Ps共存在**27**种混合模式，并分为**6**个组，我们暂且将其总结为：

**覆盖组**：正常、溶解

**加暗组**：变暗、正片叠底、颜色加深、线性加深、深色

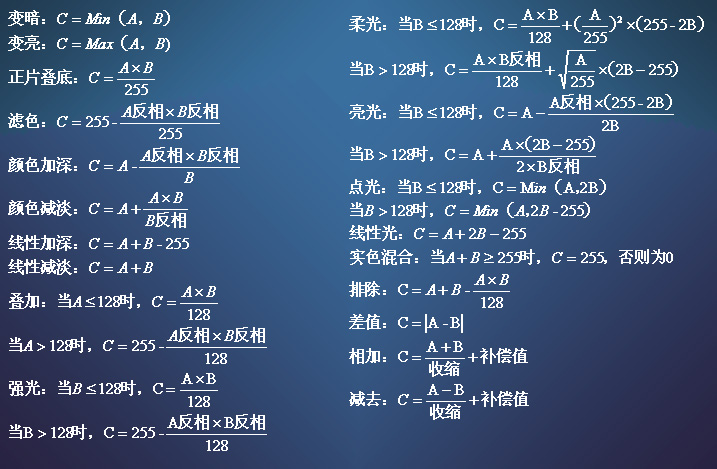
**加亮组**：变亮、滤色 、颜色减淡、线性减淡、浅色

**对比组**：叠加、柔光、强光、亮色、线性光、点光、实色混合

**差值组**：差值、排除、减去、划分

**色彩组**：色相、饱和度、颜色、明度

在网上中我们也不难搜索到相关的计算公式，其中A、B为计算层、C为结果，而在8位通道中，RGB均大于等于0小于等于255，所以0≤A、B、C≤255,下图中的计算均在此区间内计算，但为方便理解，我们将对其转换为0到1区间，文中的计算公式也将不采用下图算法，即文中ABC均已除以255进行运算。正常和溶解层不需讲述，我们按ps的排列顺序先讲解变暗层。



注：该表中点光的算法有误，B>128时，C=MAX(A,2B-255)才是正确算法。

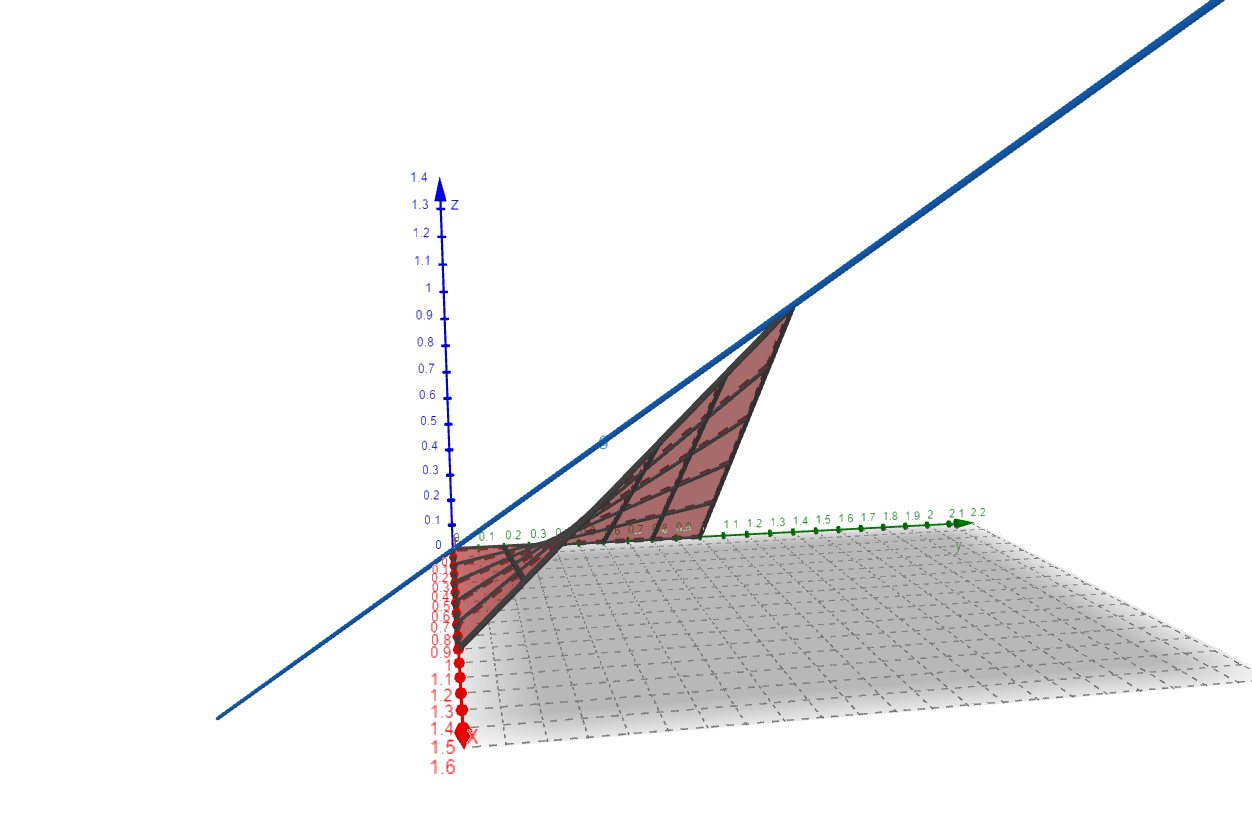
文中出现的空间直角坐标系，平面轴x、y分别代表底层A和混合层B的数值，高度z代表输出结果C的数值，即高度越高，实际画面越亮。

