

更多嵌入式 Linux 学习资料, 请关注: 一口 Linux 回复关键字:1024



一. 概述

本文章会详细的介绍 RGB 颜色空间与 RGB 三色中色调、饱和度、亮度之间的关系, 最后会介绍 HSV 颜色空间!

一. RGB 颜色空间

1. 起源

RGB 三原色起源于上世纪初 1809 年 Thomas Young 提出视觉的三原色学说, 随后 Helmholtz 在 1824 年也提出了三原色学说: 即: 视网膜存在三种视锥细胞, 分别含有对红、绿、蓝三种光线敏感的视色素, 当一定波长的光线作用于视网膜时, 以一定的比例使三种视锥细胞分别产生不同程度的兴奋, 这样的信息传至大脑中枢, 就产生某一种颜色的感觉。

在显示器发明之后, 从黑白显示器发展到彩色显示器, 人们开始使用发出不同颜色的光的荧光粉 (CRT, 等离子体显示器), 或者不同颜色的滤色片 (LCD), 或者不同颜色的半导体

发光器件 (OLED 和 LED 大型全彩显示牌) 来形成色彩, 无一例外的选择了 Red, Green, Blue 这 3 种颜色的发光体作为基本的发光单元。通过控制他们发光强度, 组合出了人眼睛能够感受到的大多数的自然色彩。

计算机显示彩色图像的时候也不例外, 最终显示的时候, 要控制一个像素中 Red, Green, Blue 的值, 来确定这个像素的颜色。计算机中无法模拟连续的存储从最暗到最亮的量值, 而只能以数字的方式表示。于是, 结合人眼睛的敏感程度, 使用 3 个字节 (3*8 位) 来分别表示一个像素里面的 Red, Green 和 Blue 的发光强度数值, 这就是常见的 RGB 格式。我们可以打开画图板, 在自定义颜色工具框中, 输入 r,g,b 值, 得到不同的颜色。

2. 概述

RGB 颜色空间以 R(Red:红)、G(Green:绿)、B(Blue:蓝)三种基本色为基础, 进行不同程度的叠加, 产生丰富而广泛的颜色, 所以俗称三基色模式。

RGB 空间是生活中最常用的一个颜色显示模型, 电视机、电脑的 CRT 显示器等大部分都是采用这种模型。自然界中的任何一种颜色都可以由红、绿、蓝三种色光混合而成, 现实生活中人们见到的颜色大多是混合而成的色彩。

肉眼可以识别世界上的所有颜色, 而 RGB 几乎可以组合成世界上所有的颜色!

二. RGB 颜色范围

RGB 的每个元素在计算机内存中占用 1 个字节, 1 个字节等于 8 个 bit 位, 所以 RGB 每个元素的取值范围为: 0~256(2^8 次方) (从 2 开始算次方我个人觉得是因为 bit 位, 因为一个 bit 位可以表示两种状态)

那么三色组合起来 $256 \times 256 \times 256 = 16777216$ (一千六百七十七万七千二百一十六) 种颜色, 这么多的颜色几乎可以覆盖世界上所有的颜色! 所以平常电视机、电脑屏幕、任何显示仪器都会通过这种 RGB 颜色空间来显示!

1. 组合方法

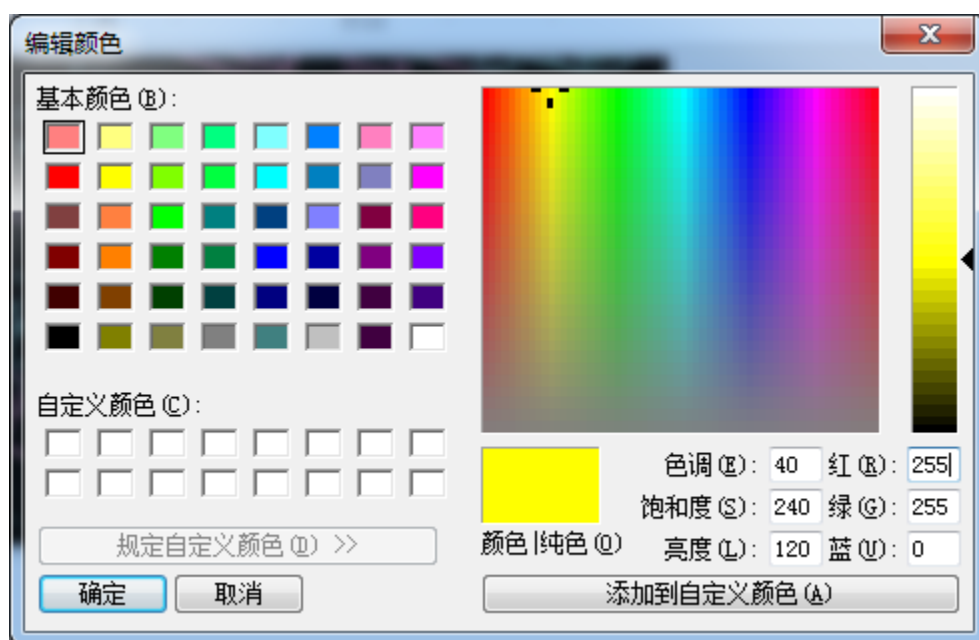
组合方法是通过互补光的形式来组合成任意颜色的

例如:

红色 + 绿色 = 黄色

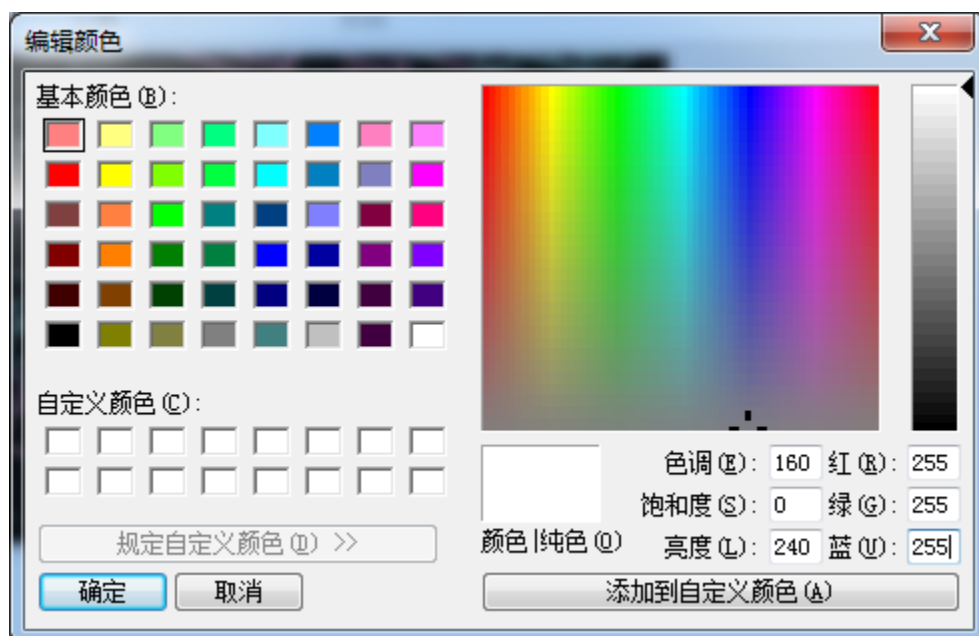
如图:



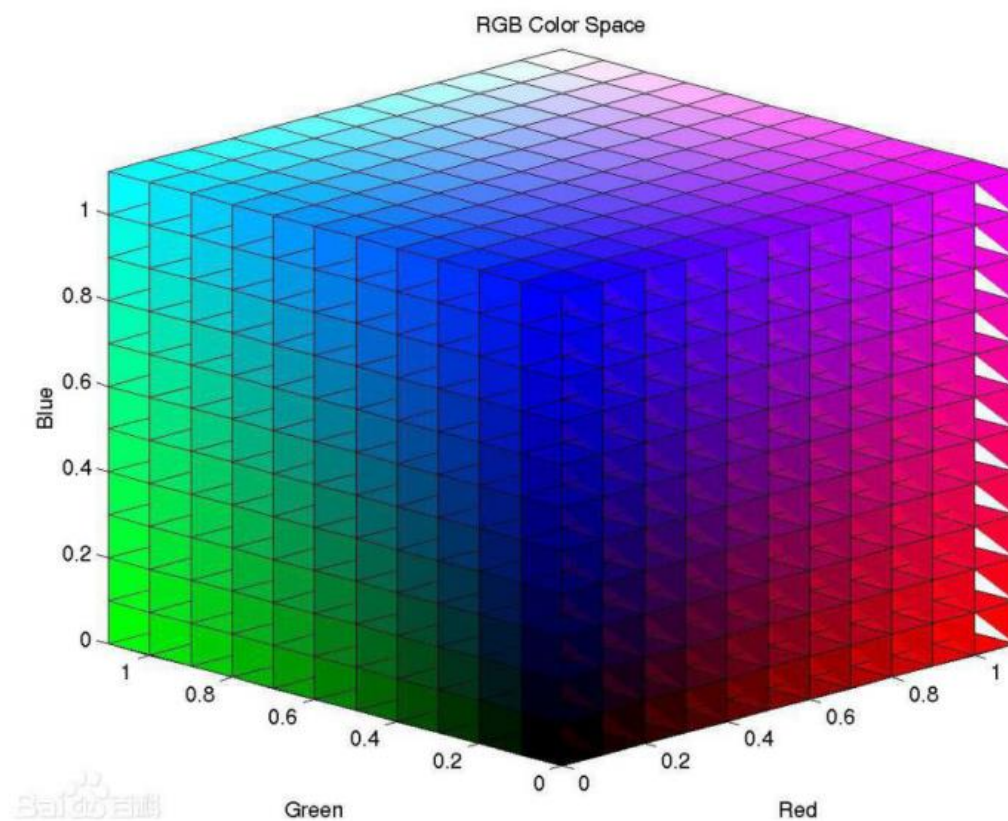


红色 + 绿色 + 蓝色 = 白色

如图:



你也可以调整 RGB 某一元素的色值来达到不同的颜色组合!

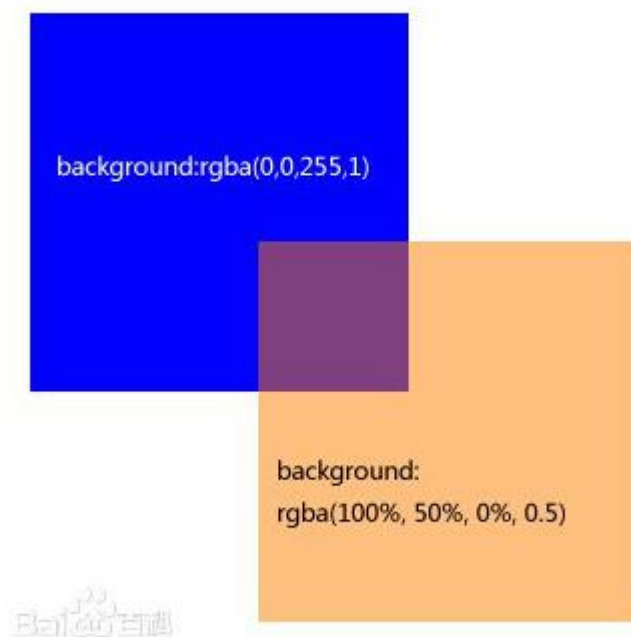


2. RGBA

RGBA (Alpha) , A 用于描述三原色的透明度!

例如:





3. RGB 文件存储类型

RGB 色彩空间根据每个分量在计算机中占用的存储字节数分为如下几种类型:

(1) RGB555

RGB555 是一种 16 位的 RGB 格式, 各分量都用 5 位表示, 剩下的一位不用。

高字节 -> 低字节(根据内存大小端)

XRRRRRGGGGGBBBBB(X 代表不用)

(2) RGB565

RGB565 也是一种 16 位的 RGB 格式, 但是 R 占用 5 位, G 占用 6 位, B 占用 5 位。

(3) RGB24

RGB24 是一种 24 位的 RGB 格式, 各分量占用 8 位, 取值范围为 0-255。

(4) RGB32

RGB32 是一种 32 位的 RGB 格式, 各分量占用 8 位, 剩下的 8 位作 Alpha 通道或者不用。

(5) RGB222

RGB222 是一种 6 位的 RGB 格式, 各分量占用 2bit 位, 剩下的 2bit 位不用!

注意这种格式一般用于灰度图, 现在的 CRT(阴极射线管)都是彩色 RGB 模型的 CRT, 所以在显示灰度图时都是放射 RGB 颜色模板(荧光粉)来做到显示颜色的, 也就是说每个像素点上的荧光粉有三个 RGB, 所以显示灰度图时也就是单通道的图片时, 是通过混合 RGB 来达到灰度色的!

还有 RGB232, RGB332, RGB233, ...

早期的 CRT 只能显示黑白照是因为工业水平有限, 注意也不一定是黑白色, 也有可能是单色, 如红色, 暗红, 紫色, 暗紫, 只能显示一种颜色!

4. 单通道与多通道

单通道:

俗称灰度图, 每个像素点只能有一个值表示颜色, 它的像素值在 0 到 255 之间, 0 是黑色, 255 是白色, 中间值是一些不同等级的灰色, 可以说灰度是黑与白之间的过渡色!

注意这个值不是 RGB 里的任何一个元素, 显示设备是直接通过 CRT(彩色阴极射线显像管)将单通道里的像素值显示黑白色图像, 值越高黑色图越亮, 一般灰度值大小不会超过 125!

多通道:

多通道也就是 RGB 三原色, 每个像素点有三个字节来表示 (RGB), 分别最大取值范围是 0-255, 可以组合成千万种颜色。

图像处理优势对比:

单通道往往应用于图像处理, 因为单通道只有一个像素点(一个字节), 所以相比多通道三个字节, 处理速度上要尤为的快!

