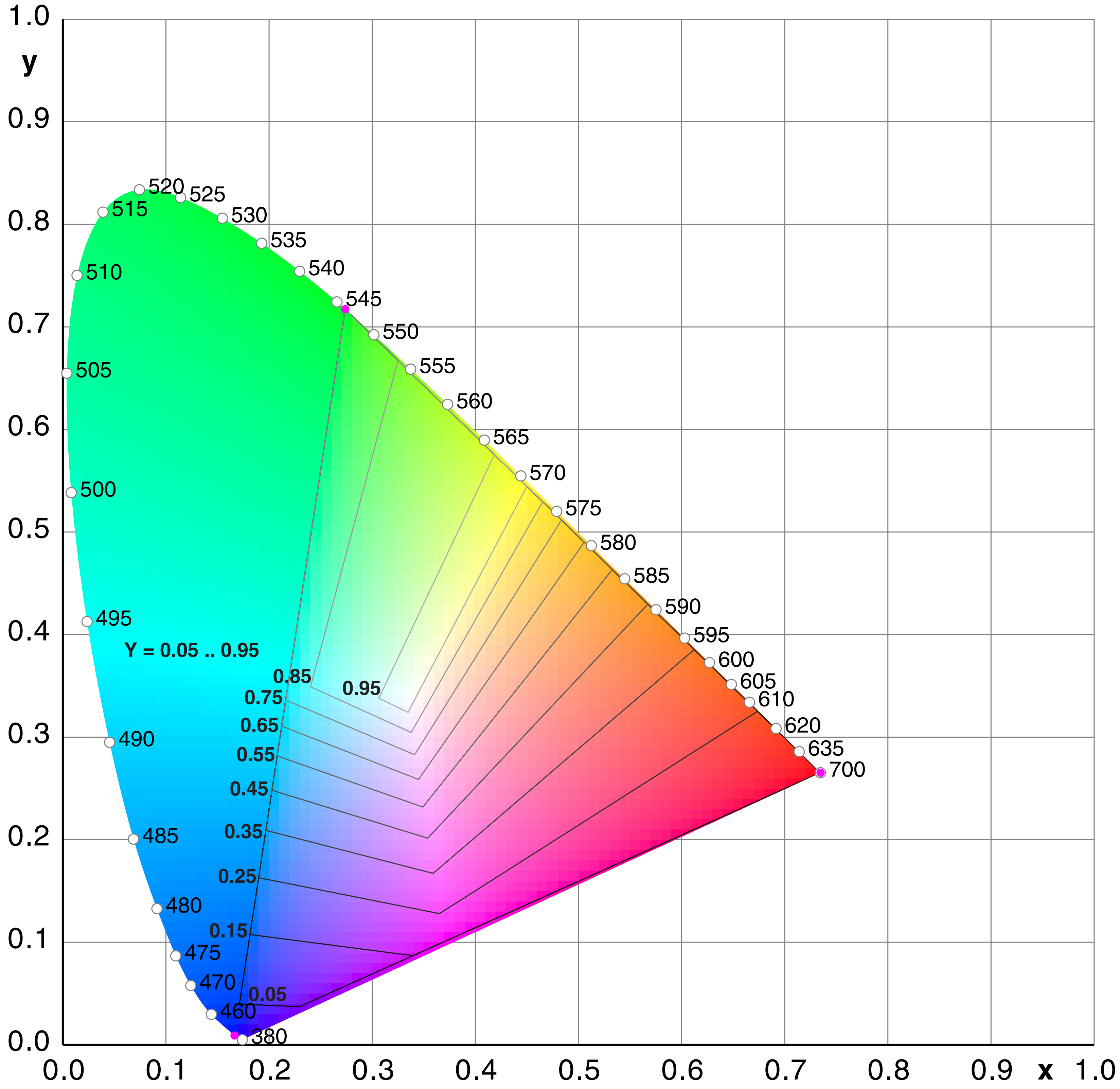


10. CIE RGB Gamut in xyY

The gamut of any RGB system is mostly visualized by a triangle in xyY. For different luminances Y=const. we get the intersection of a vertical plane and the RGB cube (chapter 4). The intersection delivers a triangle, a quadriliteral, a pentagon or a hexagon. These polygons are projected onto the xy-plane

The chromaticity diagram below shows the actual gamut for different luminances Y. Low luminances seem to produce a large gamut. But that is a fake - a result of the perspective projection from XYZ to xyY.

The gamut appears similarly in all RGB systems. A color outside the triangle (which is defined by the primaries) is always out-of-gamut. A color inside the triangle is not necessarily in-gamut.



11.1 Color Space Calculations / General

In this chapter we derive the relations between CIE xyY, CIE XYZ and any arbitrary RGB space. It is essential to understand the principle of RGB basis vectors in the XYZ coordinate system. This was shown on previous pages.

Given are the coordinates for the primaries in CIE xyY and for the white point: $x_r, y_r, x_g, y_g, x_b, y_b, x_w, y_w$. CIE xyY is the horseshoe diagram. Furtheron we need the luminance V.

We want to derive the relation between any color set r,g,b and the coordinates X,Y,Z .

(1) $\mathbf{r} = (r,g,b)^T$ Color values in RGB

(2) $\mathbf{X} = (X,Y,Z)^T$ Color values in XYZ

(3) $\mathbf{x} = (x,y,z)^T$ Color values in xyY

(4) $L = X+Y+Z$ Scaling value

(5) $x = X/L$

$y = Y/L$

$z = Z/L$

(6) $z = 1-x-y$

(7) $\mathbf{X} = L\,\mathbf{x}$

V is the luminance of the stimulus, according to the luminous efficiency function V(λ) in [3]. We should not call this immediately Y because Y is mostly normalized for 1 or 100.

(8) $X = V\,x/y$

$Y = V$

$Z = V\,z/y$

Basis vectors for the primaries and white point in XYZ:

(9) $\mathbf{R} = L\,\mathbf{x}_r = L(x_r, y_r, z_r)^T$

$\mathbf{G} = L\,\mathbf{x}_g = L(x_g, y_g, z_g)^T$

$\mathbf{B} = L\,\mathbf{x}_b = L(x_b, y_b, z_b)^T$

(10) $\mathbf{W} = L\,\mathbf{w} = L(x_w, y_w, z_w)^T$

Set of scale factors for the white point correction:

(11) $\mathbf{u} = (u,v,w)^T$

11.2 Color Space Calculations / General

For the white point correction, the basis vectors **R**,**G**,**B** are scaled by u,v,w. This does not change their coordinates in xyY .The mapping from XYZ to xyY is a central planar projection.

$$(12)\quad \mathbf{X} = L(x,y,z)^T = ru\mathbf{R} + gv\mathbf{G} + bw\mathbf{B}$$

For the white point we have r = g = b = 1.

$$(13)\quad \mathbf{W} = L(x_w,y_w,z_w)^T = Lu(x_r,y_r,z_r)^T + Lv(x_g,y_g,z_g)^T + Lw(x_b,y_b,z_b)^T$$

This can be re-arranged, L cancels on both sides.:

$$(14)\quad \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} = \begin{bmatrix} x_r & x_g & x_b \\ y_r & y_g & y_b \\ z_r & z_g & z_b \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} = \mathbf{P} \begin{bmatrix} u \\ v \\ w \end{bmatrix}$$

It is not necessary to invert the whole matrix numerically. We can simplify the calculation by adding the first two rows to the third row and find so immediately Eq.(15), which is anyway clear:

$$(15)\quad w = 1 - u - v$$

$$(16)\quad \begin{bmatrix} x_w \\ y_w \\ \ \end{bmatrix} = \begin{bmatrix} x_r & x_g & x_b \\ y_r & y_g & y_b \\ & & \end{bmatrix} \begin{bmatrix} u \\ v \\ 1-u-v \end{bmatrix}$$

$$(17)\quad \begin{aligned} x_w &= (x_r - x_b)u + (x_g - x_b)v + x_b \\ y_w &= (y_r - y_b)u + (y_g - y_b)v + y_b \end{aligned}$$

These linear equations are solved by Cramer's rule.

$$(18)\quad \begin{aligned} D &= (x_r - x_b)\ (y_g - y_b) - (y_r - y_b)\ (x_g - x_b) \\ U &= (x_w - x_b)(y_g - y_b) - (y_w - y_b)(x_g - x_b) \\ V &= (x_r - x_b)\ (y_w - y_b) - (y_r - y_b)\ (x_w - x_b) \end{aligned}$$

$$(19)\quad \begin{aligned} u &= U/D \\ v &= V/D \\ w &= 1 - u - v \end{aligned}$$

In the next step we assume that u,v,w are already calculated and we use the general color transformation Eq.(12) and furtheron Eq.(8). We get the matrices $\mathbf{C}_{xr}$ and $\mathbf{C}_{rx}$.

$$(20)\quad \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = V \begin{bmatrix} ux_r/y_w & vx_g/y_w & wx_b/y_w \\ uy_r/y_w & vy_g/y_w & wy_b/y_w \\ uz_r/y_w & vz_g/y_w & wz_b/y_w \end{bmatrix} \begin{bmatrix} r \\ g \\ b \end{bmatrix}$$

$$(21\quad \mathbf{X} = V\mathbf{C}_{xr}\mathbf{r}$$

$$(22)\quad \mathbf{r} = (1/V)\mathbf{C}_{xr}^{-1}\mathbf{X} = (1/V)\mathbf{C}_{rx}\mathbf{X}$$

11.3 Color Space Calculations / General

For better readability we show the last two equations again, but now with V = 1, as in most publications.