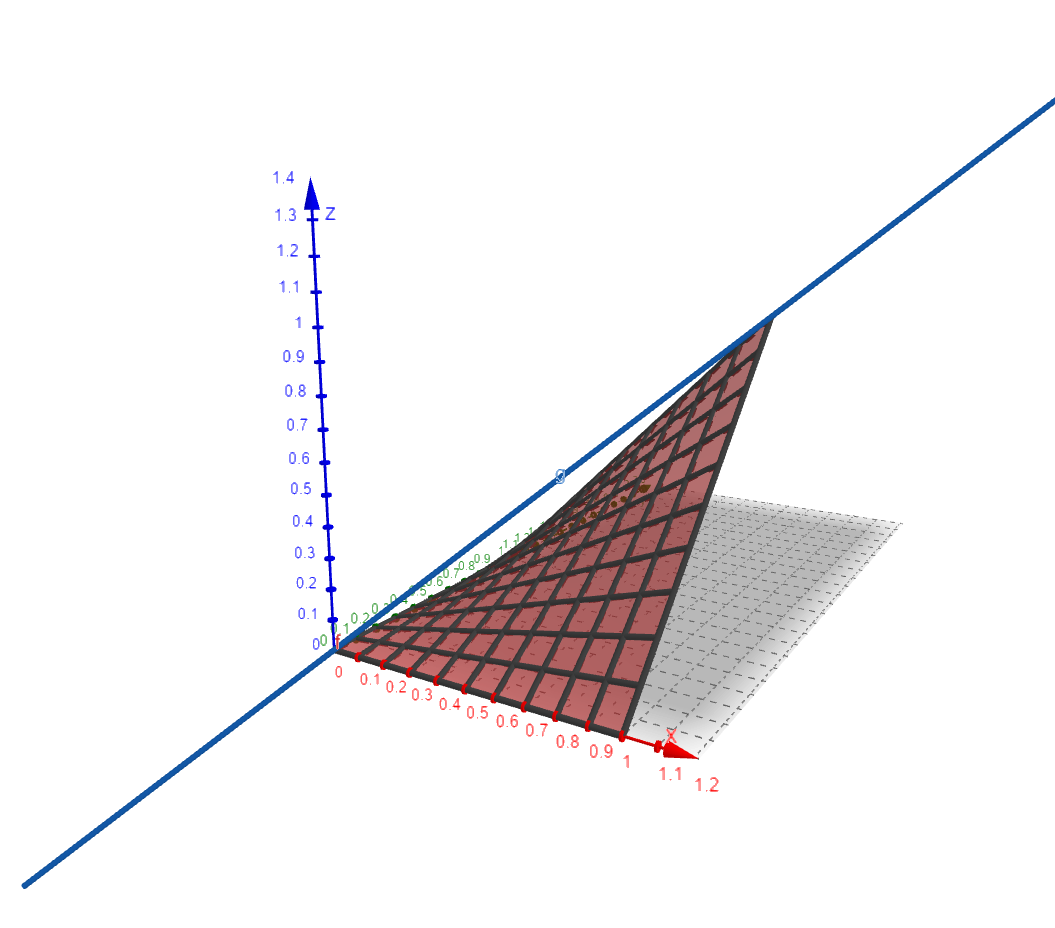
变暗：C=Min（A,B）

很方便理解，**谁数值小（即谁更暗）就取谁**，这里有一点，ABC均是RGB色彩，所以取值并不是一个维度上的，例如：A(50,50,100)和B(50,100,50),最后的输出结果为C（50,50,50）。