而且单通道能将图像以灰度形式显示出来, 不会影响图像识别, 和特征提取!

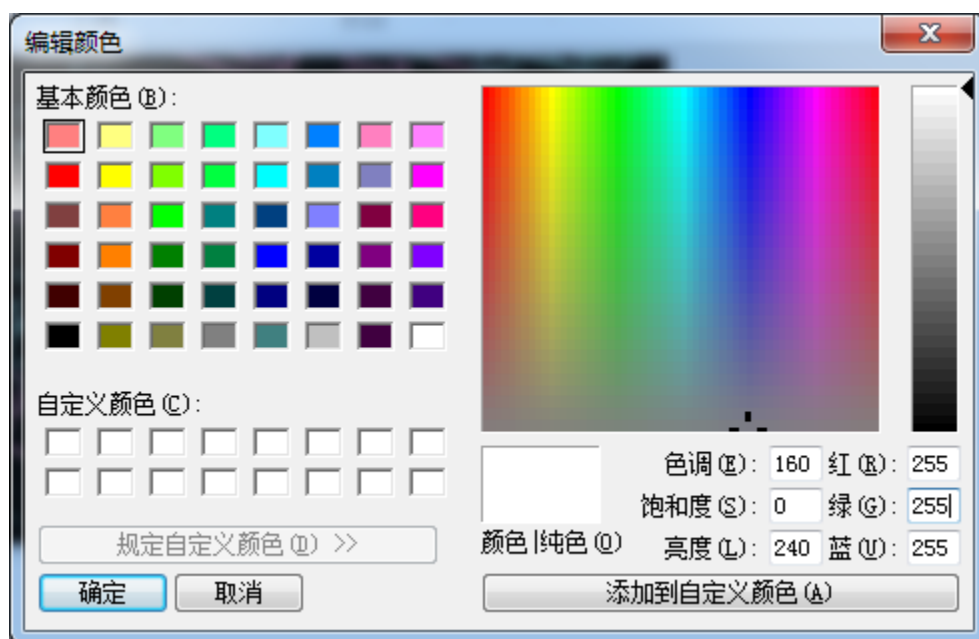
多通道以原图的形式将图像展示出来, 所以可以提取特征很多, 识别率高!

例如:



多通道也可以组合成灰度图, 上面说过, 红+绿+蓝=白色, 注意这里补充一下, 只有在三色相等时才会组合成白色!

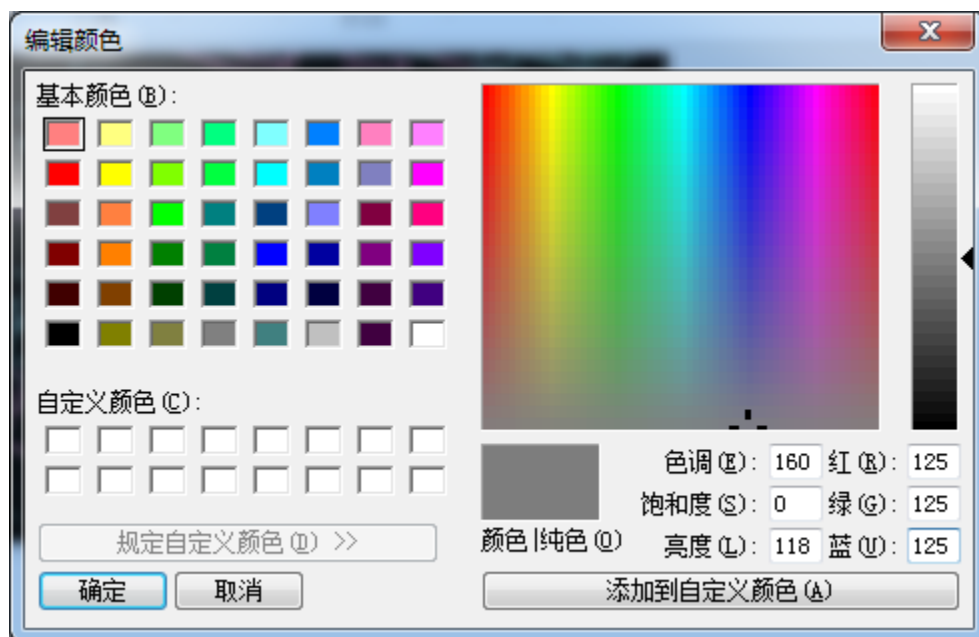
例如:



想要达到灰度只要将 RGB 三色调低一点就可以达到!

例如:





但是当某一方的值不相同就会产生其他颜色。

所以三通道想要组合成黑白色(灰度)必须三原色值相同。

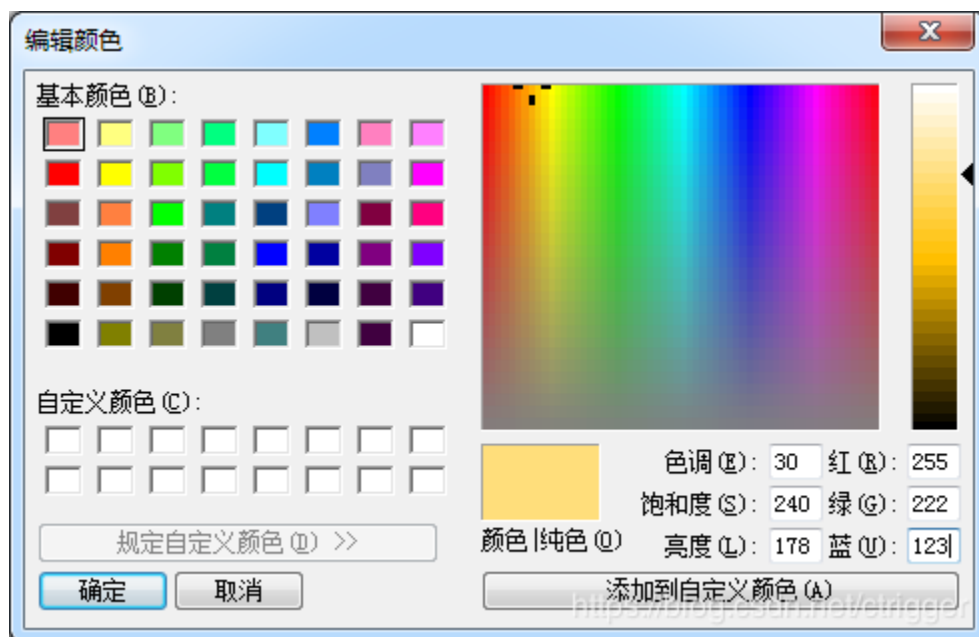
也就是说灰度图不一定是单通道, 但是单通道一定是灰度图!

