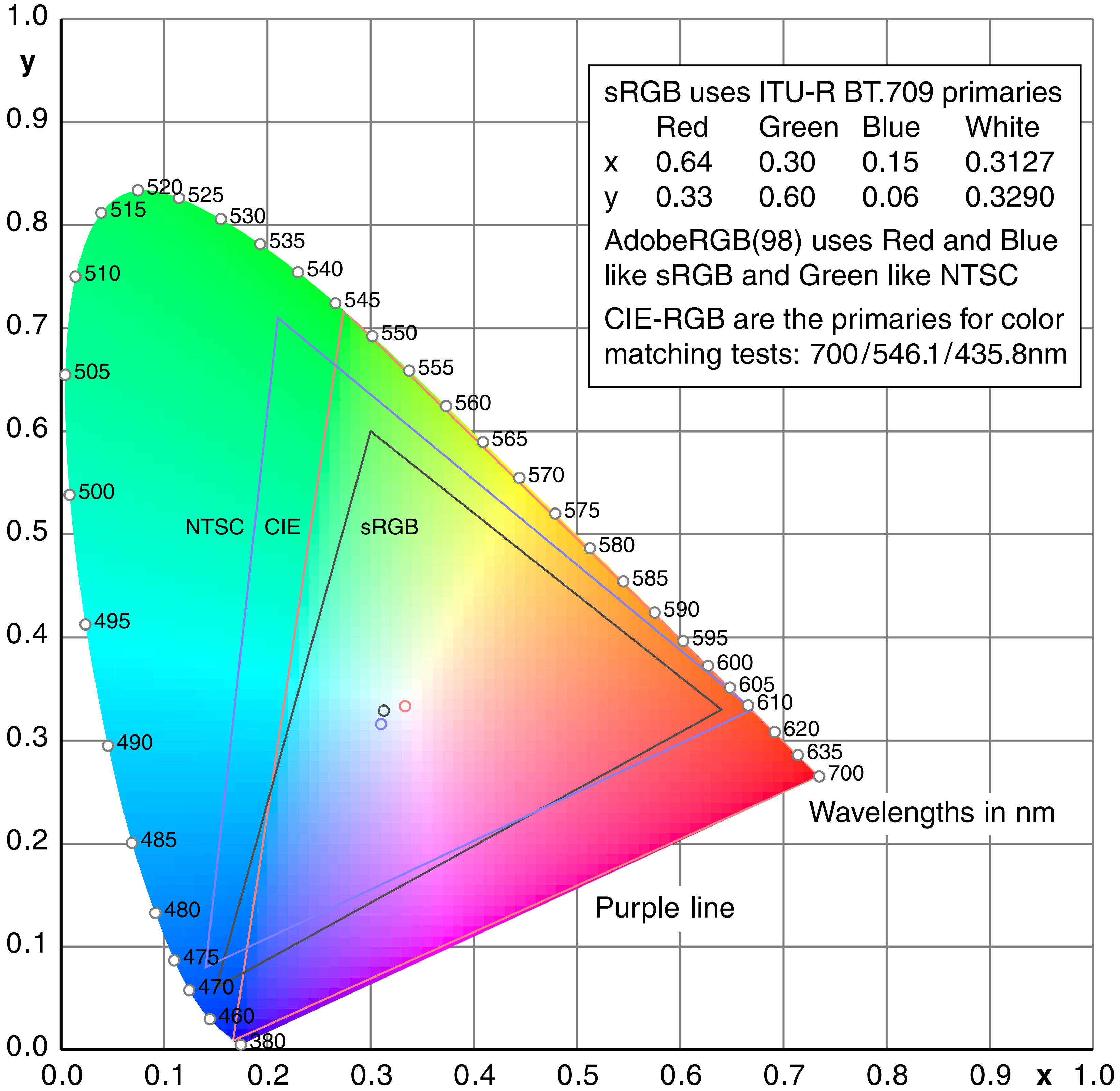
Gernot Hoffmann

CIE Color Space

Contents

1. CIE Chromaticity Diagram (1931)

The CIE chromaticity diagram xyY shows a special projection
of the threedimensional CIE color space XYZ.
This is the base for all color management systems.
The color space contains all distinguishable colors.
Many of them cannot be shown on screens or printed.
The diagram visualizes the concept however.



2. Color Matching and Primaries

Grassmann´s Experiment (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:

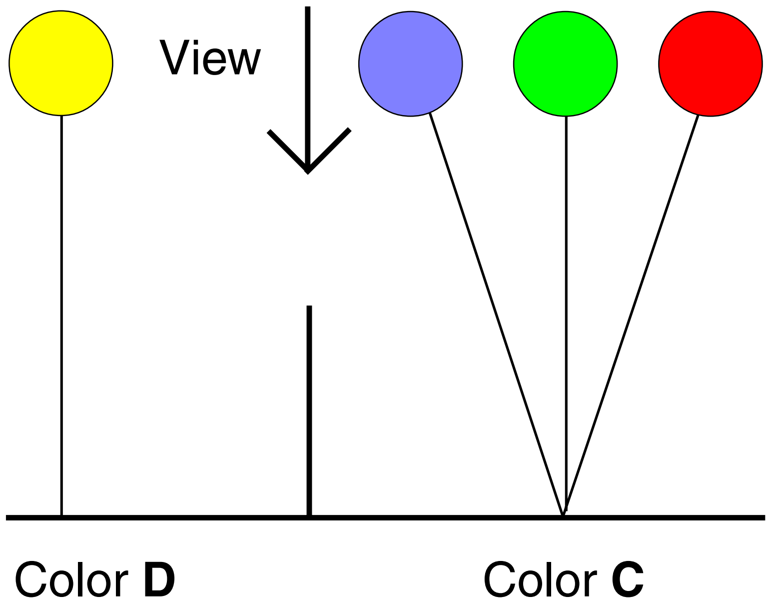
Yellow ⇔ 10**R** +11**G** + 1**B**

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

BlueGreen + 5**R** ⇔ 5**G** + 6**B**

BlueGreen ⇔ - 5**R** + 5**G** + 6**B**

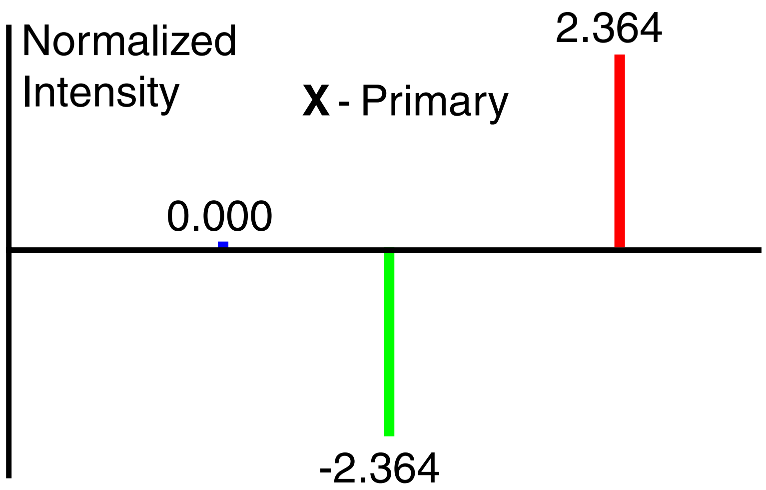
It is generally possible to match a color by three weight factors, but one can be negative.