(23) $\mathbf{X} = \mathbf{C}_{xr}\,\mathbf{r}$

(24) $\mathbf{r} = \mathbf{C}_{xr}^{-1}\,\mathbf{X} = \mathbf{C}_{rx}\,\mathbf{X}$

Now we can easily derive the relation between two different RGB spaces, e.g. working spaces and image source spaces.

(25) $\mathbf{X} = \mathbf{C}_{xr1}\,\mathbf{r}_1$

(26) $\mathbf{X} = \mathbf{C}_{xr2}\,\mathbf{r}_2$

(27) $\mathbf{r}_2 = \mathbf{C}_{xr2}^{-1}\,\mathbf{C}_{xr1}\,\mathbf{r}_1$

(28) $\mathbf{r}_2 = \mathbf{C}_{21}\,\mathbf{r}_1$

An example shows the conversion of Rec.709/D65 to D50 and D93. The resulting matrix $\mathbf{C}_{21}$ is diagonal, because the source and destination primaries are the same. The explanation as above is valid for the representation of the same physical color in two different RGB systems. For the simulation of D50 or D93 effects in the same D65 RGB system one has to apply the inverse matrix.

```
Rec.709
xr= 0.6400   yr= 0.3300   zr= 0.0300
xg= 0.3000   yg= 0.6000   zg= 0.1000
xb= 0.1500   yb= 0.0600   zb= 0.7900

D65
xw= 0.3127   yw= 0.3290   zw= 0.3583

D50
xw= 0.3457   yw= 0.3585   zw= 0.2958

Matrix Cxr: X=Cxr*R65
0.4124       0.3576       0.1805
0.2126       0.7152       0.0722
0.0193       0.1192       0.9505

Matrix Crx: R65=Crx*X
 3.2410     -1.5374      -0.4986
-0.9692      1.8760       0.0416
 0.0556     -0.2040       1.0570

Matrix Dxr: X=Dxr*R50
0.4852       0.3489       0.1303
0.2502       0.6977       0.0521
0.0227       0.1163       0.6861

Matrix Drx: R50=Drx*X
 2.7548     -1.3068      -0.4238
-0.9935      1.9229       0.0426
 0.0771     -0.2826       1.4644

Matrix Err: R50=Err*R65=Drx*Cxr*R65
0.8500       0.0000       0.0000
0.0000       1.0250       0.0000
0.0000       0.0000       1.3855

Matrix Frr: R65=Frr*R50=Crx*Dxr*R50
1.1765       0.0000       0.0000
0.0000       0.9756       0.0000
0.0000       0.0000       0.7218
```

```
Rec.709
xr= 0.6400   yr= 0.3300   zr= 0.0300
xg= 0.3000   yg= 0.6000   zg= 0.1000
xb= 0.1500   yb= 0.0600   zb= 0.7900

D65
xw= 0.3127   yw= 0.3290   zw= 0.3583

D93
xw= 0.2857   yw= 0.2941   zw= 0.4202

Matrix Cxr: X=Cxr*R65
0.4124       0.3576       0.1805
0.2126       0.7152       0.0722
0.0193       0.1192       0.9505

Matrix Crx: R65=Crx*X
 3.2410     -1.5374      -0.4986
-0.9692      1.8760       0.0416
 0.0556     -0.2040       1.0570

Matrix Dxr: X=Dxr*R93
0.3706       0.3554       0.2455
0.1911       0.7107       0.0982
0.0174       0.1185       1.2929

Matrix Drx: R93=Drx*X
 3.6066     -1.7108      -0.5549
-0.9753      1.8877       0.0418
 0.0409     -0.1500       0.7771

Matrix Err: R93=Err*R65=Drx*Cxr*R65
1.1128       0.0000       0.0000
0.0000       1.0063       0.0000
0.0000       0.0000       0.7352

Matrix Frr: R65=Frr*R93=Crx*Dxr*R93
0.8986       0.0000       0.0000
0.0000       0.9938       0.0000
0.0000       0.0000       1.3602
```

11.4 Color Space Calculations / Simplified

Now we clean up the mathematics. Eq.(14) delivers:

(29) $\mathbf{u} = \mathbf{P}^{-1}\mathbf{w}$

Eq.(12) and Eq.(20) can be written using the diagonal matrix **D** with elements u/y_w etc.:

(30) $\mathbf{X} = \mathbf{PDr}$

(31) $\mathbf{X} = \mathbf{C}_{xr}\,\mathbf{r}$

Together with Eq.(29) we find this simple formula for the matrix $\mathbf{C}_{xr}$:

(32) $$\mathbf{C}_{xr} = \mathbf{P}\begin{bmatrix} u/y_w & 0 & 0 \\ 0 & v/y_w & 0 \\ 0 & 0 & w/y_w \end{bmatrix}$$

The examples in chapter 12 were written by Pascal. Here is a new example in MatLab. Calculation of the matrices for sRGB:

```
 % G.Hoffmann
% January 14 / 2005
% Matrix Cxr and Crx for sRGB

xr=0.6400; yr=0.3300; zr=1-xr-yr;
xg=0.3000; yg=0.6000; zg=1-xg-yg;
xb=0.1500; yb=0.0600; zb=1-xb-yb;
xw=0.3127; yw=0.3290; zw=1-xw-yw;

W=[xw; yw; zw];
P=[xr xg xb;
   yr yg yb;
   zr zg zb];

u=inv(P)*W

% D=[u(1) 0    0;
%    0    u(2) 0;
%    0    0    u(3)]/yw

D=diag(u/yw)

Cxr=P*D
Crx=inv(Cxr)

% Result:

% Cxr 0.4124    0.3576    0.1805
%     0.2126    0.7152    0.0722
%     0.0193    0.1192    0.9505

% Crx 3.2410   -1.5374   -0.4986
%    -0.9692    1.8760    0.0416
%     0.0556   -0.2040    1.0570
```

11.5 Color Space Calculations / Application

The task: red, green and blue lasers generate monochromatic light at wavelengths 671 nm, 532 nm and 473 nm. The powers are to be adjusted so that the three lasers together deliver white light D65. Calculate the matrices, the radiant power ratios and the photometric ratios.

In order to test the algorithms we are doing the same for CIE primaries and Equal Energy White, just as if the lasers had these primaries. The results are known in advance, based on standard text books. Thanks to *Gerhard Fuernkranz* for important clarifications.