三. 色调、饱和度、亮度

1. 色调(色相):

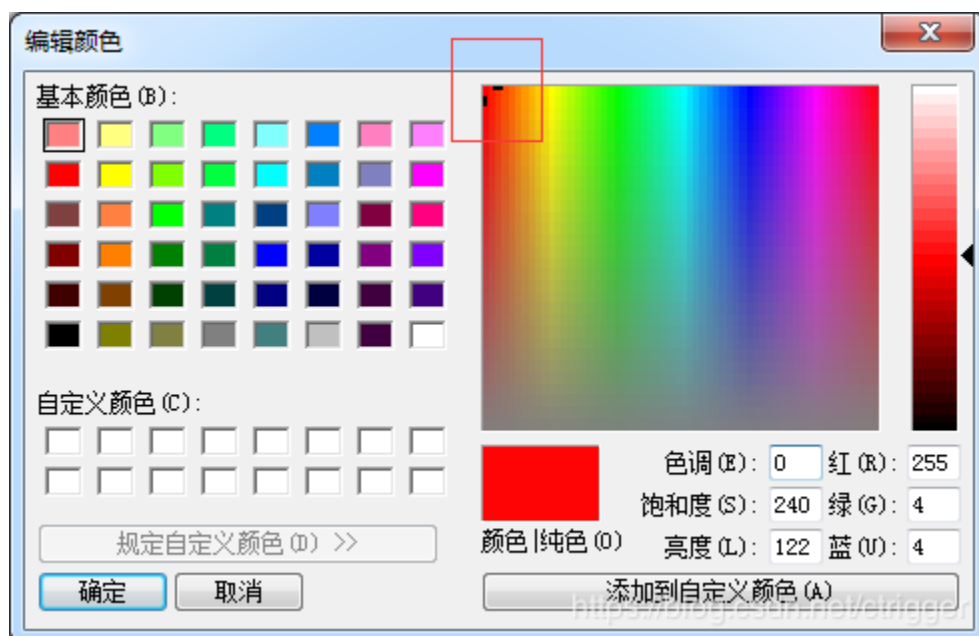
彩色图片中, 色调决定彩色图片更加偏于哪一方!

列如:

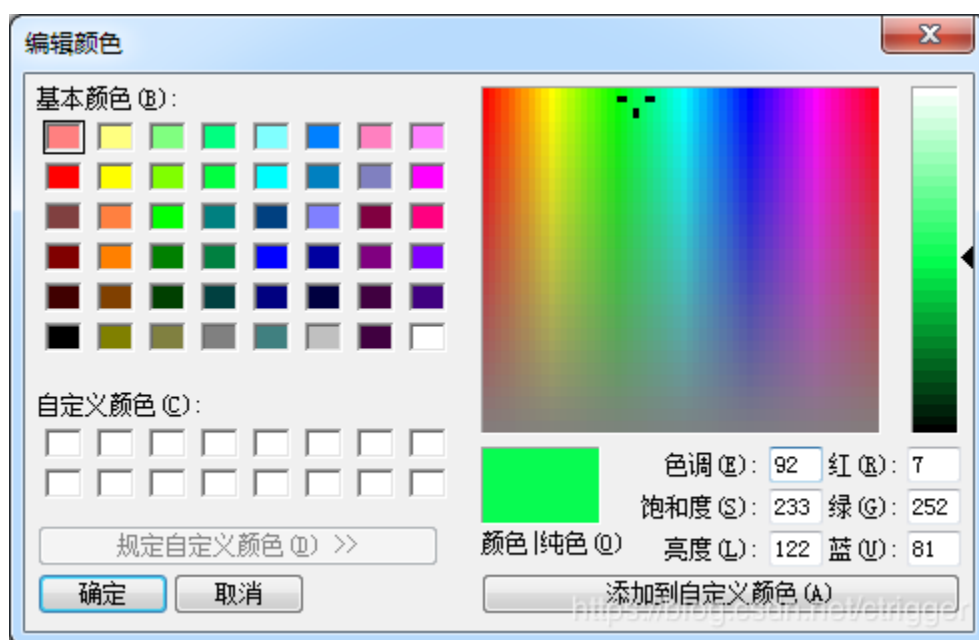


我们修改一下色调

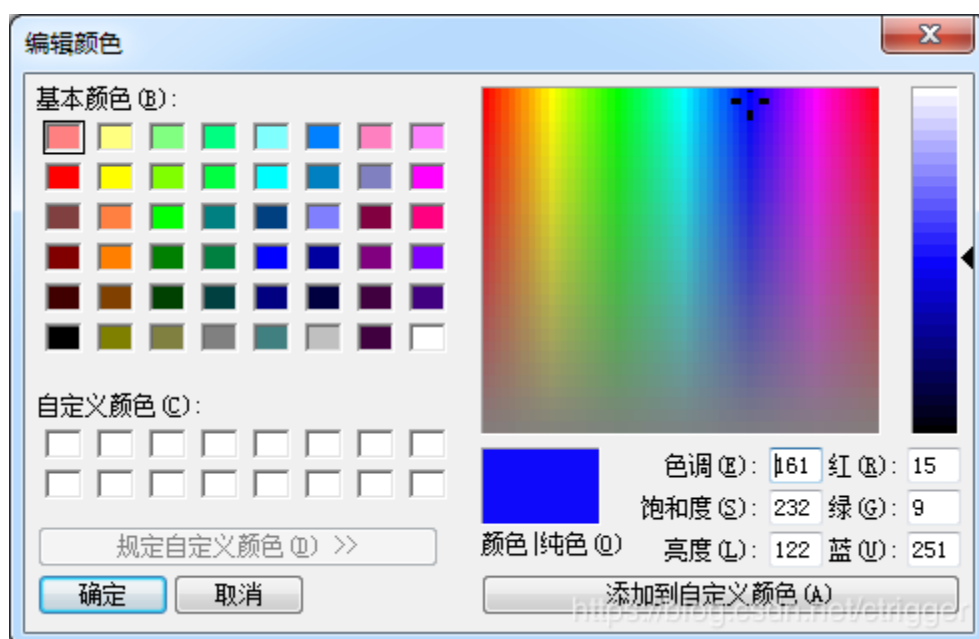
当把色调调低时, 颜色更加偏向于红色



当我们把色调调高一点时, 颜色更加偏向于绿色



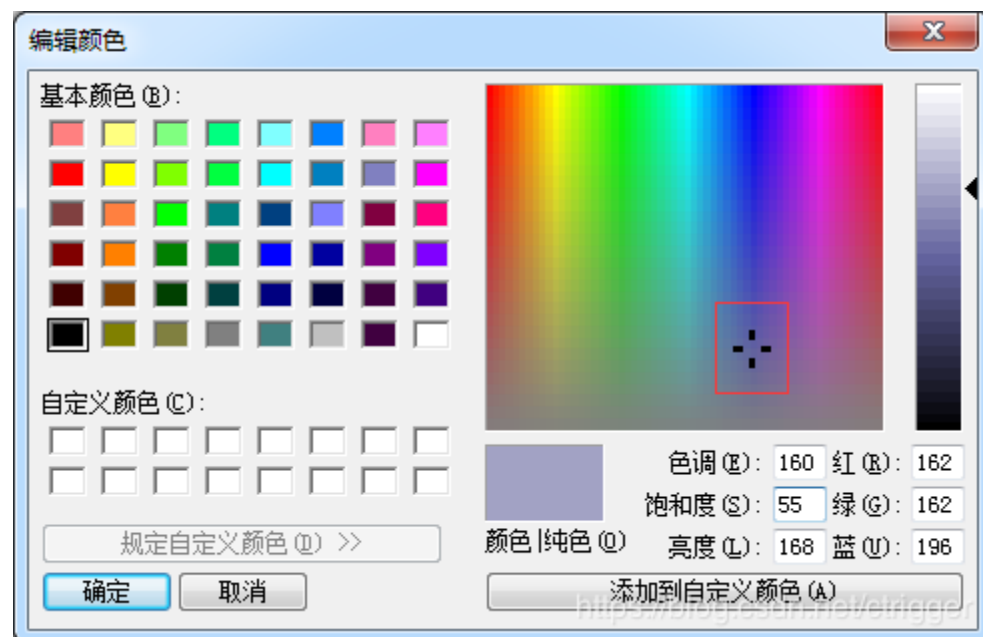
当颜色在调高一点时, 颜色更加偏向于蓝色



2. 饱和度

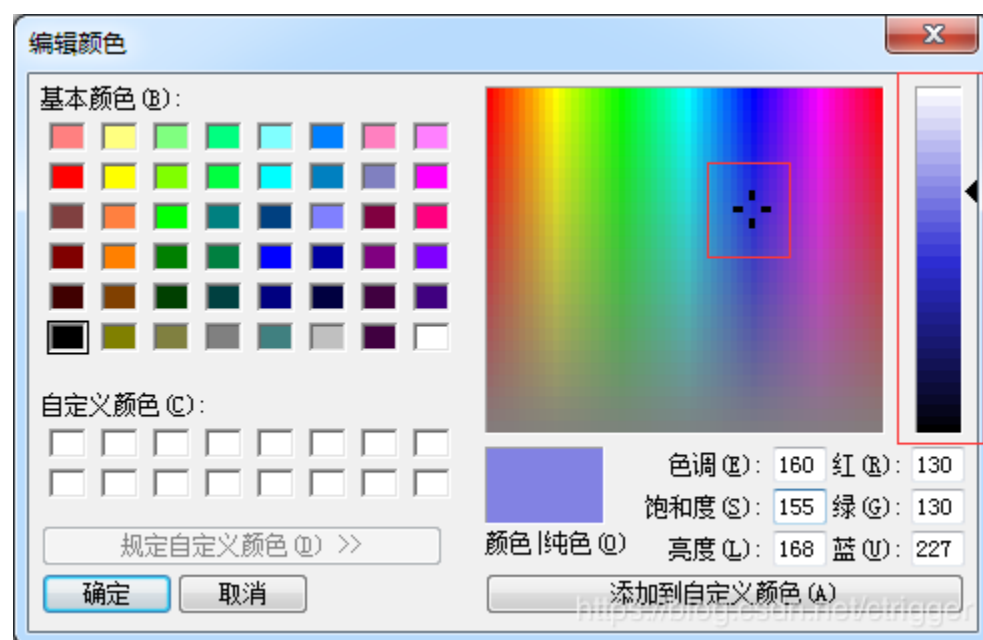
饱和度决定了颜色空间中颜色分量, 饱和度越高, 说明颜色越深, 饱和度越低, 说明颜色越浅!

如图:



当饱和度为 55 时, 可以发现该颜色空间能显示的颜色分量非常低

当我把饱和度调高一点时, 可以发现颜色分量显示的明显要深!



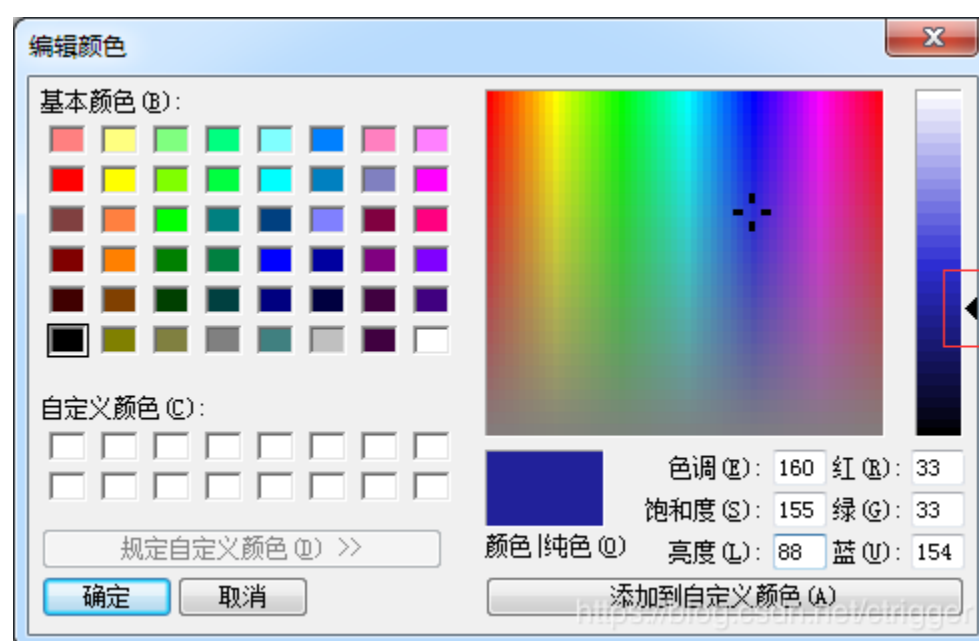
可以与上图形成鲜明的对比。

所以饱和度在颜色空间中是起到一个控制 RGB 组合色的颜色深度的作用。

3. 亮度

亮度决定颜色空间中颜色的明暗程度!

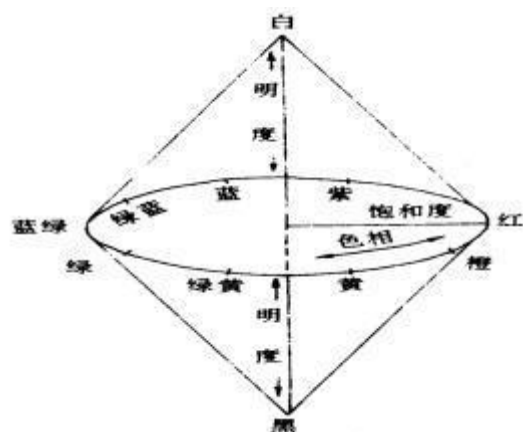
如图, 亮度设置比较高的时候会发现颜色显示的较为鲜艳



会发现颜色会变得非常暗!