CIE Standard Primaries (1931)

Primärvalenzen

The primaries are narrow band colors (line spectra) **R** (700 nm), **G** (546.1nm) and **B** (435.8 nm).
The weights R,G,B are transformed by a matrix to non-negative weights X,Y,Z for synthetical or 'imaginary' primaries **X**,**Y**,**Z** (page 6).
The spectra **X**,**Z** have zero integrals, because only **Y** contributes to the luminance.
The **X**-Primary is shown in the right image.



CIE Color Matching Functions

Normalspektralwerte

The functions x, y, z can be understood as weight factors. For a color **C** with dominant wavelength λ_c read in the diagram the three values. Then the color can be mixed by the three Standard Primaries:

$$\mathbf{C} = \bar{x}\,\mathbf{X} + \bar{y}\,\mathbf{Y} + \bar{z}\,\mathbf{Z}$$

Generally we write

$$\mathbf{C} = X\,\mathbf{X} + Y\,\mathbf{Y} + Z\,\mathbf{Z}$$

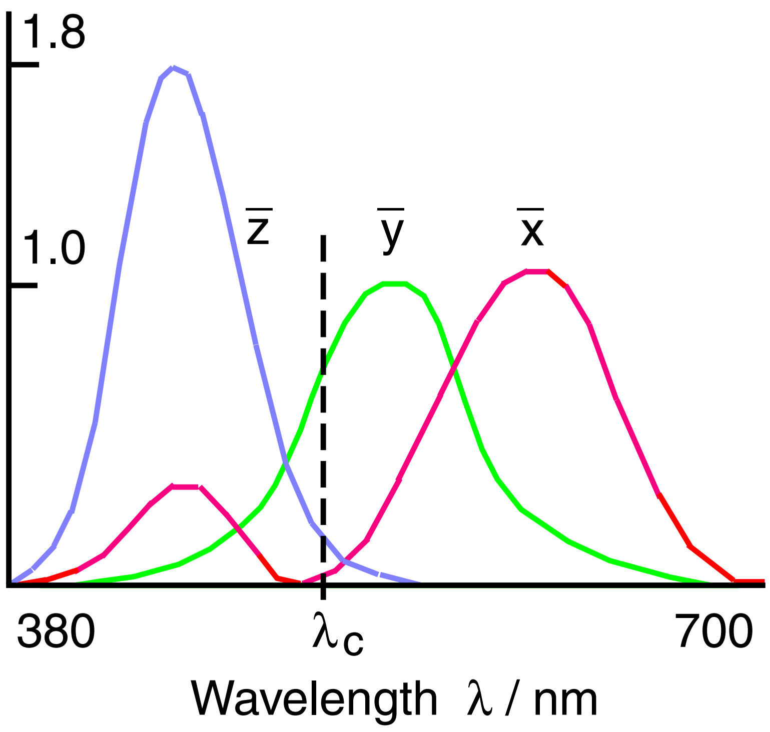
and a given spectral color distribution $P(\lambda)$ delivers the three weight factors by these integrals:

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

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

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

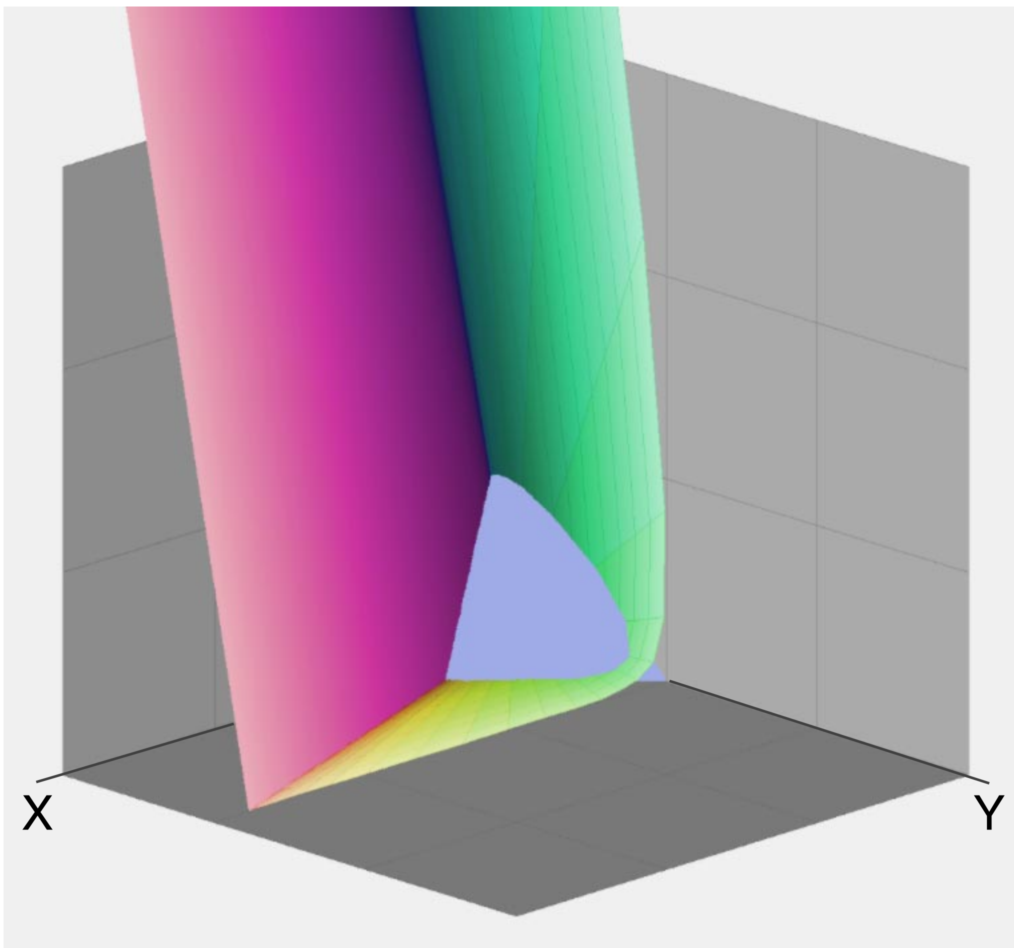
Accurate graphics [16],[17]



3. Color Volumes and Chromaticity

Color Volume

A color, based on any set of primaries **R**, **G**, **B**, can be generated if in R,G,B at least one is positive and none is negative.
A color is matchable if X,Y,Z are positive in a boundary surface, which is defined by human perception.
Not all matchable colors can be generated by the CIE Standard Primaries.
The matchable colors form the volume **C** in the coordinate system X,Y,Z.
We can embed an RGB cube. The color volume of the cube is a subset of **C**. It will appear distorted.
Page 8 shows a larger computer graphic.