CIE primaries and white point E

```
% G.Hoffmann
% January 19 / 2005
% Calculations for CIE primaries
% x-bar,y-bar,z-bar interpolated
% 700.0       546.1         435.8 nm
xbr=0.011359; xbg=0.375540; xbb=0.333181;
ybr=0.004102; ybg=0.984430; ybb=0.017769;
zbr=0.000000; zbg=0.012207; zbb=1.649716;
% Equal Energy WP
Xw=1; Yw=1; Zw=1;

%Chromaticity coordinates
D=xbr+ybr+zbr; xr=xbr/D;  yr=ybr/D; zr=zbr/D;
D=xbg+ybg+zbg; xg=xbg/D;  yg=ybg/D; zg=zbg/D;
D=xbb+ybb+zbb; xb=xbb/D;  yb=ybb/D; zb=zbb/D;
D=Xw +Yw+ Zw;  xw=Xw/D;   yw=Yw/D;   zw=Zw/D;

w=[xw; yw; zw];
P=[xr xg xb;
   yr yg yb;
   zr zg zb];

u=inv(P)*w
D=diag(u/yw)

Cxr=P*D
%  0.4902    0.3099    0.1999
%  0.1770    0.8123    0.0107
%  0.0000    0.0101    0.9899
Crx=inv(Cxr)
%  2.3635   -0.8958   -0.4677
% -0.5151    1.4265    0.0887
%  0.0052   -0.0145    1.0093

% Radiant power ratios
Xbar=[xbr xbg xbb;
      ybr ybg ybb;
      zbr zbg zbb];
W=[Xw; Yw; Zw];

R=inv(Xbar)*W

R=R/R(3)
% 71.9166 1.3751 1.0000
% 72.0962 1.3791 1.0000  Wyszecki & Stiles

% Luminous efficiency ratios
L=[R(1)*ybr; R(2)*ybg; R(3)*ybb]
L=L/L(1)
%  1.0000 4.5889 0.0602
%  1.0000 4.5907 0.0601  Wyszecki & Stiles
```

Laser primaries and white point D65

```
% G.Hoffmann
% January 19 / 2005
% Calculations for Laser primaries
% x-bar,y-bar,z-bar interpolated
% 671       532         473 nm
xbr=0.0819; xbg=0.1891; xbb=0.1627;
ybr=0.0300; ybg=0.8850; ybb=0.1034;
zbr=0.0000; zbg=0.0369; zbb=1.1388;
% D65
Xw=0.9504; Yw=1.0000; Zw=1.0890;

%Chromaticity coordinates
D=xbr+ybr+zbr; xr=xbr/D; yr=ybr/D; zr=zbr/D;
D=xbg+ybg+zbg; xg=xbg/D; yg=ybg/D; zg=zbg/D;
D=xbb+ybb+zbb; xb=xbb/D; yb=ybb/D; zb=zbb/D;
D=Xw +Yw+ Zw;  xw=Xw/D;  yw=Yw/D;  zw=Zw/D;

w=[xw; yw; zw];
P=[xr xg xb;
   yr yg yb;
   zr zg zb];

u=inv(P)*w
D=diag(u/yw)

Cxr=P*D
%  0.6571    0.1416    0.1516
%  0.2407    0.6629    0.0964
%       0    0.0276    1.0614
Crx=inv(Cxr)
%  1.6476   -0.3435   -0.2042
% -0.6005    1.6394   -0.0631
%  0.0156   -0.0427    0.9438

% Radiant power ratios
Xbar=[xbr xbg xbb;
      ybr ybg ybb;
      zbr zbg zbb];
W=[Xw; Yw; Zw];

R=inv(Xbar)*W

R=R/R(1)
%   1.0000  0.0934  0.1162
R=R/R(2)
%  10.7111  1.0000  1.2442
R=R/R(3)
%   8.6088  0.8037  1.0000

% Luminous efficiency ratios
L=[R(1)*ybr; R(2)*ybg; R(3)*ybb];
L=L/L(1)
%  1.0000   2.7542  0.4004
L=L/L(2)
%  0.3631   1.0000  0.1454
L=L/L(3)
%  2.4977   6.8791  1.0000
```

12.1 Matrices / CIE + E

CIE Primaries and white point E [3]. Page 5 shows the same results.
Data are in the Pascal source code.

```
Program CiCalcCi;
{ Calculations RGB—CIE }
{ G.Hoffmann February 01, 2002 }
Uses Crt,Dos,Zgraph00;
Var r,g,b,x,y,z,u,v,w,d                 : Extended;
    i,j,k,flag                          : Integer;
    xr,yr,zr,xg,yg,zg,xb,yb,zb,xw,yw,zw : Extended;
    prn,cie                             : Text;
Var Cxr,Crx: ANN;
Begin
ClrScr;
{ CIE Primaries }
xr:=0.73467;
yr:=0.26533;
zr:=1-xr-yr;
xg:=0.27376;
yg:=0.71741;
zg:=1-xg-yg;
xb:=0.16658;
yb:=0.00886;
zb:=1-xb-yb;
{ CIE White Point }
xw:=1/3;
yw:=1/3;
zw:=1-xw-yw;
{ White Point Correction }
D:=(xr-xb)*(yg-yb)-(yr-yb)*(xg-xb);
U:=(xw-xb)*(yg-yb)-(yw-yb)*(xg-xb);
V:=(xr-xb)*(yw-yb)-(yr-yb)*(xw-xb);
u:=U/D;
v:=V/D;
w:=1-u-v;
{ Matrix Cxr }
Cxr[1,1]:=u*xr/yw; Cxr[1,2]:=v*xg/yw; Cxr[1,3]:=w*xb/yw;
Cxr[2,1]:=u*yr/yw; Cxr[2,2]:=v*yg/yw; Cxr[2,3]:=w*yb/yw;
Cxr[3,1]:=u*zr/yw; Cxr[3,2]:=v*zg/yw; Cxr[3,3]:=w*zb/yw;
{ Matrix Crx }
HoInvers (3,Cxr,Crx,D,flag);

Assign (prn,'C:\CiMalcCi.txt'); ReWrite(prn);

Writeln (prn,'      Matrix Cxr');
Writeln (prn,Cxr[1,1]:12:4, Cxr[1,2]:12:4, Cxr[1,3]:12:4);
Writeln (prn,Cxr[2,1]:12:4, Cxr[2,2]:12:4, Cxr[2,3]:12:4);
Writeln (prn,Cxr[3,1]:12:4, Cxr[3,2]:12:4, Cxr[3,3]:12:4);

Writeln (prn,'      Matrix Crx');
Writeln (prn,Crx[1,1]:12:4, Crx[1,2]:12:4, Crx[1,3]:12:4);
Writeln (prn,Crx[2,1]:12:4, Crx[2,2]:12:4, Crx[2,3]:12:4);
Writeln (prn,Crx[3,1]:12:4, Crx[3,2]:12:4, Crx[3,3]:12:4);
Close(prn);
Readln;
End.
```

```
Matrix Cxr
X    0.4900      0.3100      0.2000
Y    0.1770      0.8124      0.0106
Z   -0.0000      0.0100      0.9900

Matrix Crx
R    2.3647     -0.8966     -0.4681
G   -0.5152      1.4264      0.0887
B    0.0052     -0.0144      1.0092
```

$$X = C_{xr} R$$

$$R = C_{rx} X$$

12.2 Matrices / 709 + D65 / sRGB

ITU-R BT.709 Primaries and white point D65 [9]. Valid for sRGB.
Data are in the Pascal source code.

```
Program CiCalc65;
{ Calculations RGB—CIE }
{ G.Hoffmann February 01, 2002 }
Uses Crt,Dos,Zgraph00;
Var r,g,b,x,y,z,u,v,w,d                 : Extended;
    i,j,k,flag                          : Integer;
    xr,yr,zr,xg,yg,zg,xb,yb,zb,xw,yw,zw : Extended;
    prn,cie                             : Text;
Var Cxr,Crx: ANN;
Begin
ClrScr;
{ Rec 709 Primaries }
xr:=0.6400;
yr:=0.3300;
zr:=1-xr-yr;
xg:=0.3000;
yg:=0.6000;
zg:=1-xg-yg;
xb:=0.1500;
yb:=0.0600;
zb:=1-xb-yb;
{ D65 White Point }
xw:=0.3127;
yw:=0.3290;
zw:=1-xw-yw;
{ White Point Correction }
D:=(xr-xb)*(yg-yb)-(yr-yb)*(xg-xb);
U:=(xw-xb)*(yg-yb)-(yw-yb)*(xg-xb);
V:=(xr-xb)*(yw-yb)-(yr-yb)*(xw-xb);
u:=U/D;
v:=V/D;
w:=1-u-v;
{ Matrix Cxr }
Cxr[1,1]:=u*xr/yw; Cxr[1,2]:=v*xg/yw; Cxr[1,3]:=w*xb/yw;
Cxr[2,1]:=u*yr/yw; Cxr[2,2]:=v*yg/yw; Cxr[2,3]:=w*yb/yw;
Cxr[3,1]:=u*zr/yw; Cxr[3,2]:=v*zg/yw; Cxr[3,3]:=w*zb/yw;
{ Matrix Crx }
HoInvers (3,Cxr,Crx,D,flag);

Assign (prn,'C:\CiMalc65.txt'); ReWrite(prn);

Writeln (prn,'      Matrix Cxr');
Writeln (prn,Cxr[1,1]:12:4, Cxr[1,2]:12:4, Cxr[1,3]:12:4);
Writeln (prn,Cxr[2,1]:12:4, Cxr[2,2]:12:4, Cxr[2,3]:12:4);
Writeln (prn,Cxr[3,1]:12:4, Cxr[3,2]:12:4, Cxr[3,3]:12:4);

Writeln (prn,'      Matrix Crx');
Writeln (prn,Crx[1,1]:12:4, Crx[1,2]:12:4, Crx[1,3]:12:4);
Writeln (prn,Crx[2,1]:12:4, Crx[2,2]:12:4, Crx[2,3]:12:4);
Writeln (prn,Crx[3,1]:12:4, Crx[3,2]:12:4, Crx[3,3]:12:4);
Close(prn);
Readln;
End.
```

```
 Matrix Cxr
X  0.4124      0.3576      0.1805
Y  0.2126      0.7152      0.0722
Z  0.0193      0.1192      0.9505
```