所以亮度在颜色空间中起到一个控制 RGB 组合色的明暗程度的作用。

下面有个答案是我在知乎上看到的觉得非常适合描述色调(色相)、饱和度、亮度与颜色空间的关系:



1. 明度为 0 的时候, 只有一个点, 只能是黑色。没有光, 啥都看不见。

后面我们要让明度是某个不为零的值, 才好谈下去。基于这个条件,

2. 纯度为 0 的时候, 只有一条线, 只能是黑白的。没有对比度, 就没有彩色。

3. 纯度也不为 0 了, 才可能出现彩色, 至于到底是哪一种颜色, 就要看色度了。

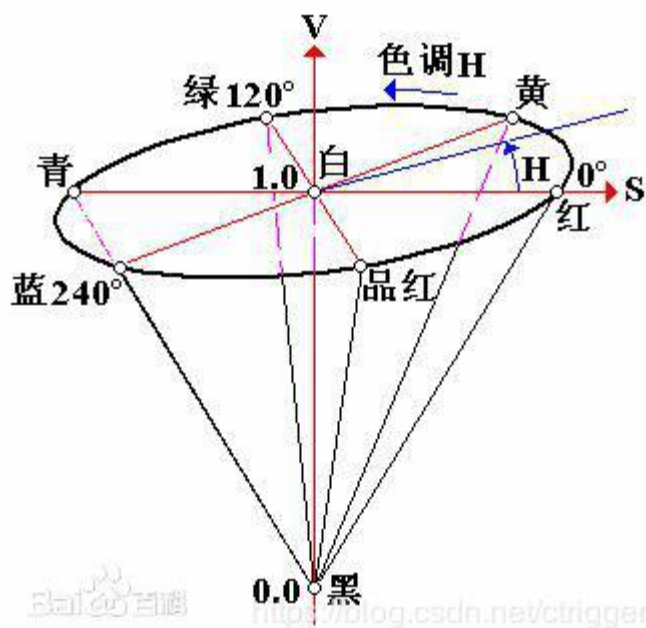
不知道大家有没有发现, 无论你怎么修改色调, 饱和度, 亮度, RGB 三色值会跟随而变化, 其实色调, 饱和度, 亮度都是通过特定的算法经过计算修改 RGB 三色而达到的控制颜色效果!

本栏属于图像理论知识不提供任何实践代码!

四. HSV 颜色空间 (部分地方是参考他人博客的文章, 然后根据自己的理解写出来的!)

1. 起源

HSV(Hue, Saturation, Value)是根据颜色的直观特性由 A. R. Smith 在 1978 年创建的一种颜色空间, 也称六角锥体模型(Hexcone Model)。

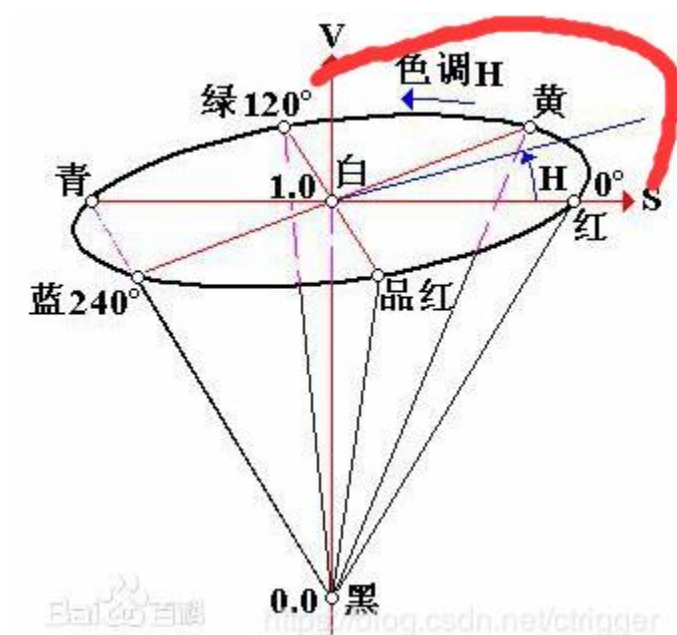


这个模型中颜色的参数分别是：色调 (H) ， 饱和度 (S) ， 明度 (V) 。

2. HSV 颜色模型介绍

2.1 色调 H (Hue)

用角度度量, 取值范围为 $0^\circ \sim 360^\circ$, 从红色开始按逆时针方向计算, 红色为 0° , 绿色为 120° , 蓝色为 240° 。它们的补色是：黄色为 60° , 青色为 180° , 品红为 300° , $0^\circ - 359^\circ$ 时颜色会依次变换当角度到达 360° 时也就是红色, 角度也就又回到 0° 了, 所以总共为 360° , 每变换 1° 时, 色相就会有轻微的变化! 如果是顺时针的话这个变换过程会从红色逐渐变换到绿色, 在由绿色逐渐变换到蓝色, 在由蓝色逐渐变换到红色! 逆时针的话就是相反的!

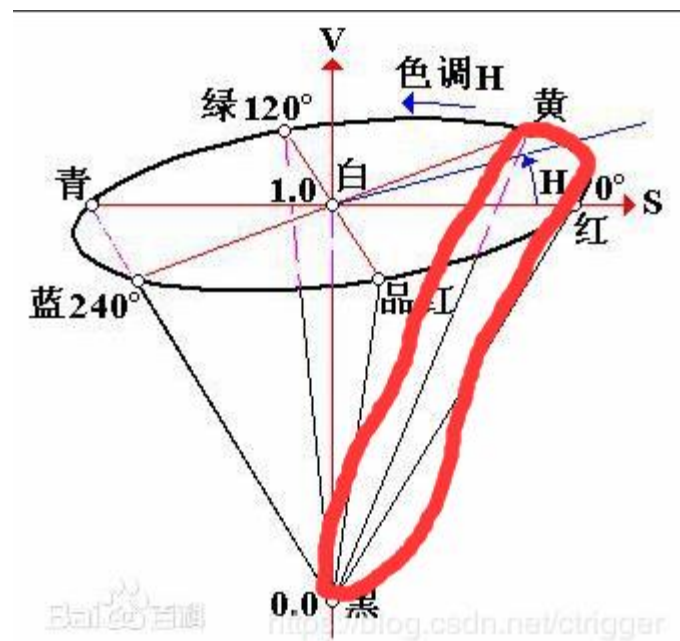


2.2 饱和度 S (Saturation)

饱和度 S 表示颜色接近光谱色的程度。一种颜色, 可以看成是某种光谱色与白色混合的结果。其中光谱色所占的比例愈大, 颜色接近光谱色的程度就愈高, 颜色的饱和度也就愈高。

