RGB色彩空间的定义

RGB色彩空间是三原色的色彩空间，RGB色彩空间中的颜色是由三原色的分量来定义。如果三原色的色彩未定义，则RGB色彩空间是相对色彩空间。但当RGB的三原色红色、绿色和蓝色在绝对色彩空间定义后，此时RGB色彩空间就变成绝对色彩空间。比如未进行色彩校正的相机，生成的RGB图就是是相对色彩空间，图片里颜色的RGB三分量的数据与相机的感光系统相关。而色彩校正的目的便是建立这个相对色彩空间与绝对色彩空间之间的关系。

RGB色彩空间是一系列色彩空间，后面主要讨论其中的绝对色彩空间，包括sRGB， Adobe RGB等。这些色彩空间进行绝对化的主要区别是在三原色的定义和线性化的标准。

三原色的定义

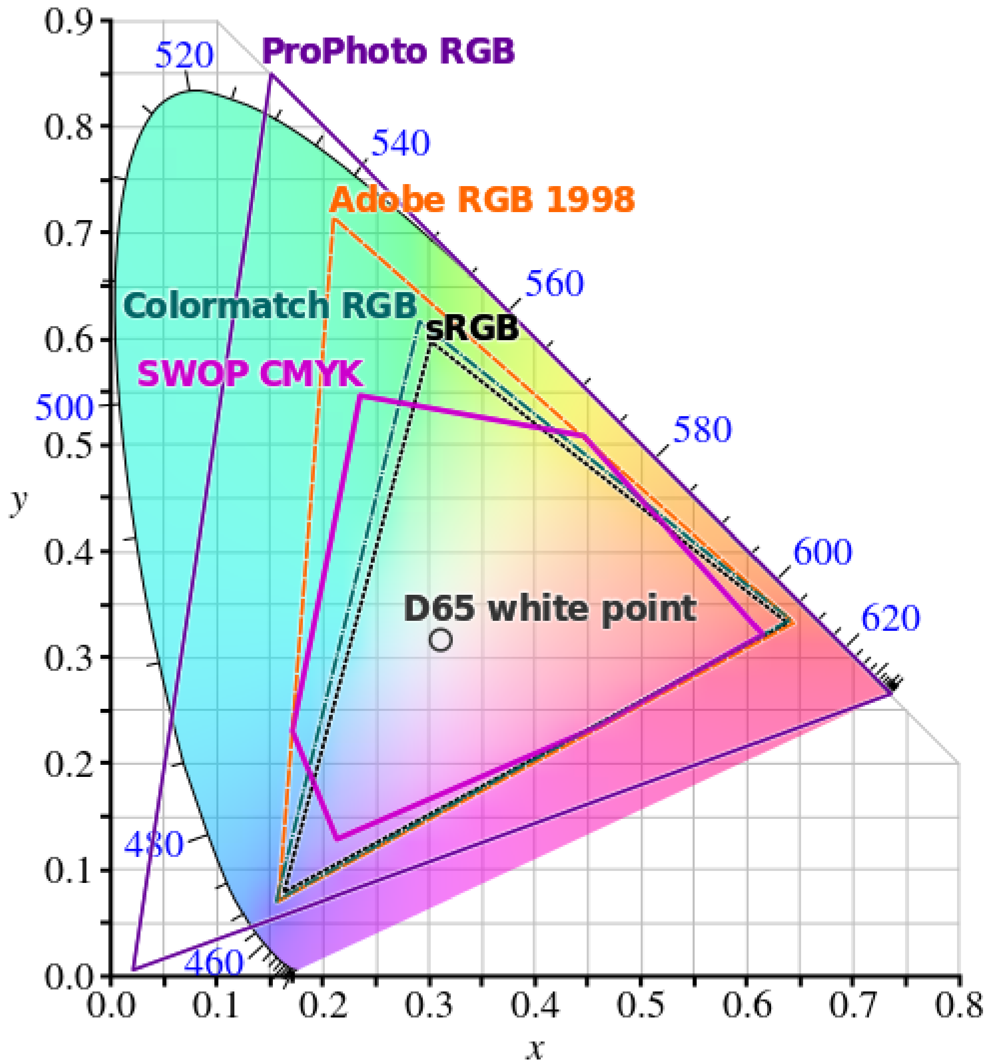
在定义色彩空间时，要先选定白点和观察者。然后在白点的基础上构建的XYZ空间上选定三原色。通常三原色选择会忽略亮度，选择在色度空间内选取。选取白点、观察者和三原色后，RGB空间便定义下来了。