Volume of matchable colors

Chromaticity Values

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

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

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

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

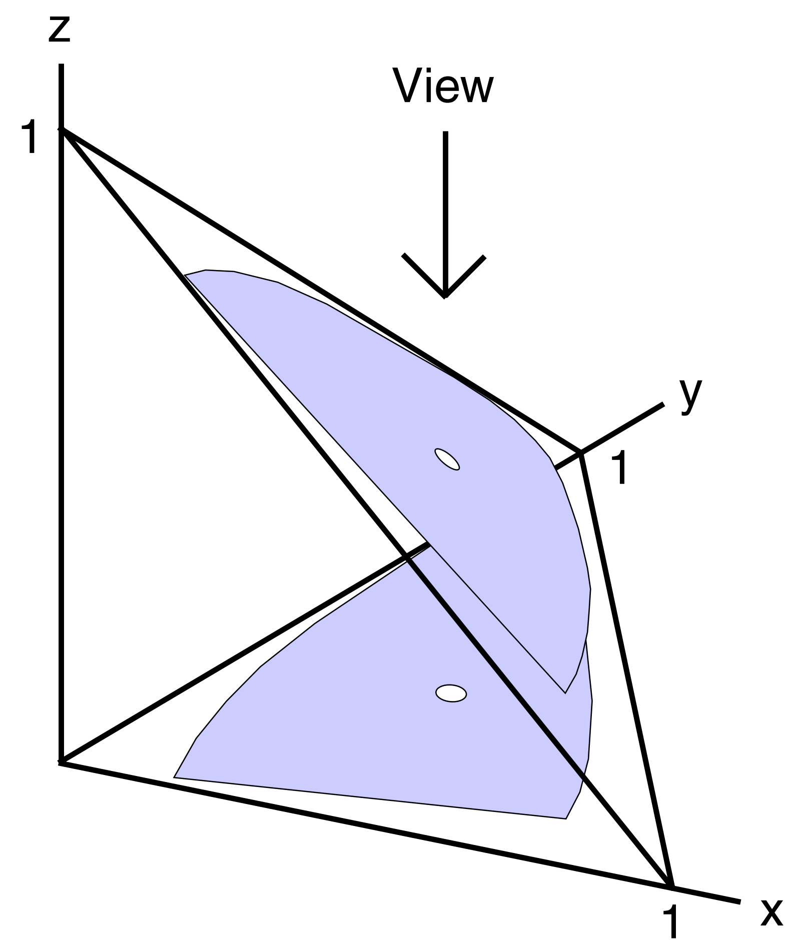
Obviously we have $x + y + z = 1$. All the values are on the triangle plane, projected by a line through X,Y,Z and the origin, if we draw X,Y,Z and x,y,z in one diagram.
The vertical projection onto the xy-plane is the chromaticity diagram.
To reconstruct a color triple X,Y,Z from the chromaticity values x,y we need an additional information, the luminance Y.

$$z = 1 - x - y$$

$$X = Yx / y$$

$$Z = Yz / y$$

All visible (matchable) colors which differ only in luminance map to the same point in the chromaticity diagram.



Chromaticity plane and projection

4. RGB to XYZ Conversion

RGB to XYZ Transformation

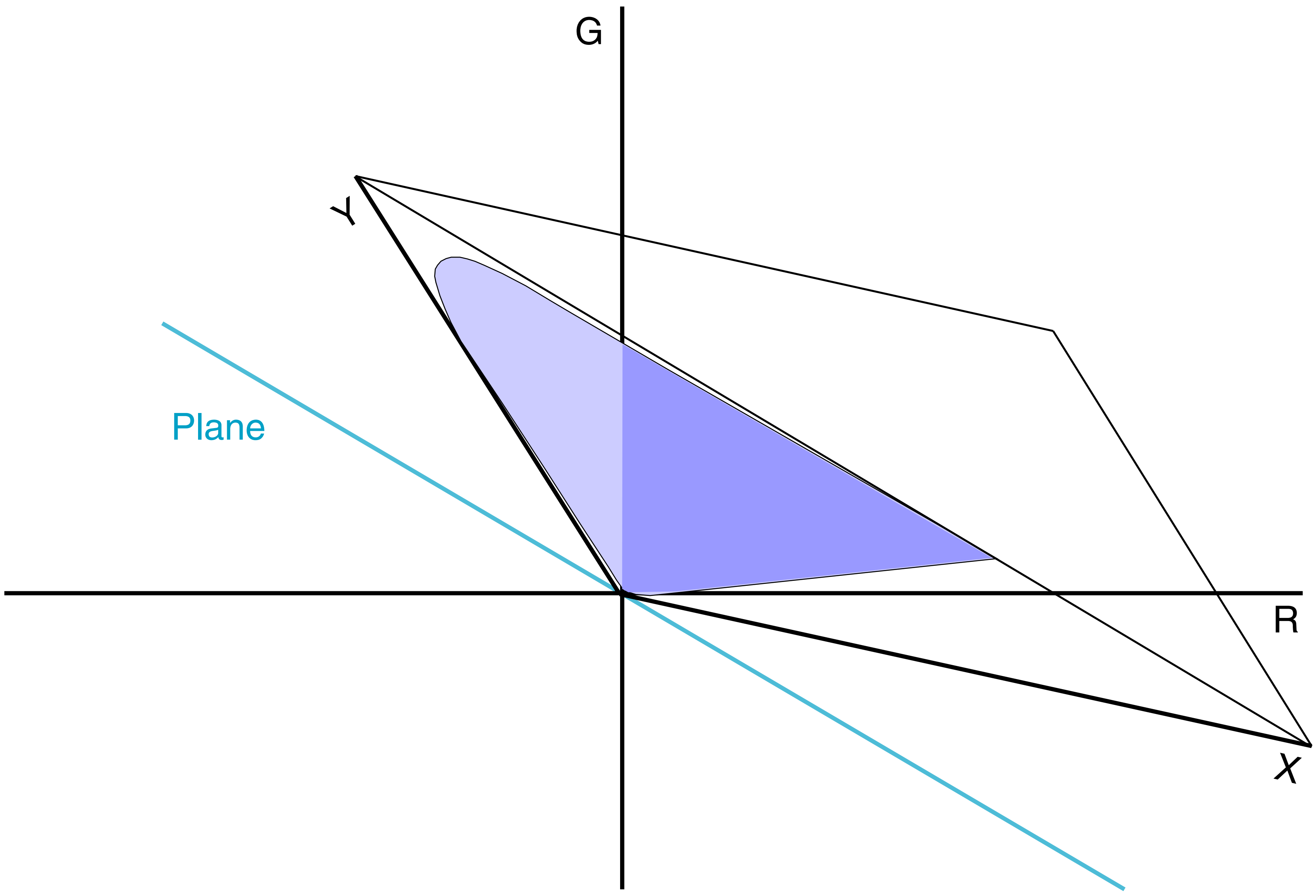
$$
\begin{aligned}
X &= +0.49000\,R + 0.31000\,G + 0.20000\,B \\
Y &= +0.17697\,R + 0.81240\,G + 0.01063\,B \qquad (1) \\
Z &= +0.00000\,R + 0.01000\,G + 0.99000\,B
\end{aligned}
$$

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

The matrices are inverse to each other.
The first was extracted from Wyszecki [3] and proved by Hunt´s Measuring Colors [1].

Visualization for two Dimensions

The contour of the shaded area is symbolically the locus of color matching numbers R,G,B and X,Y,Z. The dark *triangle* shows colors which can be matched by positive R,G,B. The corners are located at the CIE Primaries (page 6). The match refers to colors with a dominant wavelength.
Opposed to R,G,B, all coordinates X,Y,Z for the *whole* shaded area are positive.
Here the R,G,B -Diagram is drawn orthogonal. In the next step the X,Y,Z - Diagram will be drawn orthogonal and R,G,B is usually not shown (previous pages).
An arbitrary plane divides the space into two halfspaces. The only necessary condition for the coordinate transformation is, that both sets are completely in the same halfspace. Some degrees of freedom are left. Now, X,Y,Z are chosen especially so that the luminance depends on Y only. X and Z dont´t contribute. This is explained on the next page. Furtheron, Y is scaled according to the luminous efficiency function V(λ) , [3].



5. CIE Primaries

Spectra **R**, **G**, **B** and **X**,**Y**, **Z**

The Standard Primaries **R**,**G**,**B** are monochromatic stimuli.
Mathematically they are delta functions with a well defined area.
In the diagram, the height represents the contribution to the luminance.
The ratio is 1: 4.5907 : 0.0601.

