

色调、饱和度和亮度调整的调研记录

对海康相机的 HSL 三个维度的调整方式进行了研究（买了一台线扫相机，能有公司支持真是太好了），结合之前对 Photopea 这款图片处理软件的调整方式分析，结果能发现海康相机为了减小运算量，它做了一些不同的操作，有可借鉴之处，也有粗糙之处。

文档的逻辑是这样的：因为海康的操作和一些常规做法是有些区别的，有些地方可以借鉴但又不能完全照搬，所以把海康和常规都写一下，作为对比，也算是对最近看的方法的一个总结，毕竟花了相当长时间和思考。每小节先给出海康的调整方式，然后给出其他调研到的调整方式，最后是我的个人看法。所以会有点长，如果直接看海康的做法，那么就直接看每一小节的最开始部分。

一、色调

1.1 海康的调整

海康可以调整为 0-255，先归一化到 -1~1 之间，其中 0-127 是负数、129-255 是正数。

色调调整用公式表示，相当于它是做了如下的矩阵操作（这样写就可以用到 CCM 那个流程的矩阵相乘的代码）：

$$\begin{bmatrix} k1 & k2 & k3 \\ k3 & k1 & k2 \\ k2 & k3 & k1 \end{bmatrix} \times \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

其中 k1-k3 的改变方式如下（j 是归一化后的系数）：

$$k1 = \begin{cases} 1 + j, & j < 0 \\ 1 - j, & j \geq 0 \end{cases} \quad k2 = \begin{cases} -j, & j < 0 \\ 0, & j \geq 0 \end{cases} \quad k3 = \begin{cases} 0, & j < 0 \\ j, & j \geq 0 \end{cases}$$

这个调整方式理解起来就是改变了不同颜色的权重：

1. 色调 j 为 -1 的时候，只有 k2 为 1，此时的红色变成了原来图片的绿色、绿变蓝、蓝变红，所以图片的颜色发生了很厉害的翻转。
2. 随着色调从负数逐渐增加到 0，k1 变大、k2 减小，而 k1 代表了原来颜色的权重，所以红色逐步恢复回原先的红色，其他颜色同理。
3. 色调正数的调整也是如此，只不过此时 k3 代替 k2 开始起了作用，如对于红色，它逐步受原来蓝色的值的影响，直到彻底变成原来的蓝色。

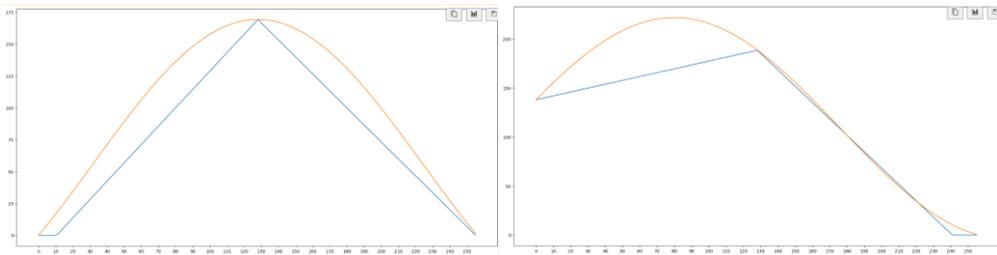
1.2 常见方式一：RGB 空间调整

第一种常见方式和海康类似(或者说海康是仿照了这种方式),一样是用了上面的那个矩阵相乘。

只不过 k1-k3 的调整有区别,调整方式如下:

$$\begin{cases} k1 = \cos(j\pi) + (1 - \cos(j\pi)) / 3 \\ k2 = -\sqrt{1/3} \sin(j\pi) + (1 - \cos(j\pi)) / 3 \\ k3 = +\sqrt{1/3} \sin(j\pi) + (1 - \cos(j\pi)) / 3 \end{cases}$$

其实就是改变的更加平滑了,如下图所示,只不过海康应该是为了减小运算,使用了自己的调整方式(蓝色是海康调整时某张图片红色随着色调变化的值,黄色是仿真使用这个方式调整的结果)。



参考: <https://stackoverflow.com/questions/8507885>

1.3 常见方式二：HSL/HSV 空间 (Photopea 处理方式)

第二种调整方式就是先把 RGB 转成 HSL 或者 HSV, 然后去调整色调 H(Hue), 之后将新的 HSL 或 HSV 转回到 RGB。其中色调调整方式为 $H = (H + \Delta) \% 360$ 。