一些RGB色彩空间的三原色的定义如下[1-9]，其中观察者均为默认观察者，因为未列入表格：

	White Point	xR	yR	xG	yG	xB	yB
sRGB	D65	0.64	0.33	0.30	0.60	0.15	0.06
Adobe RGB	D65	0.64	0.33	0.21	0.71	0.15	0.06
Wide Gamut RGB	D50	0.7347	0.2653	0.1152	0.8264	0.1566	0.0177
ProPhoto RGB	D50	0.734699	0.265301	0.159597	0.840403	0.036598	0.000105
DCI P3	D65	0.68	0.32	0.265	0.69	0.15	0.06
Apple RGB	D65	0.625	0.34	0.28	0.595	0.155	0.07
REC. 709	D65	0.64	0.33	0.30	0.60	0.15	0.06
REC. 2020	D65	0.708	0.292	0.17	0.797	0.131	0.046

由于在色度图上，三原点构成的三角形涵盖了以这三原点组合得到的所有颜色，也被称为**色域**。因此，之前定义了RGB图三原点的在色域图上的位置，可以在色域图上画出各RGB空间图的色域[1]。

sRGB是目前计算机显示器最常见的色彩空间，然而色域有限；而Adobe RGB，Wide Gamut RGB等色彩空间的色域相对就大得多。



线性化与反线性化

通过定义了白点、观察者和三原色后，此时线性化的RGB的色彩空间已经定了下来。此时色彩空间的颜色各分量值与亮度成正比，但却不符合人类的视觉感受。人类对亮度的感知却是非线性的，遵循近似的幂函数，对较暗色调之间的相对差异比起亮色调之间的相对差异更敏感。因此，通常需要对色彩空间使用**伽马校正**来符合人类视觉感受。

伽玛校正[10]由以下幂律表达式定义：

$$V_{out} = AV_{in}^{\gamma}$$

由于早期显示设备阴极射线管（CRT）显示器，光强度随电子枪电压非线性变化，因此伽玛校正不仅要用于补偿阴极射线管（CRT）显示器的非线性性，还要修正人类视觉感受。即便目前显示设备发生变化，伽马校正的标准却也定了下来。目前，绝大多数的RGB色彩空间的解码伽马值为2.2，也就是说，从线性化RGB色彩空间变成适合人眼亮度感受的非线性RGB色彩空间的关系是：

$$C_n = C_l^{1/\gamma}, \quad \gamma = 2.2$$

其中C_l为线性RGB色彩空间的颜色通道，而C_n为非线性RGB色彩空间通道，C_l与C_n一般取值在[0,1]内。

实际做色彩转换的时候，有可能计算值超出[0,1]，为了避免计算出虚数值，通常将伽玛校正的图像关于原点做对称，或者说：

从非线性RGB空间转换成线性RGB空间

$$C_l = C_n^{\gamma}, \qquad C_n \geq 0$$
$$C_l = -(-C_n)^{\gamma}, \qquad C_n < 0$$

从线性RGB空间转换成非线性RGB空间

$$C_n = C_l^{1/\gamma}, \qquad C_l \geq 0$$
$$C_n = -(-C_l)^{1/\gamma}, \qquad C_l < 0$$

以上的γ值与所处的空间相关，一般取为2.2。

由于线性RGB空间转换成非线性RGB空间的公式中，在原点附近的导数趋近无穷大，因此，为了数学计算的方便，部分色彩空间（如sRGB，REC.709等）在原点附近一定范围内会使用直线代替，在直线段和幂函数段的连接点处保持连续可导。

为了保证校正前后的色彩空间单通道从[0,1]映射到[0,1]，通常给的分段函数为[2]：

$$C_l = C_n/\phi, \qquad 0 \leq C_n \leq K_0$$
$$C_l = (\frac{C_n + a}{1 + a})^{\gamma}, \qquad C_n > K_0$$