正片叠底：C=A\*B

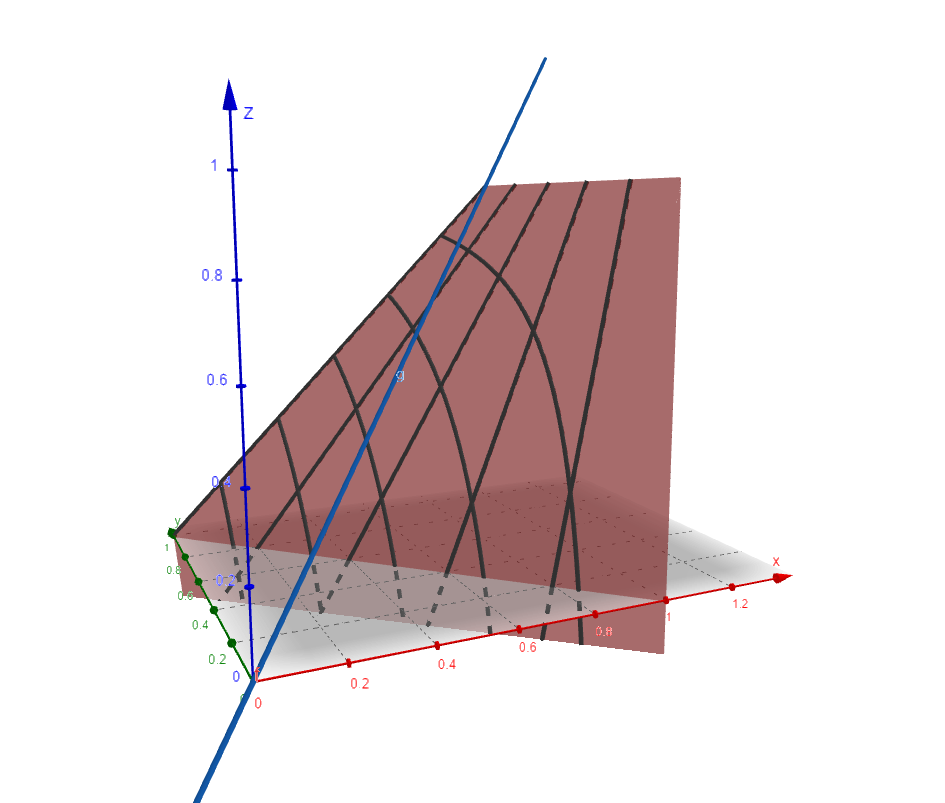
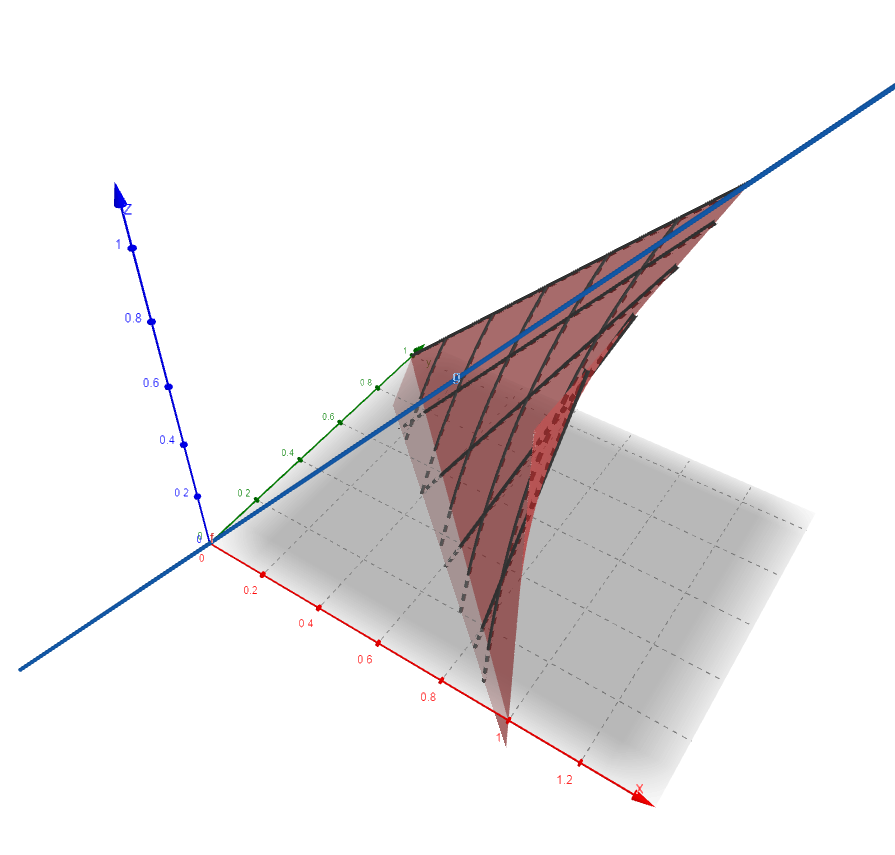
这里有一点，名称上翻译做正片叠底，实际上应该是叫做乘法。从0到1的取值区间不难得出，C一定会比A和B更暗，我们可以如此理解该混合模式：**A和B互为用作描述变暗程度的图层**，我们用三维坐标系来描述，从图中可以看出，使用正片叠底混合后，除非二者中有一者为是1，否则画面一定会变暗，且这种变暗是线性的（一次方程的导数是个固定值）。