这种方式的好处就是精确,因为色调本身的定义就是从 HSL 和 HSV 空间来的,这种调整最准确;但是坏处就是做了两次转换(RGB 转 HSL 再转 RGB),运算量会大,尤其这两个空间转到 RGB 的时候有一些条件判断,是并行计算不是很喜欢的方式。

1.4 常见方式三：YIQ 空间

第三种方式是利用了 YIQ 空间,这个空间和 YUV 空间类似, Y 表示亮度,而 I 和 Q 则代表了颜色,一个是红色分量,一个是蓝色分量。

思路时先转到 YIQ 空间,这个很好办,是一个线性操作;然后做一些调整就能改变色调和饱和度;最后再转回到 RGB,这个操作也方便(也是线性)。

其实也可以用 YUV、YcrBr 这些类似空间做操作,但是研究人员(其实就是某个博主)认为在 YIQ 空间调整色调、饱和度之类的稍微会更准确一些。

总之，最终如果利用 YIQ 空间进行色调变换再返回到 RGB 空间，它最终的公式就变为为了：

$$\begin{bmatrix} 0.299 + 0.701 * \cos(j\pi) + 0.168 * \sin(j\pi) & 0.587 - 0.587 * \cos(j\pi) + 0.330 * \sin(j\pi) & 0.114 - 0.114 * \cos(j\pi) - 0.497 * \sin(j\pi) \\ 0.299 - 0.299 * \cos(j\pi) - 0.328 * \sin(j\pi) & 0.587 + 0.413 * \cos(j\pi) + 0.035 * \sin(j\pi) & 0.114 - 0.114 * \cos(j\pi) + 0.292 * \sin(j\pi) \\ 0.299 - 0.300 * \cos(j\pi) + 1.250 * \sin(j\pi) & 0.587 - 0.588 * \cos(j\pi) - 1.050 * \sin(j\pi) & 0.114 + 0.886 * \cos(j\pi) - 0.203 * \sin(j\pi) \end{bmatrix} \times \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

看起来很复杂，实际都是线性操作，很适合并行计算。这个方法的优点就是快，缺点是这种调整方式有点非主流，首发来自一篇博客，后面也有人讨论说有的时候不符合预期。

参考

1. <https://beesbuzz.biz/code/16-hsv-color-transforms>
2. <https://stackoverflow.com/a/8509802/7780906>
3. <https://zhuanlan.zhihu.com/p/98832366>

1.5 个人想法

海康的色调调整是有借鉴之处的，没有使用先前文档中 RGB 转 HSL 再转 RGB 的操作，给了我更深入调研的契机。但是它的调整系数我认为太简单粗糙了，如果调整方式一中能用 MCU 先计算好 k1-k3 值（涉及到三角函数），或许调整方式一更为合理一些。

即进行如下调整：

$$\begin{bmatrix} k1 & k2 & k3 \\ k3 & k1 & k2 \\ k2 & k3 & k1 \end{bmatrix} \times \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

其中系数调整如下（j 表示用户输入的系数）：

$$\begin{cases} k1 = \cos(j\pi) + (1 - \cos(j\pi)) / 3 \\ k2 = -\sqrt{1/3} \sin(j\pi) + (1 - \cos(j\pi)) / 3 \\ k3 = +\sqrt{1/3} \sin(j\pi) + (1 - \cos(j\pi)) / 3 \end{cases}$$