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

An example:

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

$$\begin{aligned}\mathbf{X} = &+ 2.36461 \cdot 1.0000\ \mathbf{R}\\ &- 0.51517 \cdot 4.5907\ \mathbf{G}\\ &+ 0.00520 \cdot 0.0601\ \mathbf{B}\end{aligned}$$

$$\begin{aligned}\mathbf{X} = &+ 2.36461\ \mathbf{R}\\ &- 2.36499\ \mathbf{G}\\ &+ 0.00031\ \mathbf{B}\end{aligned}$$

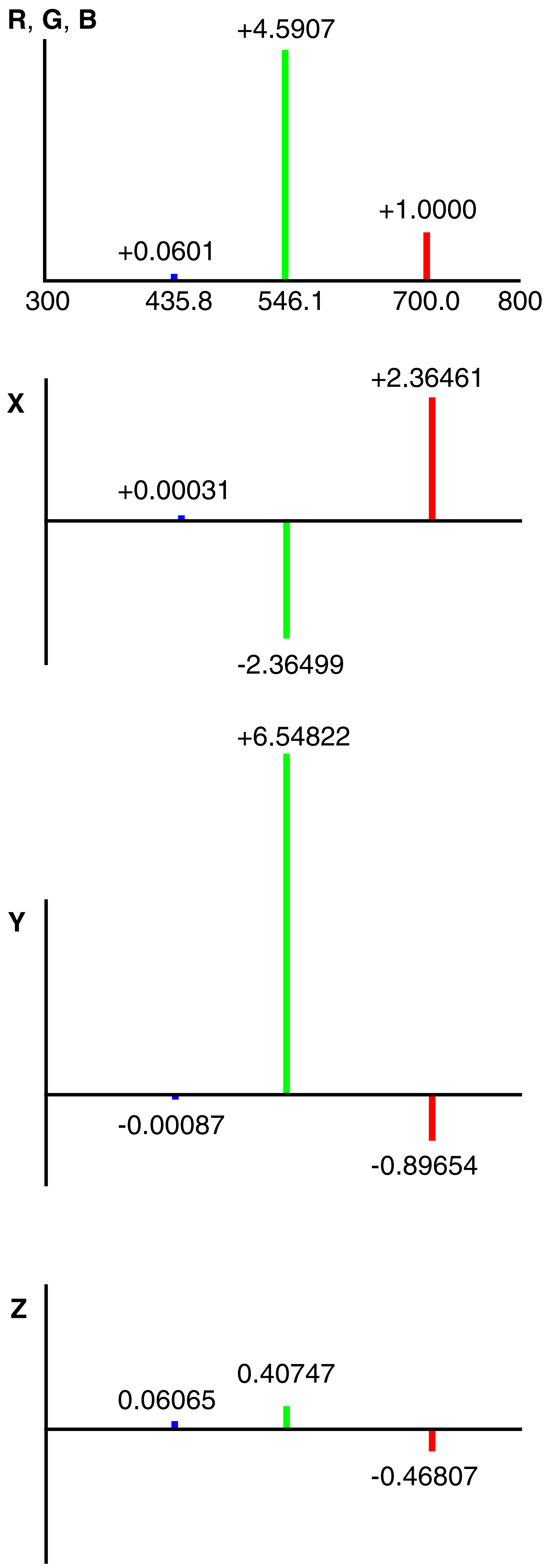
The primaries **X**,**Y**,**Z** are sums of delta functions. **X** and **Z** don´t contribute to the luminance. The integral are zero, here represented by the sum of the heights.
The luminance is defined by **Y** only.

Any color

$$\mathbf{C} = X\ \mathbf{X} + Y\ \mathbf{Y} + Z\ \mathbf{Z} = R\ \mathbf{R} + G\ \mathbf{G} + B\ \mathbf{B}$$

is visible, if in R,G,B at least one is positive and none negative.
Because R,G,B and X,Y,Z are related by the matrix transformation (1), the „gamut“ or set of all possible colors in X,Y,Z for these Standard Primaries is confined by three planes through the origin. In x,y it´s a triangle, the corners on the horseshoe contour, primaries as above.

This has nothing to do with color matching. In color matching experiments negative values R, G, B are allowed. The set of matchable colors defines the horseshoe area. This is a result of the features of vision. Some matchable colors cannot be generated by the Standard Primaries, other light sources are required.



6. Visibility for Positive Weights

Pyramide of visibility, based on CIE primaries

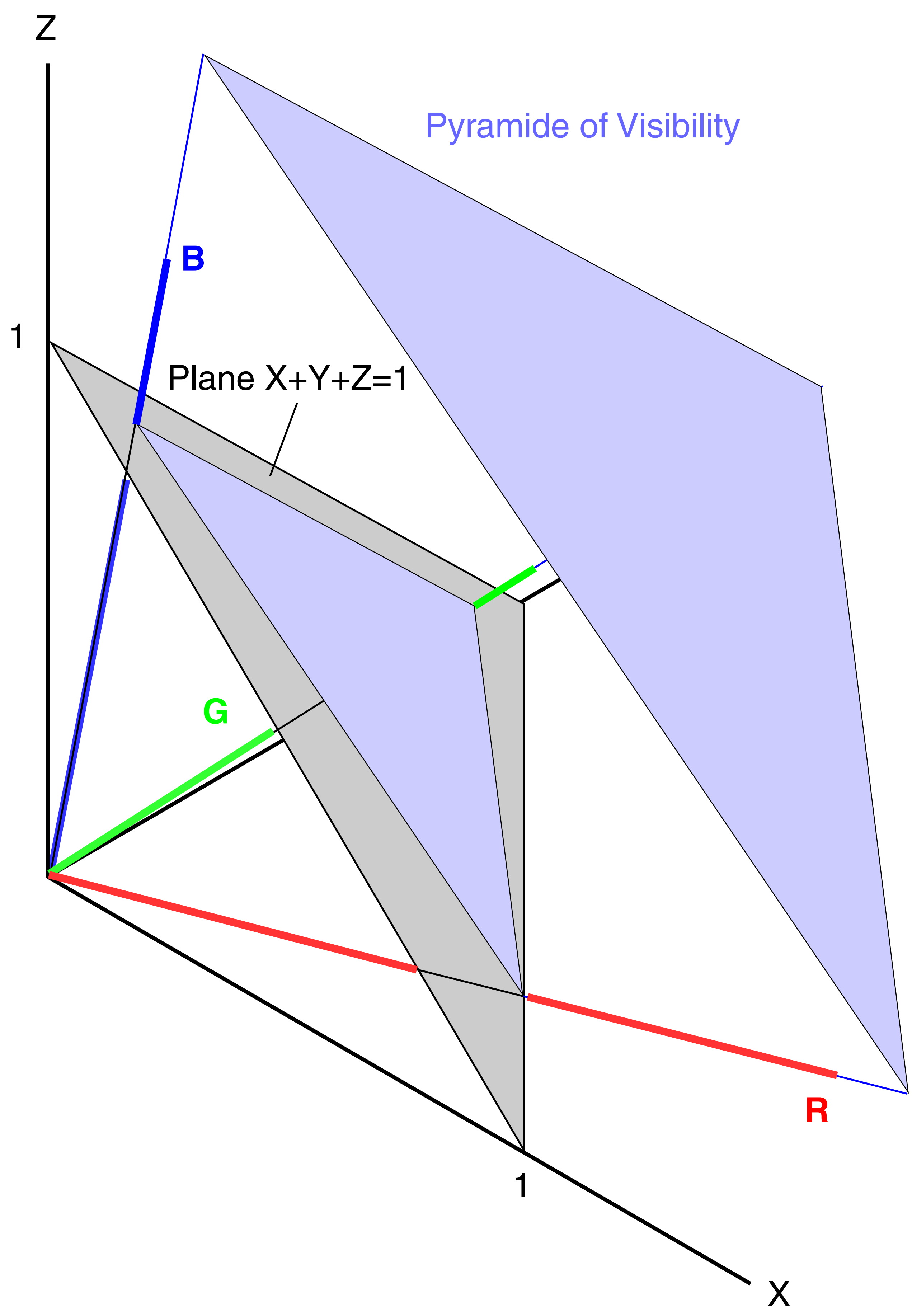
The pyramide of visibility for positive weights (gamut for CIE primaries) is constructed from three planes, three equations (2) with R,G=0,0, R,B=0,0, G,B=0,0.
The three planes are intersected with the gray plane, the equation