$$\mathbf{X} = \mathbf{C}_{xr}\mathbf{R}$$

```
 Matrix Crx
R  3.2410     -1.5374     -0.4986
G -0.9692      1.8760      0.0416
B  0.0556     -0.2040      1.0570
```

$$\mathbf{R} = \mathbf{C}_{rx}\mathbf{X}$$

12.3 Matrices / AdobeRGB + D65

AdobeRGB(98), D65.
Data are in the Pascal source code.

```
Program CiCalc98;
{ Calculations RGB—AdobeRGB98 }
{ G.Hoffmann März 28, 2004 }
Uses Crt,Dos,Zgraph00;
Var r,g,b,x,y,z,u,v,w,d                 : Double;
    i,j,k,flag                          : Integer;
    xr,yr,zr,xg,yg,zg,xb,yb,zb,xw,yw,zw : Double;
    prn,cie                             : Text;
Var Cxr,Crx: ANN;
Begin
ClrScr;
{ AdobeRGB(98) }
xr:=0.6400;
yr:=0.3300;
zr:=1-xr-yr;
xg:=0.2100;
yg:=0.7100;
zg:=1-xg-yg;
xb:=0.1500;
yb:=0.0600;
zb:=1-xb-yb;
{ D65 White Point }
xw:=0.3127;
yw:=0.3290;
zw:=1-xw-yw;
{ White Point Correction }
D:=(xr-xb)*(yg-yb)-(yr-yb)*(xg-xb);
U:=(xw-xb)*(yg-yb)-(yw-yb)*(xg-xb);
V:=(xr-xb)*(yw-yb)-(yr-yb)*(xw-xb);
u:=U/D;
v:=V/D;
w:=1-u-v;
{ Matrix Cxr }
Cxr[1,1]:=u*xr/yw; Cxr[1,2]:=v*xg/yw; Cxr[1,3]:=w*xb/yw;
Cxr[2,1]:=u*yr/yw; Cxr[2,2]:=v*yg/yw; Cxr[2,3]:=w*yb/yw;
Cxr[3,1]:=u*zr/yw; Cxr[3,2]:=v*zg/yw; Cxr[3,3]:=w*zb/yw;
{ Matrix Crx }
HoInvers (3,Cxr,Crx,D,flag);

Assign (prn,'C:\CiMalc98.txt'); ReWrite(prn);

Writeln (prn,'      Matrix Cxr');
Writeln (prn,Cxr[1,1]:12:4, Cxr[1,2]:12:4, Cxr[1,3]:12:4);
Writeln (prn,Cxr[2,1]:12:4, Cxr[2,2]:12:4, Cxr[2,3]:12:4);
Writeln (prn,Cxr[3,1]:12:4, Cxr[3,2]:12:4, Cxr[3,3]:12:4);
Writeln (prn,'');
Writeln (prn,'      Matrix Crx');
Writeln (prn,Crx[1,1]:12:4, Crx[1,2]:12:4, Crx[1,3]:12:4);
Writeln (prn,Crx[2,1]:12:4, Crx[2,2]:12:4, Crx[2,3]:12:4);
Writeln (prn,Crx[3,1]:12:4, Crx[3,2]:12:4, Crx[3,3]:12:4);
Writeln (prn,'dummy');
Readln;
End.
```

```
Matrix Cxr
X  0.5767      0.1856      0.1882
Y  0.2973      0.6274      0.0753
Z  0.0270      0.0707      0.9913

Matrix Crx
R  2.0416     -0.5650     -0.3447
G -0.9692      1.8760      0.0416
B  0.0134     -0.1184      1.0152
```

$$\mathbf{X} = \mathbf{C}_{xr}\mathbf{R}$$

$$\mathbf{R} = \mathbf{C}_{rx}\mathbf{X}$$

12.4 Matrices / NTSC + C

NTSC Primaries and white point C [4], also used as YIQ Model.
Data are in the Pascal source code.

```
Program CiCalcNT;
{ Calculations RGB–NTSC }
{ G.Hoffmann April 01, 2002 }
Uses Crt,Dos,Zgraph00;
Var r,g,b,x,y,z,u,v,w,d                 : Extended;
    i,j,k,flag                          : Integer;
    xr,yr,zr,xg,yg,zg,xb,yb,zb,xw,yw,zw : Extended;
    prn,cie                             : Text;
Var Cxr,Crx: ANN;
Begin
ClrScr;
{ NTSC Primaries }
xr:=0.6700;
yr:=0.3300;
zr:=1-xr-yr;
xg:=0.2100;
yg:=0.7100;
zg:=1-xg-yg;
xb:=0.1400;
yb:=0.0800;
zb:=1-xb-yb;
{ NTSC White Point }
xw:=0.3100;
yw:=0.3160;
zw:=1-xw-yw;
{ White Point Correction }
D:=(xr-xb)*(yg-yb)-(yr-yb)*(xg-xb);
U:=(xw-xb)*(yg-yb)-(yw-yb)*(xg-xb);
V:=(xr-xb)*(yw-yb)-(yr-yb)*(xw-xb);
u:=U/D;
v:=V/D;
w:=1-u-v;
{ Matrix Cxr }
Cxr[1,1]:=u*xr/yw; Cxr[1,2]:=v*xg/yw; Cxr[1,3]:=w*xb/yw;
Cxr[2,1]:=u*yr/yw; Cxr[2,2]:=v*yg/yw; Cxr[2,3]:=w*yb/yw;
Cxr[3,1]:=u*zr/yw; Cxr[3,2]:=v*zg/yw; Cxr[3,3]:=w*zb/yw;
{ Matrix Crx }
HoInvers (3,Cxr,Crx,D,flag);

Assign (prn,'C:\CiMalcNT.txt'); ReWrite(prn);

Writeln (prn,'      Matrix Cxr');
Writeln (prn,Cxr[1,1]:12:4, Cxr[1,2]:12:4, Cxr[1,3]:12:4);
Writeln (prn,Cxr[2,1]:12:4, Cxr[2,2]:12:4, Cxr[2,3]:12:4);
Writeln (prn,Cxr[3,1]:12:4, Cxr[3,2]:12:4, Cxr[3,3]:12:4);
Writeln (prn,'');
Writeln (prn,'      Matrix Crx');
Writeln (prn,Crx[1,1]:12:4, Crx[1,2]:12:4, Crx[1,3]:12:4);
Writeln (prn,Crx[2,1]:12:4, Crx[2,2]:12:4, Crx[2,3]:12:4);
Writeln (prn,Crx[3,1]:12:4, Crx[3,2]:12:4, Crx[3,3]:12:4);
Close(prn);
Readln;
End.
```

```
Matrix Cxr
X    0.6070      0.1734      0.2006
Y    0.2990      0.5864      0.1146
Z   -0.0000      0.0661      1.1175
```

$$X = C_{xr} R$$

```
Matrix Crx
R    1.9097     -0.5324     -0.2882
G   -0.9850      1.9998     -0.0283
B    0.0582     -0.1182      0.8966
```