二、饱和度

2.1 海康的调整

和色调一样，海康可以调整饱和度为 0-255，还是先归一化到 -1~1 之间，其中 0-127 是负数、129-255 是正数。

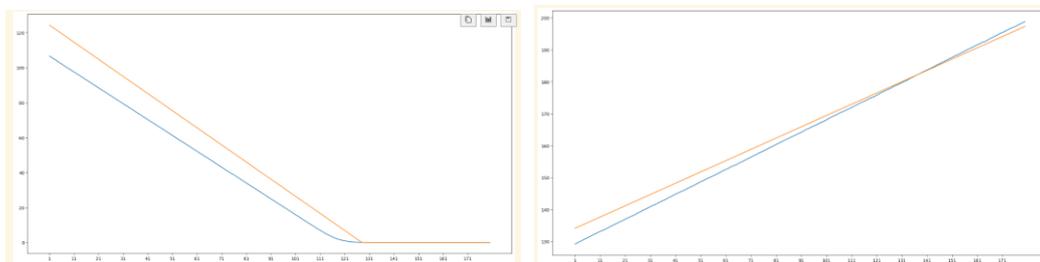
海康的饱和度调整直接在 RGB 空间上做的，其调整方式如下（以红色为例）：

$$newR = srcR + (srcR - Y) \times 0.744 \times j$$

$$\text{其中：} Y = 0.299 \times R + 0.587 \times G + 0.114 \times B$$

公式中的 j 就是归一化后的系数， Y 没有什么特殊含义，就是 YUV 空间中的 Y ，表示亮度。

这里的 0.744 很奇怪，但所得的结论确实如此，如下图所示，横坐标是不同饱和度，纵坐标时蓝色线是海康相机采的图，黄色线是仿真的结果（以海康相机没开饱和度调整时采集的图为基础，使用上面公式模拟不同饱和度时的情况）。两条线非常接近。



猜测可能是 0.744 是一个 bug，因为 $0.744 = 128/172$ ，可能原来是打算设置为 $128/127$ ？

2.2 常见方式一：RGB 空间调整

方式一和海康的调整类似，其方式如下：

$$\begin{aligned} newR &= srcR + (avg - srcR) \times j \\ avg &= (srcR + srcG + srcB) \div 3 \end{aligned}$$

区别就是用平均值代替 Y ，以及没有了 0.744 这个奇怪的数字。也有方式用 RGB 三个值的中位数来代替平均值。我个人觉得海康中使用 Y 来代替平均值是一个很不错的想法，因为饱和度调整为 1 的时候，彩色图变为灰色图，而此时灰色图的值为 Y 值比 RGB 平均值更好一些。

2.3 常见方式二：HSL 空间调整 (Photopea 处理方式)

Photoshop 使用的是 HSL 空间，调整方式如下。

$$newS = \begin{cases} \frac{1}{1 - addS} * srcS & \text{if } addS \geq 0 \\ (1 + addS) * srcS & \text{if } addS < 0 \end{cases}$$

2.4 个人想法

如 2.2 节所说，海康的饱和度调整有其比较好的想法，即用 Y 来代替平均值，但是它的调整权重里面有一个 0.744 奇怪数字，感觉可以去掉。即调整方式修改为：

$$R^* = R + (R - Y) \times j$$

$$\text{其中：} Y = 0.299 \times R + 0.587 \times G + 0.114 \times B$$

其中 j 表示归一化后的系数。

三、 亮度

3.1 海康的调整

海康的亮度调整简单粗暴，直接去调整曝光时间或模拟增益。我觉得这种方式显然不合理。

亮度通过 *Analog Control* 属性下的 *Brightness* 参数进行设置，参数范围为 0 ~ 255。
设置 *Brightness* 后，相机会自动调整曝光时间或模拟增益，使图像亮度达到目标亮度。
Brightness 设置的越大，自动曝光或自动增益模式下，图像调整越亮。*Brightness* 设置的越小，自动曝光或自动增益模式下，图像调整越暗。

3.2 常见方式一：HSL 空间调整

亮度 Lightness 也是 HSL 中的一个维度，所以理所当然地也会想到转到 HSL 空间，然后调整 L 维度，最后转回 RGB。这是一个可行的方案，但是 Photoshop 和海康都没有这样做，估计亮度调整很直观（亮度的定义也很直观，即 RGB 中 $(\max + \min)/2$ ），没有必要这么麻烦，还要转到 HSL 空间再去做。

3.3 常见方式二：RGB 空间调整 (Photopea 处理方式)

如上小节所述，亮度定义非常直观，RGB 中 $(\max + \min)/2$ 即为亮度，所以只要对 RGB 做同样处理，那么本质上和在 HSL 空间对 L 做处理是一样的。

有直接 RGB 乘以一个相同系数的做法，但是这种方式上下限有问题，Photopea 中做法如下（以红色为例）：

$$newR = \begin{cases} srcR + addL * (255 - srcR) & if \ addL \geq 0 \\ srcR + addL * (srcR - 000) & if \ addL < 0 \end{cases}$$

其他颜色同理。

3.4 个人想法

我觉得海康通过直接去调整曝光时间或模拟增益来调整亮度不太合理，曝光时间在线扫相机中和行频强相关，而模拟增益也和噪声有较大关联。常见方式二的调整操作比较直观方便，并且此时能和色调饱和度的操作保持一致性。