$1 = X + Y + Z \quad (3)$.

This means: solve the three sets of linear equations

(2.1)+(2.2)+(3) R=0, G=0: $X_b = 0.16658$, $Y_b = 0.00885$, $Z_b = 0.82457$
(2.1)+(2.3)+(3) R=0, B=0: $X_g = 0.27375$, $Y_g = 0.71741$, $Z_g = 0.00883$
(2.2)+(2.3)+(3) G=0, B=0: $X_r = 0.73467$, $Y_r = 0.26533$, $Z_r = 0.00000$

The pyramide reaches to infinity, another arbitrary intersection is marked blue too.
Three base vectors **R**, **G**, **B** with arbitrary lengths are shown. The relation of the lengths defines the whitepoint. This is important for other RGB gamuts.

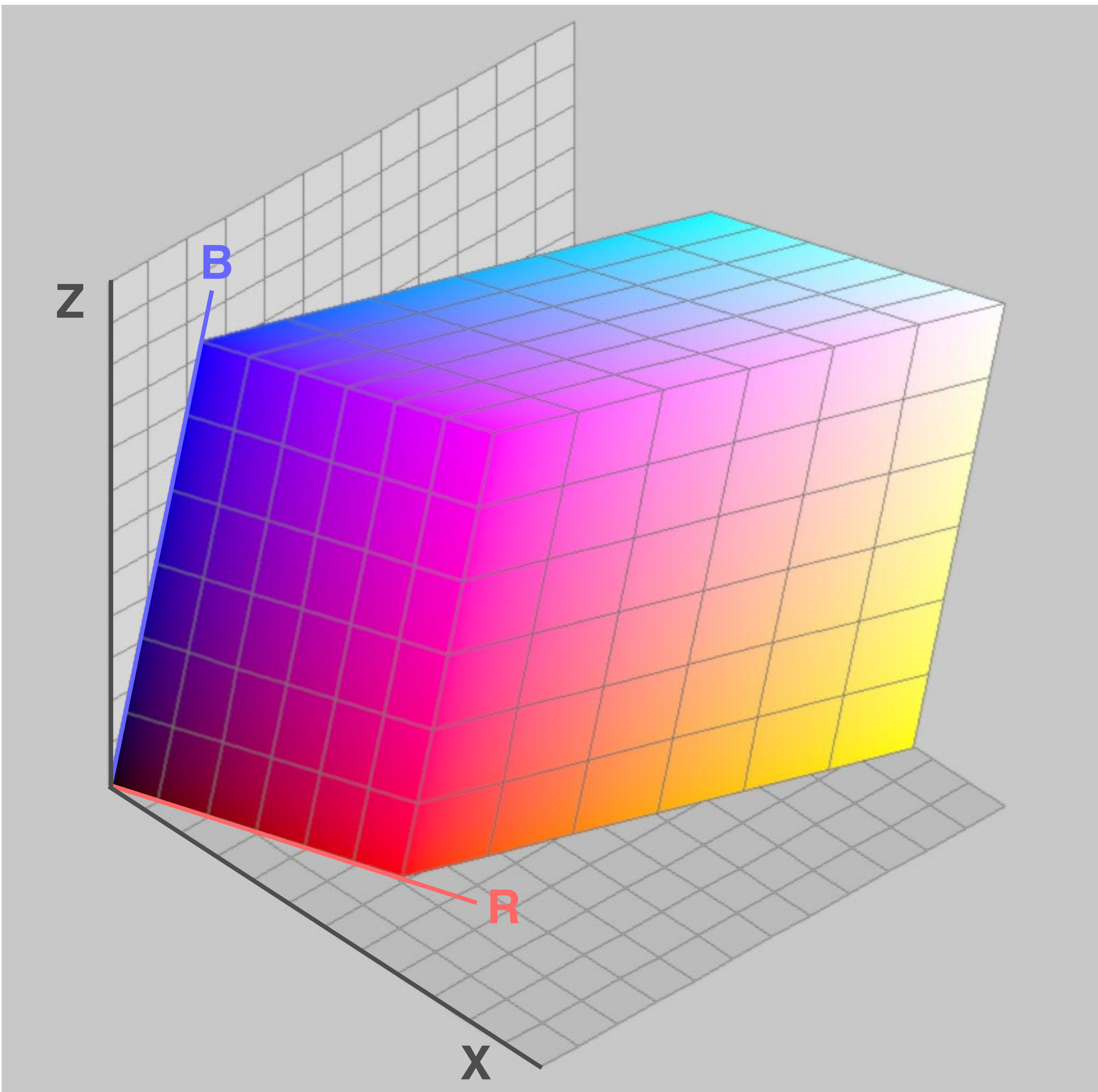


7. RGB Basis Vectors in XYZ

RGB Cube in XYZ Coordinates

This is a computer graphic, everything defined by numbers and accurate transformations.
Parallel perspective, approximately isometric in X,Y,Z .
The image has been inverse gamma compensated for gamma = 1.7 .
This delivers the best appearance for gamma = 2.2 screens.
Because of JPEG compression, the grids cannot be shown accurately in PDF.

This is a geometry transformation. It is not a color transformation (refer to next page).
The three axes of the RGB cube are the base vectors in XYZ for an RGB color system.