$$R = C_{rx} X$$

12.5 Matrices / NTSC + C + YIQ

NTSC Primaries and white point C [4], YIQ Conversion.
Data are in the Pascal source code.

```
Program CiCalcYI;
{ Calculations RGB–NTSC YIQ }
{ G.Hoffmann April 01, 2002 }
Uses Crt,Dos,Zgraph00;
Var r,g,b,x,y,z,u,v,w,d                  : Extended;
    i,j,k,flag                           : Integer;
    xr,yr,zr,xg,yg,zg,xb,yb,zb,xw,yw,zw : Extended;
    prn,cie                              : Text;
Var Cyr,Cry: ANN;
Begin
ClrScr;
{ NTSC Primaries }
xr:=0.6700;
yr:=0.3300;
zr:=1-xr-yr;
xg:=0.2100;
yg:=0.7100;
zg:=1-xg-yg;
xb:=0.1400;
yb:=0.0800;
zb:=1-xb-yb;
{ NTSC White Point }
xw:=0.3100;
yw:=0.3160;
zw:=1-xw-yw;
{ Matrix Cyr, Sequence Y I Q }
Cyr[1,1]:= 0.299; Cyr[1,2]:= 0.587; Cyr[1,3]:= 0.114;
Cyr[2,1]:= 0.596; Cyr[2,2]:=-0.275; Cyr[2,3]:=-0.321;
Cyr[3,1]:= 0.212; Cyr[3,2]:=-0.528; Cyr[3,3]:= 0.311;
{ Matrix Cry }
HoInvers (3,Cyr,Cry,D,flag);

Assign (prn,'C:\CiMalcYI.txt'); ReWrite(prn);

Writeln (prn,'      Matrix Cyr');
Writeln (prn,Cyr[1,1]:12:4, Cyr[1,2]:12:4, Cyr[1,3]:12:4);
Writeln (prn,Cyr[2,1]:12:4, Cyr[2,2]:12:4, Cyr[2,3]:12:4);
Writeln (prn,Cyr[3,1]:12:4, Cyr[3,2]:12:4, Cyr[3,3]:12:4);
Writeln (prn,'');
Writeln (prn,'      Matrix Cry');
Writeln (prn,Cry[1,1]:12:4, Cry[1,2]:12:4, Cry[1,3]:12:4);
Writeln (prn,Cry[2,1]:12:4, Cry[2,2]:12:4, Cry[2,3]:12:4);
Writeln (prn,Cry[3,1]:12:4, Cry[3,2]:12:4, Cry[3,3]:12:4);
Close(prn);
Readln;
End.
```

```
Matrix Cyr
Y    0.2990      0.5870      0.1140
I    0.5960     -0.2750     -0.3210
Q    0.2120     -0.5280      0.3110
```

$$\mathbf{Y} = \mathbf{C}_{yr}\mathbf{R}$$

```
Matrix Cry
R    1.0031      0.9548      0.6179
G    0.9968     -0.2707     -0.6448
B    1.0085     -1.1105      1.6996
```

$$\mathbf{R} = \mathbf{C}_{ry}\mathbf{Y}$$

12.6 Matrices / NTSC + C + YCbCr

NTSC Primaries and white point C [4], YCbCr Conversion.
Data are in the Pascal source code.

```
Program CiCalcYC;
{ Calculations RGB—NTSC YCbCr }
{ G.Hoffmann April 03, 2002 }
Uses Crt,Dos,Zgraph00;
Var r,g,b,x,y,z,u,v,w,d                 : Extended;
    i,j,k,flag                          : Integer;
    xr,yr,zr,xg,yg,zg,xb,yb,zb,xw,yw,zw : Extended;
    prn,cie                             : Text;
Var Cyr,Cry: ANN;
Begin
ClrScr;
{ NTSC Primaries }
xr:=0.6700;
yr:=0.3300;
zr:=1-xr-yr;
xg:=0.2100;
yg:=0.7100;
zg:=1-xg-yg;
xb:=0.1400;
yb:=0.0800;
zb:=1-xb-yb;
{ NTSC White Point }
xw:=0.3100;
yw:=0.3160;
zw:=1-xw-yw;
{ Matrix Cxr, Sequence Y Cb Cr }
Cyr[1,1]:= 0.2990; Cyr[1,2]:= 0.5870; Cyr[1,3]:= 0.1140;
Cyr[2,1]:=-0.1687; Cyr[2,2]:=-0.3313; Cyr[2,3]:=+0.5000;
Cyr[3,1]:= 0.5000; Cyr[3,2]:=-0.4187; Cyr[3,3]:=-0.0813;
{ Matrix Cry }
HoInvers (3,Cyr,Cry,D,flag);

Assign (prn,'C:\CiMalcYC.txt'); ReWrite(prn);

Writeln (prn,'      Matrix Cyr');
Writeln (prn,Cyr[1,1]:12:4, Cyr[1,2]:12:4, Cyr[1,3]:12:4);
Writeln (prn,Cyr[2,1]:12:4, Cyr[2,2]:12:4, Cyr[2,3]:12:4);
Writeln (prn,Cyr[3,1]:12:4, Cyr[3,2]:12:4, Cyr[3,3]:12:4);
Writeln (prn,'');
Writeln (prn,'      Matrix Cry');
Writeln (prn,Cry[1,1]:12:4, Cry[1,2]:12:4, Cry[1,3]:12:4);
Writeln (prn,Cry[2,1]:12:4, Cry[2,2]:12:4, Cry[2,3]:12:4);
Writeln (prn,Cry[3,1]:12:4, Cry[3,2]:12:4, Cry[3,3]:12:4);
Close(prn);
Readln;
End.
```

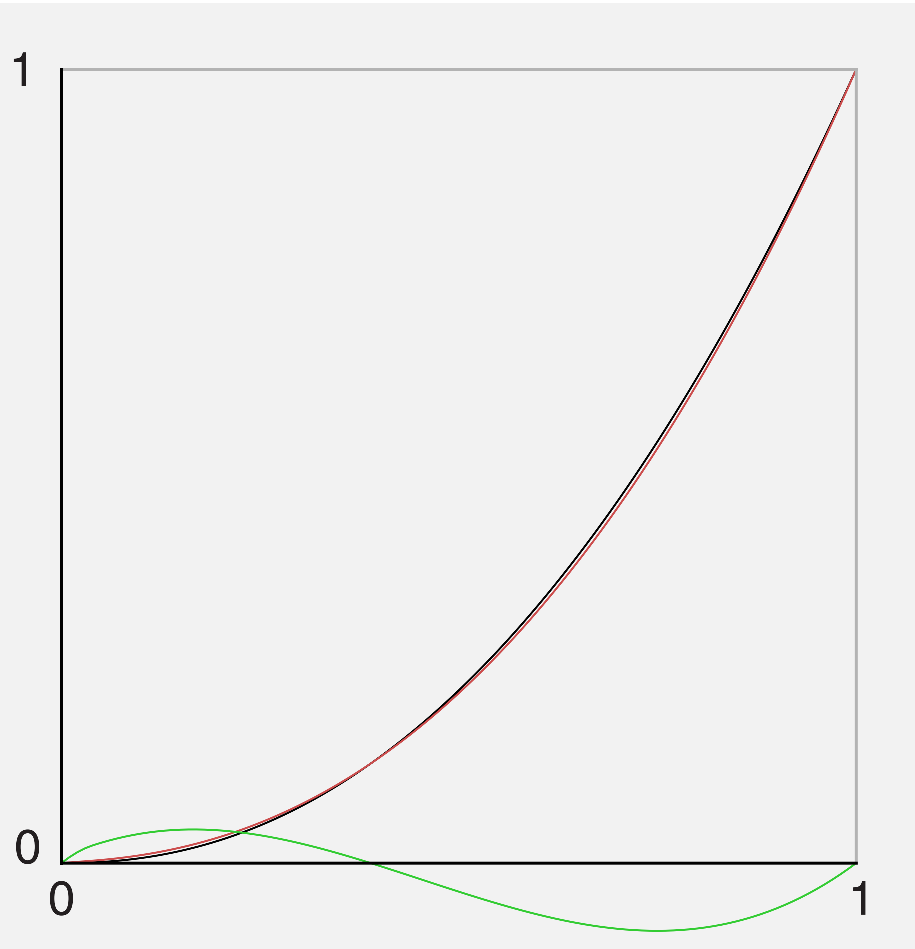
```
Matrix Cyr
Y    0.2990      0.5870      0.1140       Note
Cb  -0.1687     -0.3313      0.5000       This is a linear conversion, as used for JPEG
Cr   0.5000     -0.4187     -0.0813       In TV systems the conversion is different

Matrix Cry
R    1.0000      0.0000      1.4020       Note
G    1.0000     -0.3441     -0.7141       Rounded for structural zeros
B    1.0000      1.7722      0.0000
```

13. sRGB

sRGB is a standard color space, defined by companies, mainly Hewlett-Packard and Microsoft [9], [12].
The transformation of RGB image data to CIE XYZ requires primarily a Gamma correction, which compensates an expected inverse Gamma correction, compared to linear light data, here for normalized values C = R,G,B = 0...1:

If $C \le 0.03928$ Then $C = C/12.92$
Else $C = ((0.055+C)/1.055)^{2.4}$

The formula in the document [12] is misleading because a bracket was forgotten.



Black $C = C^{2.2}$

Red sRGB, as above

Green ten times the difference

The conversion for D65 RGB to D65 XYZ uses the matrix on page 14, ITU-R BT.709 Primaries. D65 XYZ means XYZ without changing the illuminant.