为了保持连续可导，则在K_0处，函数值连续且左导数和右导数相等：

$$(\frac{K_0 + a}{1 + a})^{\gamma} = \frac{K_0}{\phi}$$
$$\gamma(\frac{K_0 + a}{1 + a})^{\gamma - 1} = \frac{1}{\phi}$$

这意味着K_0, a, ϕ中只有一个独立的量。

通常用a表示其他量，有：

$$K_0 = \frac{a}{\gamma - 1}$$
$$\phi = \frac{(1 + a)^{\gamma}(\gamma - 1)^{\gamma - 1}}{a^{\gamma - 1}\gamma^{\gamma}}$$

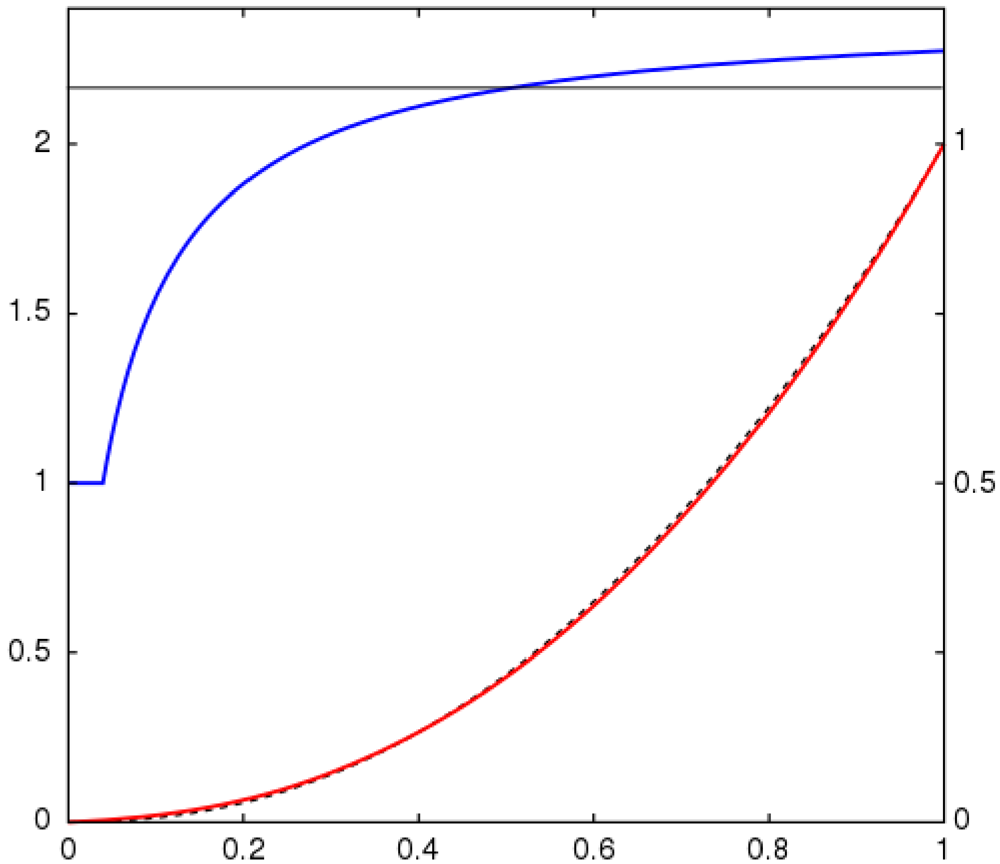
此外，为了使得反线性化函数表达简洁，通常引入以下变量：

$$\alpha = a + 1$$
$$\beta = K_0 \phi$$

则从线性RGB到非线性的过程为：

$$C_n = \phi C_l, \qquad 0 \leq C_n \leq \beta$$
$$C_l = \alpha C_l^{1/\gamma} - a, \qquad C_n > \beta$$

下图便是sRGB线性化的变换曲线，红色是线性化函数，蓝色是导函数，红色下方的黑色虚线是$\gamma = 2.2$时的伽马校正曲线。由于原点附近的值被直线替代，使得幂函数部分曲线会有所平移。sRGB为了取到平均γ为2.2的结果，实际计算使用的γ值为2.4。此外，由于截断误差的原因，实际使用的sRGB转换函数在K_0附近会有所不连续。