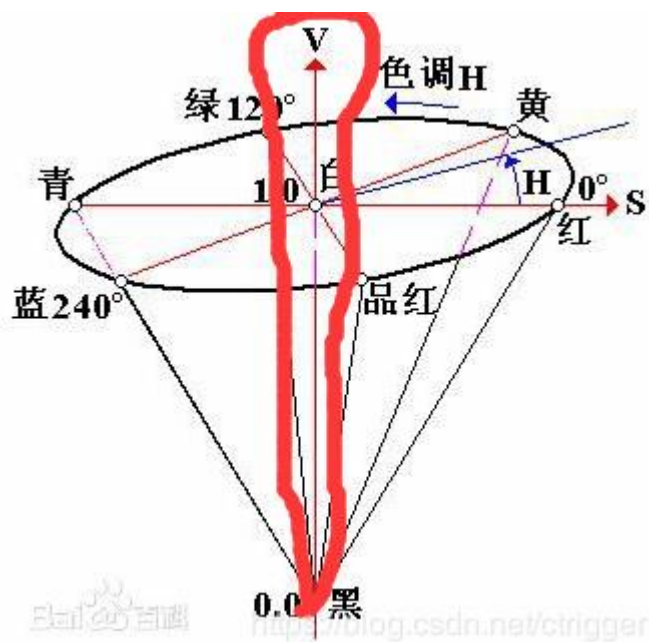
饱和度高, 颜色则深而艳。光谱色的白光成分为 0, 饱和度达到最高。通常取值范围为 0% ~

100%, 值越大, 颜色越饱和。

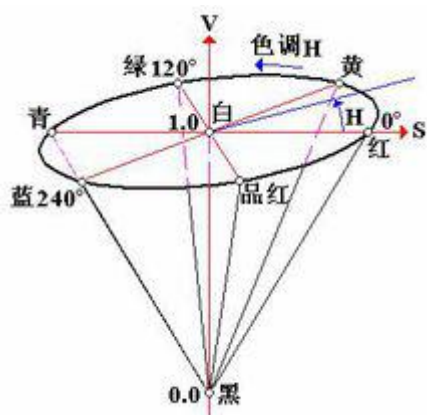


2.3 明度 V (Value)

明度表示颜色明亮的程度, 对于光源色, 明度值与发光体的光亮度有关; 通常取值范围为 0% (黑) 到 100% (白)。



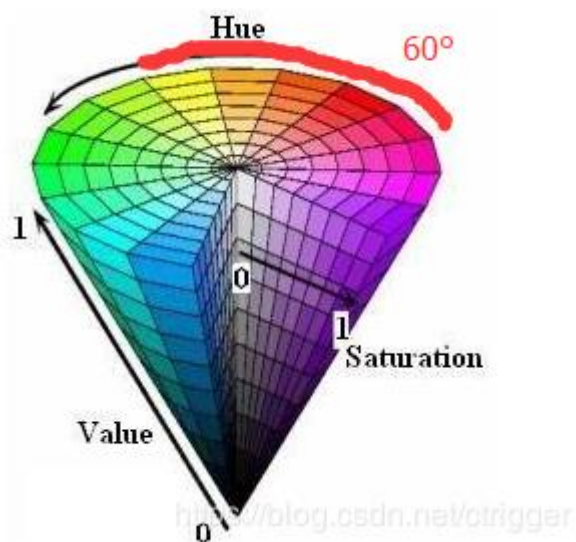
2.4 模型分析:



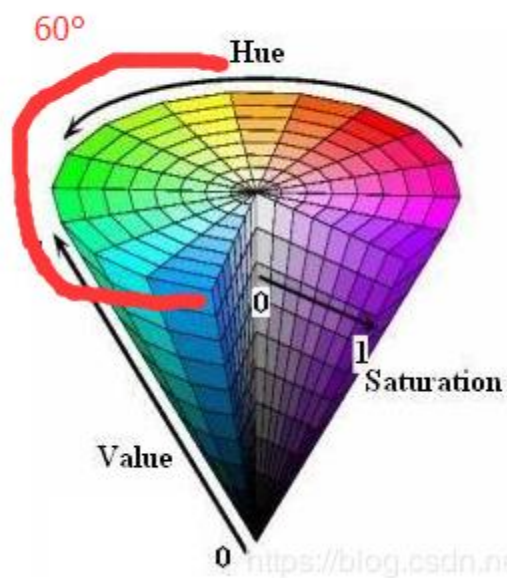
H 参数表示色彩信息, 即所处的光谱颜色的位置。该参数用一角度量来表示, 红、绿、蓝分别相隔 120 度。HSV 中每一种颜色的互补色分别相差 180 度。意思就是说: 两种颜色在互补时最大为 180°

例如:

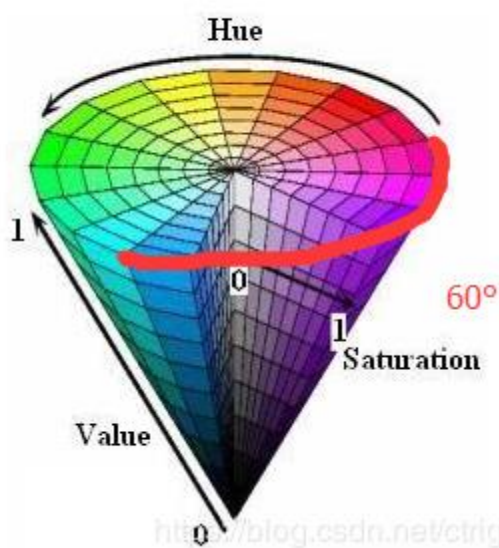
在 HSV 模型中红与绿的互补色为黄色, 其角度为 60°



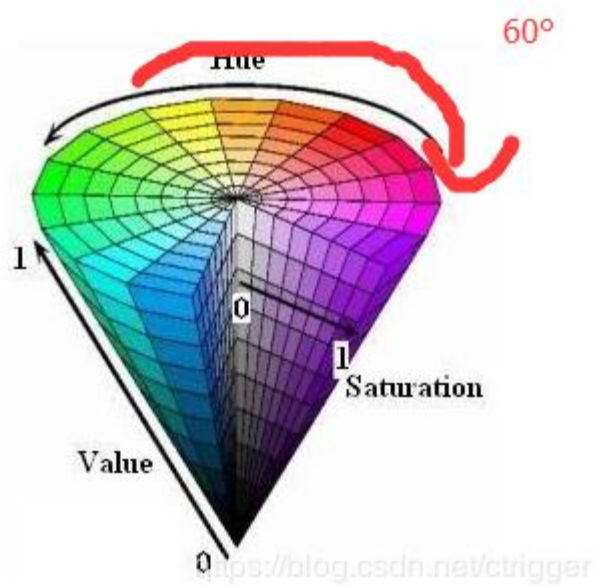
绿色与蓝色的互补光为青色其角度也为 60°



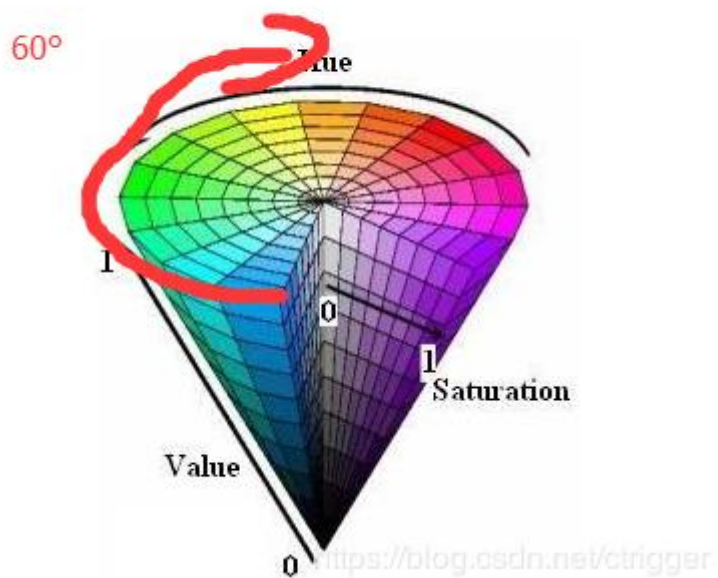
蓝色与红色的互补光为品红色其角度也为 60°



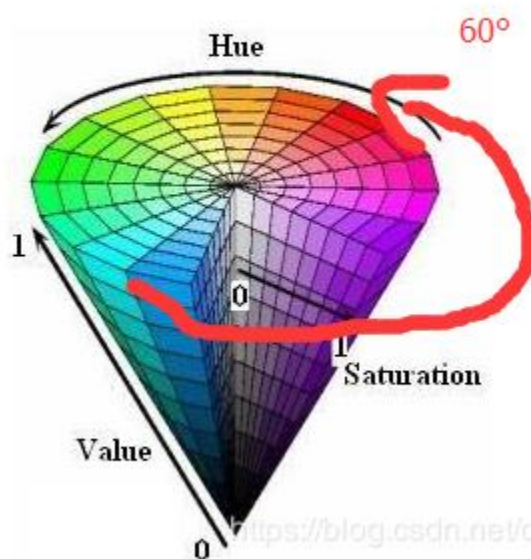
那么按逆反的方向来算, 绿色到红色的互补光为 60°



蓝色到绿色的互补光也为 60°



红色到蓝色的互补光也为 60°



所以通过以上知识可以知道, 红色到绿色之间的互补光为 60° , 而绿色到红色之间的互补光也为 60° 所以每一种颜色的色差是: $60^\circ + 60^\circ = 180^\circ$

互补光的色差在 HSV 颜色模型中是这样来算的!

纯度 S 为一比例值, 范围从 0 到 1, 它表示成所选颜色的纯度和该颜色最大的纯度之间的比率。S=0 时, 只有灰度。

V 表示色彩的明亮程度, 范围从 0 到 1。有一点要注意: 它和光强度之间并没有直接的联系。

2.5 HSV 颜色分析

HSV 对用户来说是一种直观的颜色模型。我们可以从一种纯色彩开始, 即指定色彩角 H, 并让 $V=S=1$, 然后我们可以通过向其中加入黑色和白色来得到我们需要的颜色。增加黑色

可以减小 V 而 S 不变, 同样增加白色可以减小 S 而 V 不变。例如, 要得到深蓝色, $V=0.4$
 $S=1$ $H=240$ 度。要得到浅蓝色, $V=1$ $S=0.4$ $H=240$ 度。

如图:

Color	R	G	B	H	H ₂	C	C ₂	V	L	I	Y ₆₀₁	S _{HSV}	S _{HSL}	S _{HSI}
	1.000	1.000	1.000	n/a	n/a	0.000	0.000	1.000	1.000	1.000	1.000	0.000	0.000	0.000
	0.500	0.500	0.500	n/a	n/a	0.000	0.000	0.500	0.500	0.500	0.500	0.000	0.000	0.000
	0.000	0.000	0.000	n/a	n/a	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	1.000	0.000	0.000	0.0°	0.0°	1.000	1.000	1.000	0.500	0.333	0.299	1.000	1.000	1.000
	0.750	0.750	0.000	60.0°	60.0°	0.750	0.750	0.750	0.375	0.500	0.664	1.000	1.000	1.000
	0.000	0.500	0.000	120.0°	120.0°	0.500	0.500	0.500	0.250	0.167	0.293	1.000	1.000	1.000
	0.500	1.000	1.000	180.0°	180.0°	0.500	0.500	1.000	0.750	0.833	0.850	0.500	1.000	0.400
	0.500	0.500	1.000	240.0°	240.0°	0.500	0.500	1.000	0.750	0.667	0.557	0.500	1.000	0.250
	0.750	0.250	0.750	300.0°	300.0°	0.500	0.500	0.750	0.500	0.583	0.457	0.667	0.500	0.571
	0.628	0.643	0.142	61.8°	61.5°	0.501	0.494	0.643	0.393	0.471	0.581	0.779	0.638	0.699
	0.255	0.104	0.918	251.1°	250.0°	0.814	0.750	0.918	0.511	0.426	0.242	0.887	0.832	0.756
	0.116	0.675	0.255	134.9°	133.8°	0.559	0.504	0.675	0.396	0.349	0.460	0.828	0.707	0.667
	0.941	0.785	0.053	49.5°	50.5°	0.888	0.821	0.941	0.497	0.593	0.748	0.944	0.893	0.911
	0.704	0.187	0.897	283.7°	284.8°	0.710	0.636	0.897	0.542	0.596	0.423	0.792	0.775	0.686
	0.931	0.463	0.316	14.3°	13.2°	0.615	0.556	0.931	0.624	0.570	0.586	0.661	0.817	0.446
	0.998	0.974	0.532	56.9°	57.4°	0.466	0.454	0.998	0.765	0.835	0.931	0.467	0.991	0.363
	0.099	0.795	0.591	162.4°	163.4°	0.696	0.620	0.795	0.447	0.495	0.564	0.875	0.779	0.800
	0.211	0.149	0.597	248.3°	247.3°	0.448	0.420	0.597	0.373	0.319	0.219	0.750	0.601	0.533
	0.495	0.493	0.721	240.5°	240.4°	0.228	0.227	0.721	0.607	0.570	0.520	0.316	0.290	0.135

一般说来, 人眼最大能区分 128 种不同的色彩, 130 种色饱和度, 23 种明暗度。如果我们
 用 16Bit 表示 HSV 的话, 可以用 7 位存放 H, 4 位存放 S, 5 位存放 V, 即 745 或者 655
 就可以满足我们的需要了。

由于 HSV 是一种比较直观的颜色模型, 所以在许多图像编辑工具中应用比较广泛, 如
 Photoshop (在 Photoshop 中叫 HSB) 等等, 但这也决定了它不适合使用在光照模型中,
 许多光线混合运算、光强运算等都无法直接使用 HSV 来实现!

RGB 颜色空间更加面向于工业, 而 HSV 更加面向于用户, 大多数做图像识别这一块都会运用 HSV 颜色空间, 因为 HSV 颜色空间表达起来更加直观!

公众号: 一口 Linux