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{D65} = \begin{bmatrix} 0.4124 & 0.3576 & 0.1805 \\ 0.2126 & 0.7152 & 0.0722 \\ 0.0193 & 0.1192 & 0.9505 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}_{D65}$$

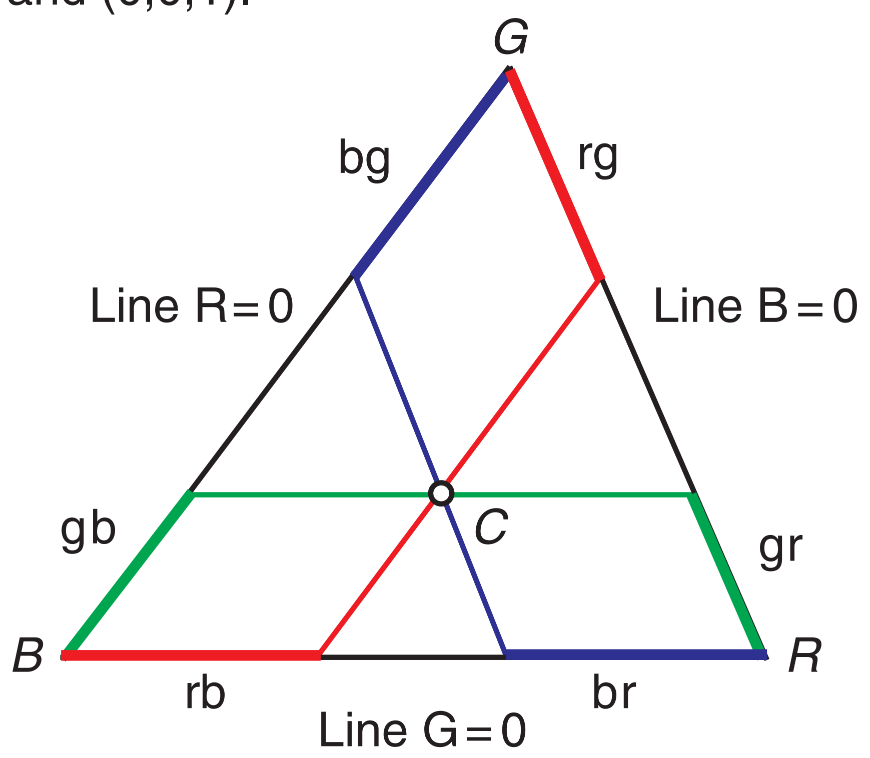
The conversion for D65 RGB to D50 XYZ applies additionally (by multiplication) the Bradford correction, which takes the adaptation of the eyes into account. This correction is an improved alternative to the Von Kries corrrection [1].

Monitors are assumed D65, but for printed paper the standard illuminant is D50. Therefore this transformation is recommended if the data are used for printing:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{D50} = \begin{bmatrix} 0.4361 & 0.3851 & 0.1431 \\ 0.2225 & 0.7169 & 0.0606 \\ 0.0139 & 0.0971 & 0.7141 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}_{D65}$$

14.1 Barycentric Coordinates / Concept

The corners *R*,*G*,*B* of a triangular gamut, e.g. for a monitor, are described in CIE xyY by three vectors **r**,**g**,**b** which have two components x,y each.
A color *C* is described either by **c** with two values c_x, c_y or by three values R,G,B. These are the barycentric coordinates of *C*.
All points inside and on the triangle are reachable by $0 \le R,G,B \le 1$. Points outside have at least one negative coordinate. The corners *R*,*G*,*B* have barycentric coordinates (1,0,0), (0,1,0) and (0,0,1).



$rg = R\,\underline{RG}$

$rb = R\,\underline{RB}$

$gr = G\,\underline{GR}$

$gb = G\,\underline{GB}$

$bg = B\,\underline{BG}$

$br = B\,\underline{BR}$

Underline means length of ..

(1) $\mathbf{c} = R\mathbf{r} + G\mathbf{g} + B\mathbf{b}$

(2) $1 = R + G + B$

Substitute R in(1) by (2):

(3) $R = 1 - G - B$

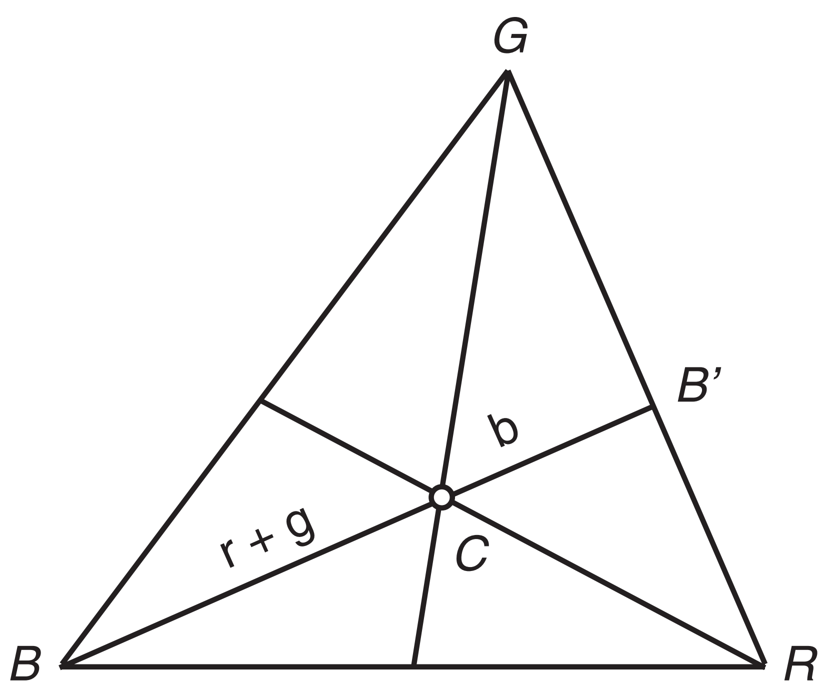
(4) $G(\mathbf{g} - \mathbf{r}) + B(\mathbf{b} - \mathbf{r}) = \mathbf{c} - \mathbf{r}$

(4) consists of two linear equations for G,B, which can be solved by *Cramer's* rule.

R is calculated by (3).

$(\mathbf{g} - \mathbf{r})$ and $(\mathbf{b} - \mathbf{r})$ are the edge vectors from *R* to *G* and *R* to *B*. The edge vectors are used in (4) as a vector base.

Any point inside the triangle is reached by $G + B < 1$, which leads to $R + G + B = 1$.