在色彩校正中，有可能出现小于0的输入值。为了保证函数的输出值仍在实数域，同样将函数关于原点做中心对称。

一些常见的RGB色彩空间线性化参数：

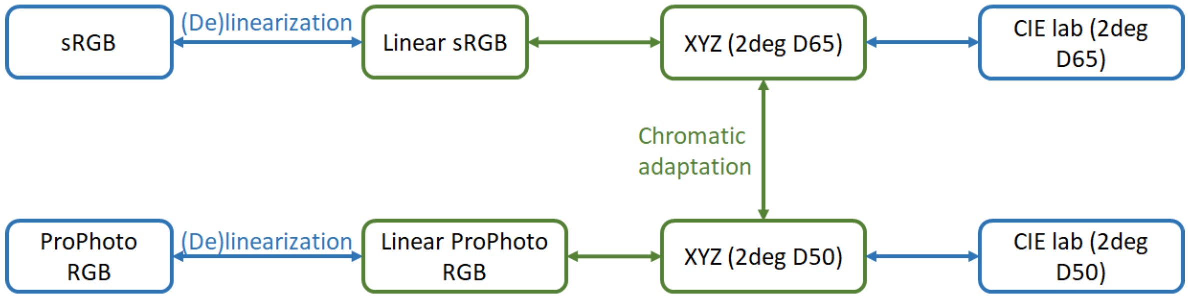
	γ	a
sRGB	2.4	0.055
Adobe RGB	2.2	/
Wide Gamut RGB	2.2	/
ProPhoto RGB	1.8	/
DCI P3	2.6	/
Apple RGB	1.8	/
REC. 709	1/0.45	0.099
REC. 2020	1/0.45	0.0993

RGB色彩空间的转换

在色彩校正中，需要将RGB色彩空间转换成CIE lab色彩空间。下图以sRGB和ProPhoto RGB为例，展示色彩空间转换的过程。其中绿色的箭头表示线性变换，而蓝色表示非线性变换。绿色的方框代表的色彩空间之间可以通过线性变换来实现。

整个拟合过程所有的RGB色彩空间数值都必须做归一化，即取值在[0, 1]之间。

CS Conversion



线性RGB转换成同白点XYZ

当三原色和白点已定时，线性RGB空间就已定下了。由于XYZ空间还需要定义白点才能表示绝对色彩，因此我们可以把RGB空间表示为白点定义在RGB空间白点上的XYZ色彩空间。

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = M_{RGBL2XYZ} \begin{bmatrix} R_l \\ G_l \\ B_l \end{bmatrix}$$

其中转换矩阵可以通过三原色和白点的xyY值来计算[11]。

其中xyY色彩需转换成XYZ色彩，具体的公式为[12]：

$$X = \frac{xY}{y}$$
$$Y = Y$$
$$Z = \frac{(1-x-y)Y}{y}$$

色彩适应

虽然转换成XYZ空间，但空间的白点却不相同，为此需要在不同白点的XYZ空间间转换，将色彩从一种光源变换成另一种光源下的操作叫色彩适应[18]。由于色彩是各波长反射函数到三刺激值的非单映射，这意味着在一种光源下相同的颜色可能在另一种光源下会有差异。因此正确的进行色彩适应是不可能的。但通常会用线性变换来近似色彩适应。

这个转换过程首先需要把白点的XYZ空间变换成视锥反应空间（CRD, Cone Response Domain），(ρ, γ, β)，有[13]：

$$\begin{bmatrix} \rho \\ \gamma \\ \beta \end{bmatrix} = M_{XYZ2CRD} \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix}$$

其中$M_{XYZ2CRD}$可选择单位矩阵，Bradford矩阵和Von Kries矩阵。默认选择Bradford矩阵。

色彩适应最终可以表示为：

$$\begin{bmatrix} X_d \\ Y_d \\ Z_d \end{bmatrix} = M_{XYZ2XYZ} \begin{bmatrix} X_s \\ Y_s \\ Z_s \end{bmatrix}$$