8.1 Color Space Visualization

XYZ Color Space **C**

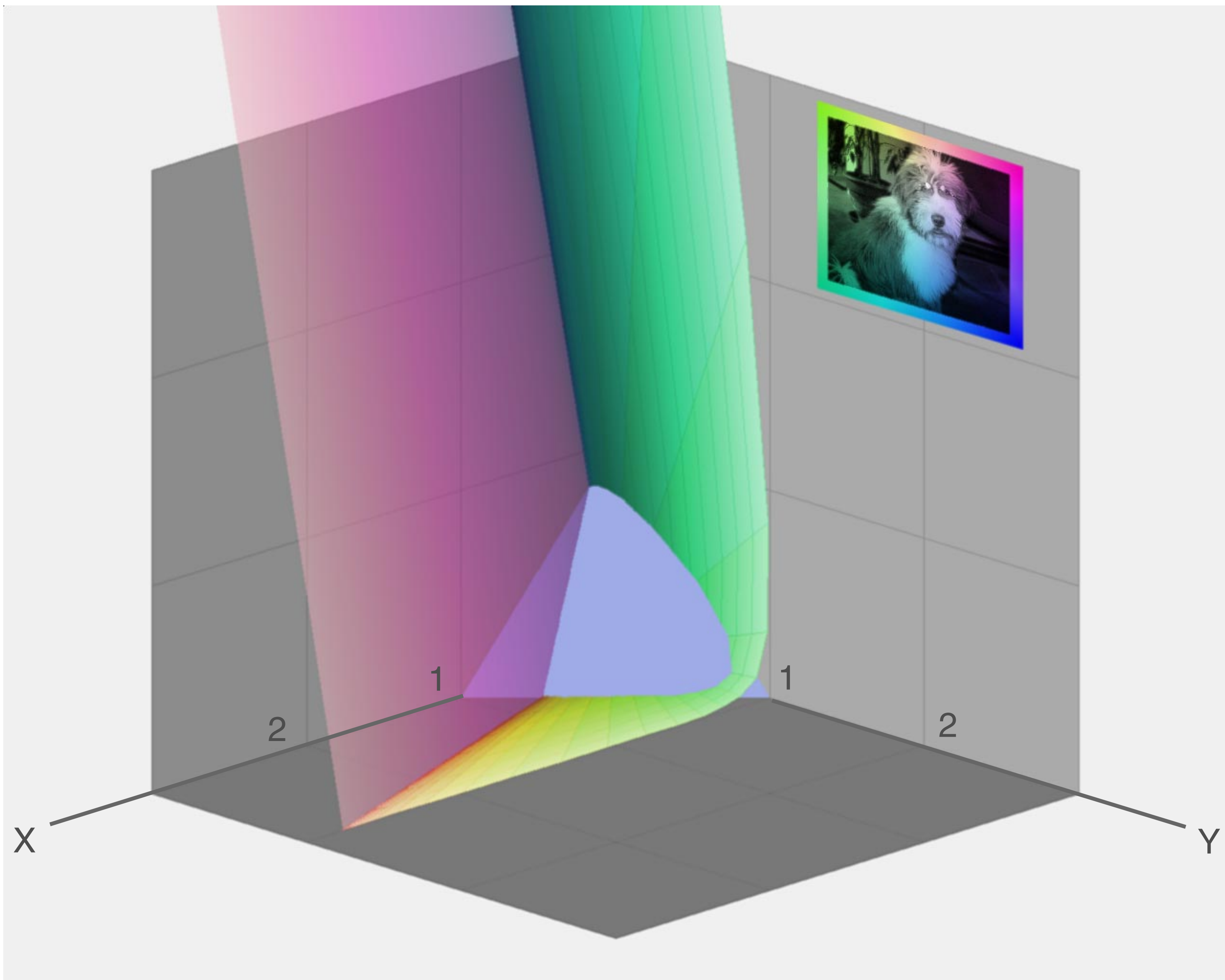
This is also a computer graphic, everything defined by numbers and accurate transformations and a few applications of image processing.
The contour of the horseshoe is mapped to XYZ for luminances Y = 0...1 .

$$z = 1 - x - y$$
$$X = Y\,x\,/\,y$$
$$Z = Y\,z\,/\,y$$

The purple surface is shown transparent. All colors were selected with respect to readabilty. The colors are not correct, this is anyway impossible. More important is here the geometry .
The volume **C** is confined by the color surface (pure spectral colors), the purple plane and the plane Y = 1.

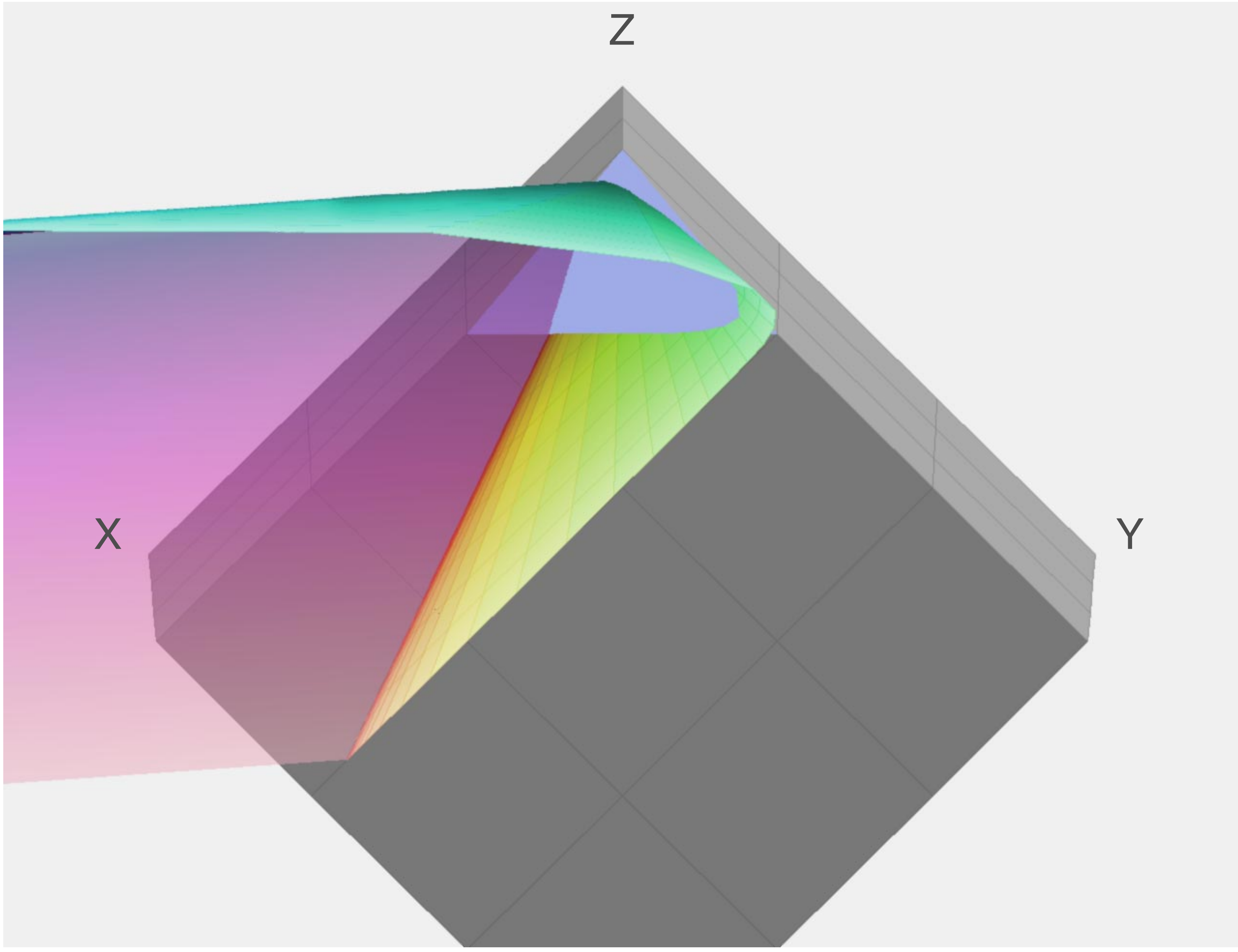
The regions with small values Y appear extremely distorted - near to a singularity.
For Blue, high values Z are necessary to match a color with specified luminance Y = 1.
This can be shown by an example:
For λ = 380 nm we find x,y,z in tables (Wyszecki) and calculate then X,Y,Z:

x = 0.1741	y = 0.0050	z = 0.8290
X = 34.82	Y = 1.00	Z = 165.80



8.2 Color Space Visualization

XYZ Color Space **C**, Bird´s View

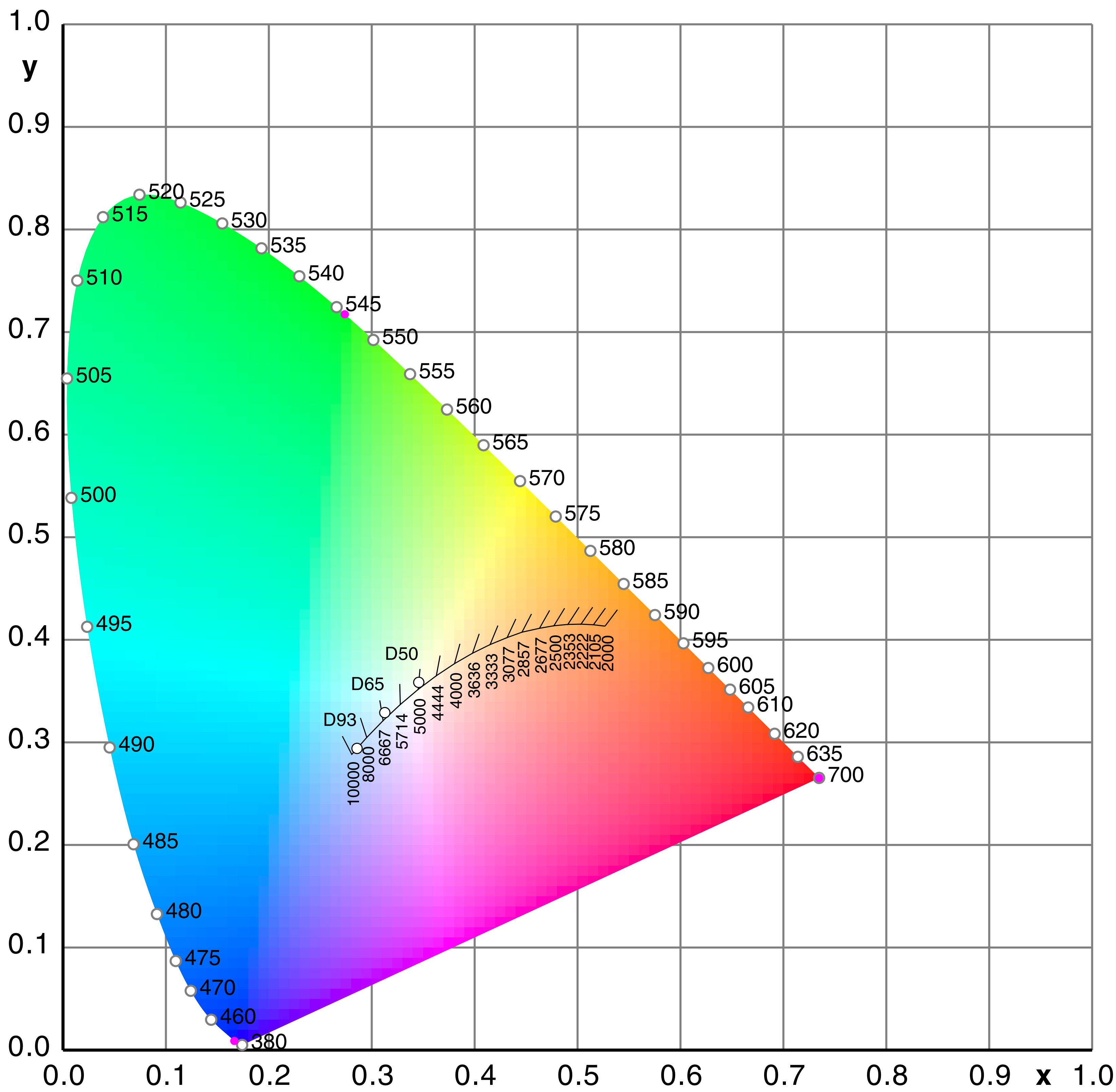


9. Color Temperature and White Points

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



10.1 Color Space Calculations

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 is shown on the pages 7, 8 .

Given are the coordinates of the primaries in CIE xyY and 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 absolute 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)
$$\begin{aligned} x &= X/L \\ y &= Y/L \\ z &= Z/L \end{aligned}$$

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

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

V is the luminance, according to the luminous efficiency function V(λ) in [3].
This cannot be called Y here. V defines the absolute luminance of the stimulus.

(8)
$$\begin{aligned} X &= V\,x/y \\ Y &= V \\ Z &= V\,z/y \end{aligned}$$

Basis vectors for the primaries in XYZ:

(9)
$$\begin{aligned} \mathbf{R} &= L\,(x_r, y_r, z_r)^T \\ \mathbf{G} &= L\,(x_g, y_g, z_g)^T \\ \mathbf{B} &= L\,(x_b, y_b, z_b)^T \end{aligned}$$

White Point in XYZ:

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

Set of scale factors for the White Point correction:

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

10.2 Color Space Calculations

For the White Point correction, the basis vectors **R**,**G**,**B** are scaled by u,v,w. This doesn´t change their coordinates in xyY .The mapping from XYZ to xyY is a central projection.

$$(12) \qquad \mathbf{X} = L(x,y,z)^T = r\,u\,\mathbf{R} + g\,v\,\mathbf{G} + b\,w\,\mathbf{B}$$

For the White Point we have r = g = b = 1

$$(13) \qquad \mathbf{W} = L(x_w,y_w,z_w)^T = L\,u\,(x_r,y_r,z_r)^T + L\,v\,(x_g,y_g,z_g)^T + L\,w\,(x_b,y_b,z_b)^T$$

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

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

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:

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

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

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

These linear equations are solved by Cramer.

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

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

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 $\mathbf{C}_{rx}$.

$$(20) \qquad \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = V \begin{bmatrix} u\,x_r/y_w & v\,x_g/y_w & w\,x_b/y_w \\ u\,y_r/y_w & v\,y_g/y_w & w\,y_b/y_w \\ u\,z_r/y_w & v\,z_g/y_w & w\,z_b/y_w \end{bmatrix} \begin{bmatrix} r \\ g \\ b \end{bmatrix}$$

$$(21) \qquad \mathbf{X} = V\,\mathbf{C}_{xr}\,\mathbf{r}$$

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

10.3 Color Space Calculations

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
```

10.4 Color Space Calculations

ITU-R BT.709 Primaries and D65 [9].
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

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

$$X = C_{xr} R$$

$$R = C_{rx} X$$

10.5 Color Space Calculations

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
```

$$X = C_{xr} R$$

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

$$R = C_{rx} X$$

10.6 Color Space Calculations

CIE Primaries and White Point [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
```

$X = C_{xr} R$

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

$R = C_{rx} X$

10.7 Color Space Calculations

NTSC Primaries and White Point [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
```

$$R = C_{rx} X$$

10.8 Color Space Calculations

NTSC Primaries and White Point [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