$r+g = (R+G)\,\underline{BB'}$

$b = B\,\underline{BB'}$

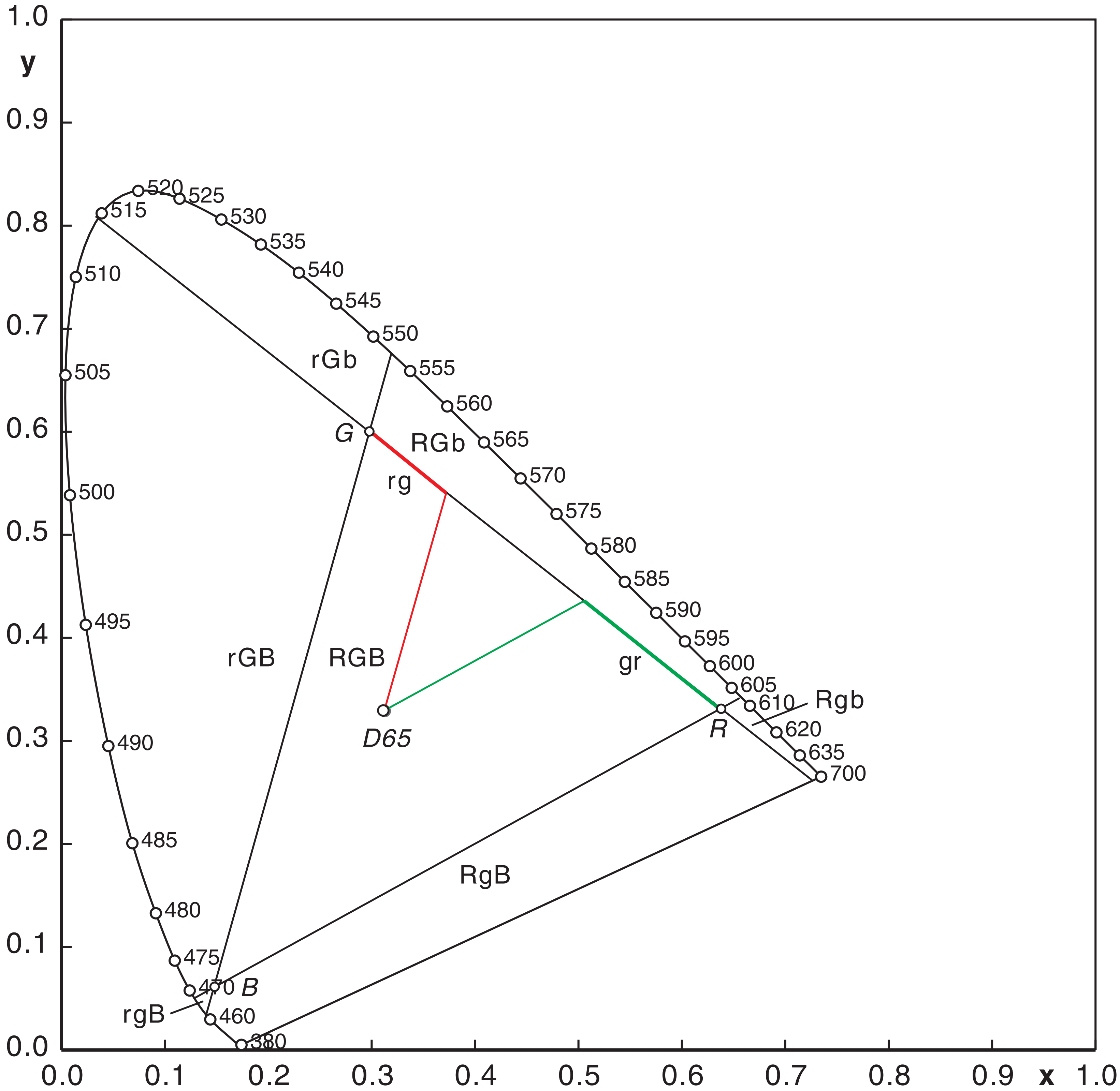
14.2 Barycentric Coordinates / Wrong

So far the barycentric coordinates remind much to the explanations in [3], chapter 3.2.2. It should be possible to find the relative values R,G,B for a given point **c**=(c_x,c_y) by measuring the proportions R=rg/<u>*RG*</u>, G=gr/<u>*GR*</u> with <u>*RG*</u>=<u>*GR*</u>, then B=1-R-G.

Unfortunately this interpretation is wrong. The drawing shows the D65 white point and the measurable values R=0.219, G=0.385 and B=0.396 instead of the correct values R=1/3, G=1/3, B=1/3.

The base vectors **R**,**G**,**B** in CIE XYZ (chapter 4 for CIE primaries) do not have the same lengths. In [3] the mathematics were explained for unit vectors.
So far it is not clear, how the geometrically interpretation for barycentric coordinates could be applied to the actual task.

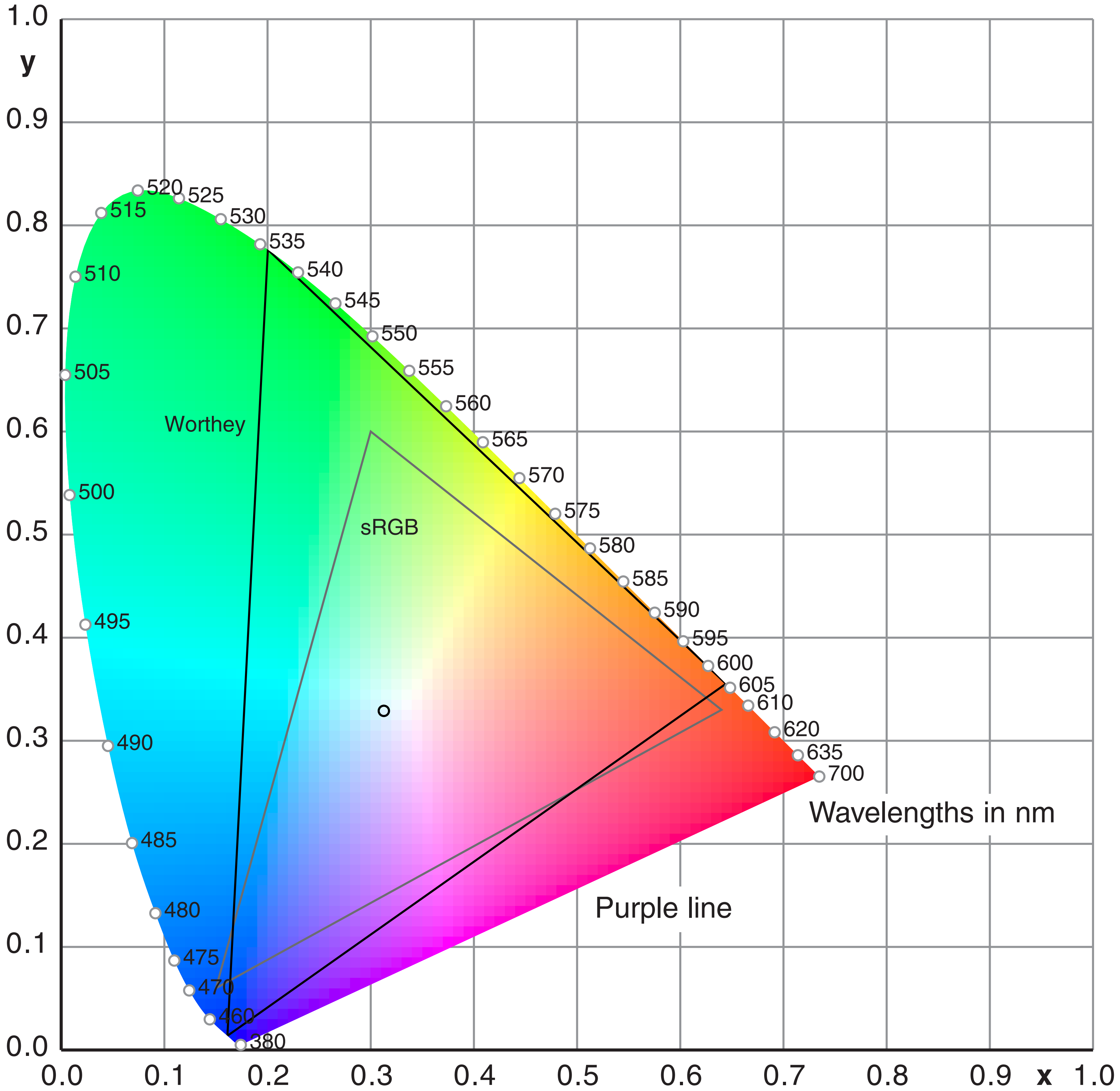
The diagram below shows additionally seven sectors. 'RGB' means, all values are positive (inside the triangle). 'rGB' means R<0, G>0, B>0 and so on. Negative values are not prohibited by the definition of coordinates. They just do not appear in technical RGB system. Of course they are essential for the color matching theory.



15. Optimal Primaries

James A. Worthey had shown in recent publications [18] how to find optimal primaries. This approach is based on 'Amplitude not left out' . Which primaries should be used if the power is limited for each light source ?
The resulting wavelengths are shown by the corners of the triangle below: 445, 536, 604 nm. At least, the wavelengths should be near to these values.
For a real system (besides tests in a laboratory) pure spectral colors cannot be used. The corners have to be shifted on a radius towards the white point (which is here indicated by the circle for D65).
The optimal red at 604nm is hardly a good candidate for technical systems - it is more a kind of orange instead of vibrant red.

Additional illustrations for J.Worthey's concepts are in [19]. Everything PostScript vector graphics.