其中s下标为源XYZ空间，而d下标为目标XYZ空间，而：

$$M_{XYZ2XYZ} = M_{XYZ2CRD}^{-1} \begin{bmatrix} \rho_d/\rho_s & 0 & 0 \\ 0 & \gamma_d/\gamma_s & 0 \\ 0 & 0 & \beta_d/\beta_s \end{bmatrix} M_{XYZ2CRD}$$

其中$(\rho_s, \gamma_s, \beta_s)$和$(\rho_d, \gamma_d, \beta_d)$分别是源XYZ空间白点和目标XYZ空间白点的CRD值。

线性RGB转换成XYZ

因此线性RGB空间转换成某白点的XYZ空间，最终的表达式是：

$$\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = M_{XYZ2XYZ'} M_{RGBL2XYZ} \begin{bmatrix} R_l \\ G_l \\ B_l \end{bmatrix}$$

在程序实现的时候，通常会一次计算多种色彩，此时会将通道作为最后一个维度，因此，如果是二维矩阵，则可以表达成：

$$\begin{bmatrix} X'_1 & Y'_1 & Z'_1 \\ X'_2 & Y'_2 & Z'_2 \\ \dots & \dots & \dots \end{bmatrix} = \begin{bmatrix} R_{l1} & G_{l1} & B_{l1} \\ R_{l2} & G_{l2} & B_{l2} \\ \dots & \dots & \dots \end{bmatrix} \times (M_{XYZ2XYZ'} M_{RGBL2XYZ})^T$$

上式中，乘法符号被显示的写出来了。此时，该乘法与矩阵乘法相同，在numpy中可以使用numpy.dot, numpy.matmul或者@符号来实现。

如果是多阶张量（比如一张图片可以看出w×h×3的三阶张量），此时可示意表达成：

$$T_{XYZ'} = T_{RGBL} \times (M_{XYZ2XYZ} M_{RGBL2XYZ})^T$$

此时的乘法是扩展的矩阵乘法，依然可以使用numpy.dot, numpy.matmul或者@符号来实现，此时numpy.dot, numpy.matmul的结果是一致的。

XYZ转换成线性RGB

XYZ空间转换成线性RGB空间，仅需要乘以转换矩阵的逆即可。

线性RGB空间转换成其他线性RGB空间

RGB空间转换成XYZ空间后，然后再由XYZ空间转换成其他RGB空间。如果是线性RGB空间转换成另一个线性RGB空间，则两者仅相差一个矩阵，即：

$$T'_{RGBL} = T_{RGBL} \times (M_{RGBL2RGBL'})^T$$

其中：

$$M_{RGBL2RGBL'} = (M_{XYZ'2RGBL'} M_{XYZ2XYZ'} M_{RGBL2XYZ})^T$$

线性RGB空间与CIE lab之间的变换

此外，通过转换成XYZ空间可以转换为CIE lab空间，类似的，可以实现CIE lab空间转换成某XYZ空间。XYZ与CIE lab空间的转换关系可见[14-15]。

参考文献

1. https://en.wikipedia.org/wiki/RGB_color_space
2. https://en.wikipedia.org/wiki/SRGB
3. https://en.wikipedia.org/wiki/Adobe_RGB_color_space

4. https://en.wikipedia.org/wiki/Wide-gamut_RGB_color_space
5. https://en.wikipedia.org/wiki/ProPhoto_RGB_color_space
6. https://en.wikipedia.org/wiki/DCI-P3
7. https://en.wikipedia.org/wiki/Rec._709
8. https://en.wikipedia.org/wiki/Rec._2020
9. http://www.brucelindbloom.com/WorkingSpaceInfo.html
10. https://en.wikipedia.org/wiki/Gamma_correction
11. http://www.brucelindbloom.com/Eqn_RGB_XYZ_Matrix.html
12. http://www.brucelindbloom.com/Eqn_xyY_to_XYZ.html
13. http://www.brucelindbloom.com/Eqn_ChromAdapt.html
14. http://www.brucelindbloom.com/Eqn_XYZ_to_Lab.html
15. http://www.brucelindbloom.com/Eqn_Lab_to_XYZ.html