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

$Y = C_{yr} R$

$R = C_{ry} Y$

10.9 Color Space Calculations

NTSC Primaries and White Point [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.
```

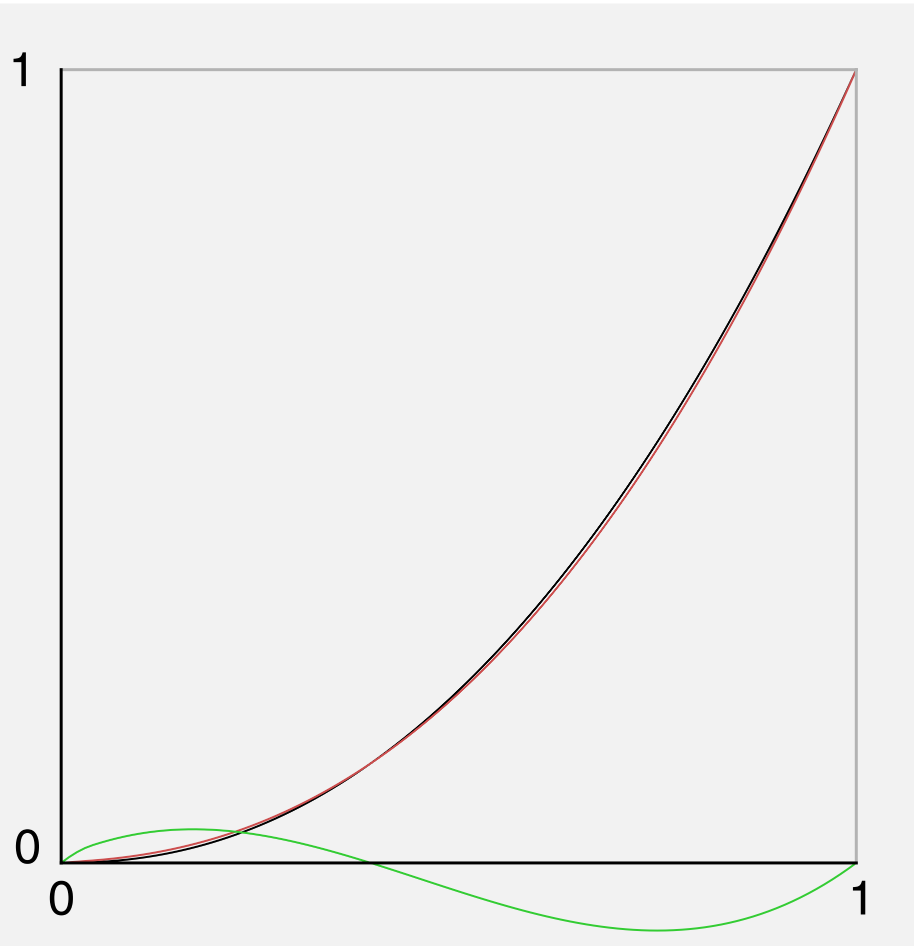
```
Matrix Cyr
Y    0.2990      0.5870      0.1140       Note
Cb  -0.1687     -0.3313      0.5000       This is a linear conversion, as used for JPEG
Cr   0.5000     -0.4187     -0.0813       In TV systems the conversion is different

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
```

11. 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 10 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:

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

12. Barycentric Coordinates

The corners of a triangular gamut, e.g. for a monitor are described in CIE xyY by three corners **R**,**G**,**B** which have two components x,y each.
A color **C** is described either by 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).

Using **R**,**G**,**B** as 2D base vectors we get two equations for three unknowns:

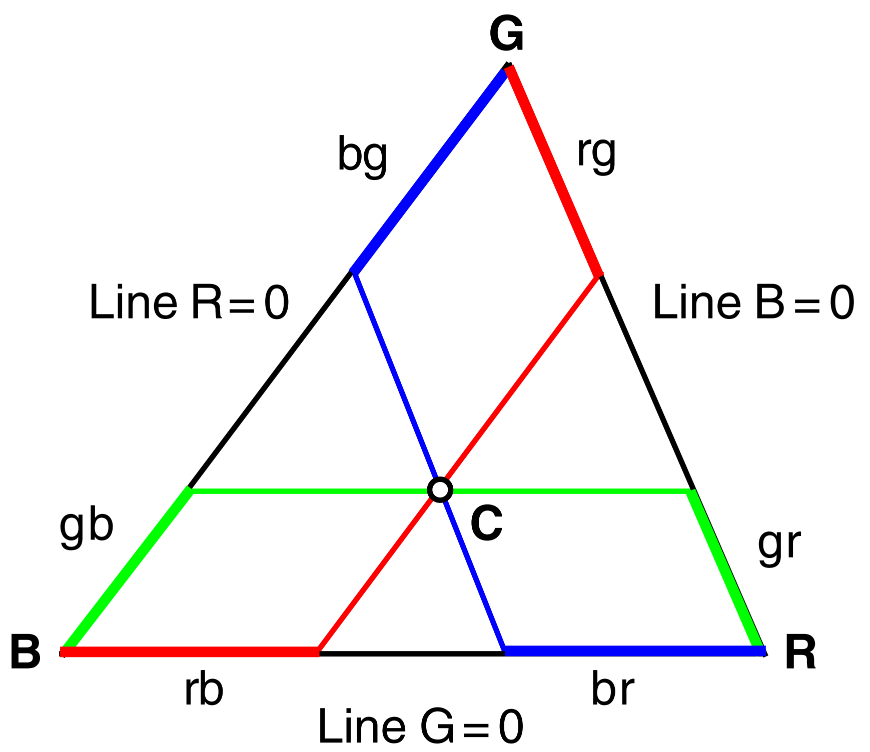
$$\mathbf{C} = R\,\mathbf{R} + G\,\mathbf{G} + B\,\mathbf{B}$$

The third equation is the plane equation

$$1 = R + G + B\,.$$

This system of three linear equations has to be solved for R, G, B:

$$\begin{aligned} R R_x + G G_x + B B_x &= C_x \\ R R_y + G G_y + B B_y &= C_y \\ R + G + B &= 1 \end{aligned}$$



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

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

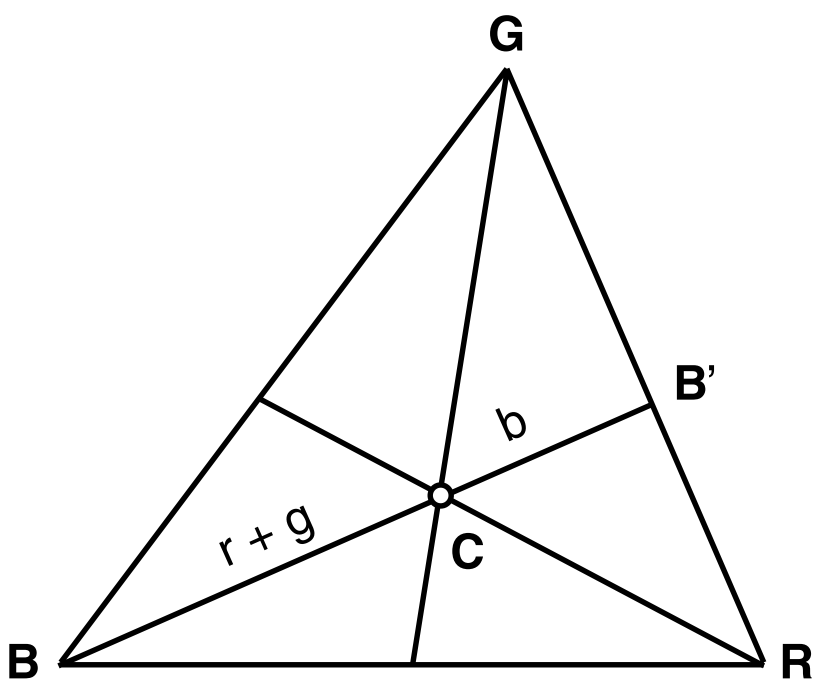
$gr = G\,\underline{\mathbf{GR}}$

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

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

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

Underline means length of ..



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

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

13.1 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

13.2 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

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

Gernot Hoffmann

July 06 / 2004
Website
Load Browser / Click here