16.1 References

[1] R.W.G.Hunt
Measuring Colour
Fountain Press, England, 1998

[2] E.J.Giorgianni + Th.E.Madden
Digital Color Management
Addison-Wesley, Reading Massachusetts ,..., 1998

[3] G.Wyszecki + W.S.Stiles
Color Science
John Wiley & Sons, New York ,..., 1982

[4] J.D.Foley + A.van Dam+ St.K.Feiner + J.F.Hughes
Computer Graphics
Addison-Wesley, Reading Massachusetts,...,1993

[5] C.H.Chen + L.F.Pau + P.S.P.Wang
Handbook of Pttern recognition and Computer Vision
World Scientific, Singapore, ..., 1995

[6] J.J.Marchesi
Handbuch der Fotografie Vol. 1 - 3
Verlag Fotografie, Schaffhausen, 1993

[7] T.Autiokari
Accurate Image Processing
http://www.aim-dtp.net
2001

[8] Ch.Poynton
Frequently asked questions about Gamma
http://www.inforamp.net/~poynton/
1997

[9] M.Stokes + M.Anderson + S.Chandrasekar + R.Motta
A Standard Default Color Space for the Internet - sRGB
http://www.w3.org/graphics/color/srgb.html
1996

[10] G.Hoffmann
Corrections for Perceptually Optimized Grayscales
http://www.fho-emden.de/~hoffmann/optigray06102001.pdf
2001

[11] G.Hoffmann
Hardware Monitor Calibration
http://www.fho-emden.de/~hoffmann/caltutor270900.pdf
2001

[12] M.Nielsen + M.Stokes
The Creation of the sRGB ICC Profile
http://www.srgb.com/c55.pdf
Year unknown, after 1998

[13] G.Hoffmann
CieLab Color Space
http://www.fho-emden.de/~hoffmann/cielab03022003.pdf

[14] Everything about Color and Computers
http://www.efg2.com

16.2 References

[15] CIE Chromaticity Diagram, EPS Graphic
http://www.fho-emden.de/~hoffmann/ciesuper.txt

[16] Color-Matching Functions RGB, EPS Graphic
http://www.fho-emden.de/~hoffmann/matchrgb.txt

[17] Color-Matching Functions XYZ, EPS Graphic
http://www.fho-emden.de/~hoffmann/matchxyz.txt

[18] James A. Worthey
Color Matching with Amplitude Not Left Out
http://users.starpower.net/jworthey/FinalScotts2004Aug25.pdf

[19] G.Hoffmann
Locus of Unit Monochromats
http://www.fho-emden.de/~hoffmann/jimcolor12062004.pdf

This document
http://www.fho-emden.de/~hoffmann/ciexyz29082000.pdf

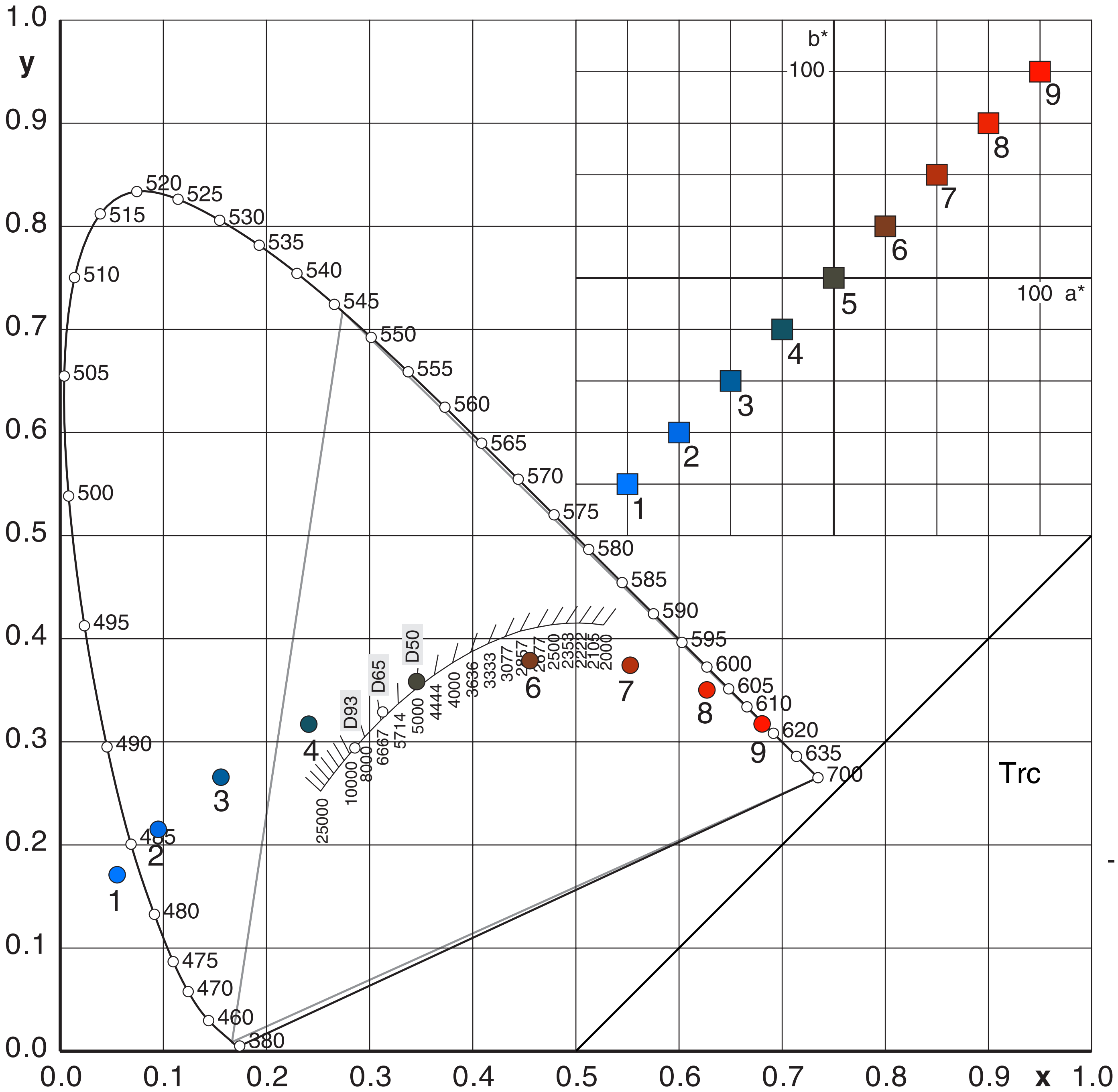
Gernot Hoffmann
December 07 /2006
Website
Load Browser / Click here

A. Appendix Color Matching

The calculation shows colors as defined by equal distances in CIELab. The corresponding values are drawn in the chromaticity diagram.
BlueGreen (4) can be matched by positive weights RGB for CIE primaries.
Vibrant BlueGreen (2) requires negative R. BlueGreen (1) is out of human gamut.
RGB values are here normalized for 0...100.

ColorCalc	**Med.White:**	**Eq.Energy**	**Primaries:**	**CIE**	**Intent:**	**AbsCol**	
G.Hoffmann	**Ref.White:**	**D50**	**Trc:**	**1.0**			
Dec.06 / 2006	**Input:**	**Lab**	**Bradford:**	**No**	**Set:**	**13**	

X	0.090936	0.124317	0.165004	0.213721	0.271192	0.338140	0.415287	0.503358	0.603075
Y	0.281233	0.281233	0.281233	0.281233	0.281233	0.281233	0.281233	0.281233	0.281233
Z	1.271889	0.902067	0.611932	0.391815	0.232047	0.122959	0.054882	0.018146	0.001827
x	0.055312	0.095071	0.155933	0.241011	0.345700	0.455510	0.552683	0.627052	0.680568
y	0.171060	0.215073	0.265774	0.317144	0.358500	0.378851	0.374278	0.350343	0.317371
z	0.773627	0.689856	0.578293	0.441845	0.295800	0.165639	0.073039	0.022605	0.002062
L*	60.0000	60.0000	60.0000	60.0000	60.0000	60.0000	60.0000	60.0000	60.0000
a*	-100.0000	-75.0000	-50.0000	-25.0000	0.0000	25.0000	50.0000	75.0000	100.0000
b*	-100.0000	-75.0000	-50.0000	-25.0000	0.0000	25.0000	50.0000	75.0000	100.0000
R	-63.2553	-38.0477	-14.8426	6.9838	28.0552	48.9952	70.4276	92.9761	117.3231
G	46.7170	41.7155	37.0448	32.5818	28.2034	23.7865	19.2082	14.3453	9.0636
B	127.9629	90.6690	61.4185	39.2362	23.1472	12.1761	5.3480	1.6875	0.0930
R’	0.0000	0.0000	0.0000	6.9838	28.0552	48.9952	70.4276	92.9761	100.0000
G’	46.7170	41.7155	37.0448	32.5818	28.2034	23.7865	19.2082	14.3453	9.0636
B’	100.0000	90.6690	61.4185	39.2362	23.1472	12.1761	5.3480	1.6875	0.0930
CCT	none	none	none	12527 K	5001 K	2504 K	none	none	none
RGB	out-gam	out-gam	out-gam	in-gam	in-gam	in-gam	in-gam	in-gam	out-gam



Matrix Crx

2.364998	-0.896709	-0.468149
-0.515142	1.426371	0.088744
0.005202	-0.014403	1.008898

Matrix Cxr

0.489921	0.310016	0.200063
0.176937	0.812422	0.010641
0.000000	0.009999	0.990301