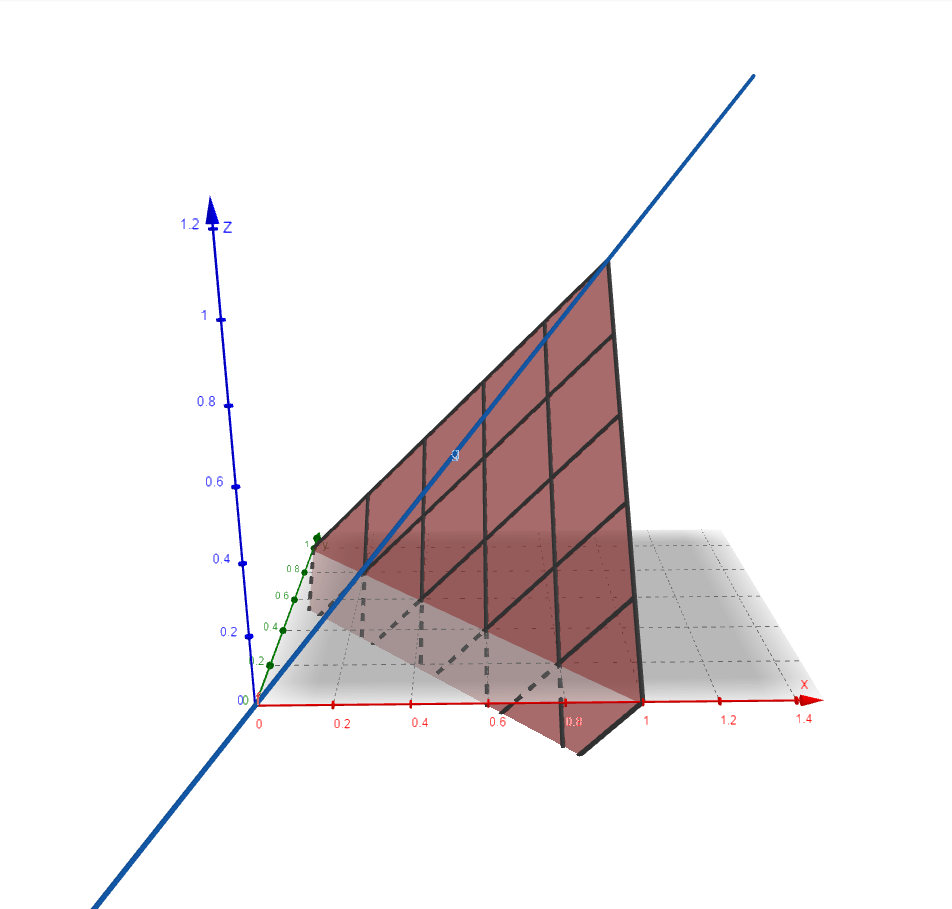
颜色加深：C=(A-1)/B+1

这个公式有出现负值的可能性，例如：A、B取0.2时，结果为-3，当然，不可能有比绝对的黑更黑的东西，**所以当A+B的总量小于1（即公式出现负值），画面会变为纯黑**，A+B总量大于1小于2时会变暗，二者均为1时不变，如此理解：**B是用以描述变暗程度的图层，变暗程度是一个二次方程，B越小，影响越大**。

线性加深：C=A+B-1

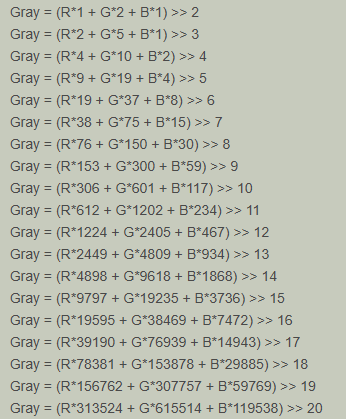
这里再一次有出现负值的可能性，同样是A+B小于1时会出现纯黑，颜色加深和线性加深的区别只在于中间的过渡，**颜色加深过渡有曲率，而线性加深如其名字，过渡是平直的**。



深色：C=Min [A(rgb),B(rgb)]

RGB通道相加算出总量，并显示总量更少的，多的不显示。

这里要注意，总量并非直接相加，而是采用RGB to Gray算法转换，算法如下：Gray = (R^2.2 \* 0.2973 + G^2.2 \* 0.6274 + B^2.2 \* 0.0753)^(1/2.2)，并非是R+G+B，在此感谢韩世麟的指点。RGB to Gray的算法有多种，下图是另一种gamma2.2时的计算方式，但在PS中不适用。

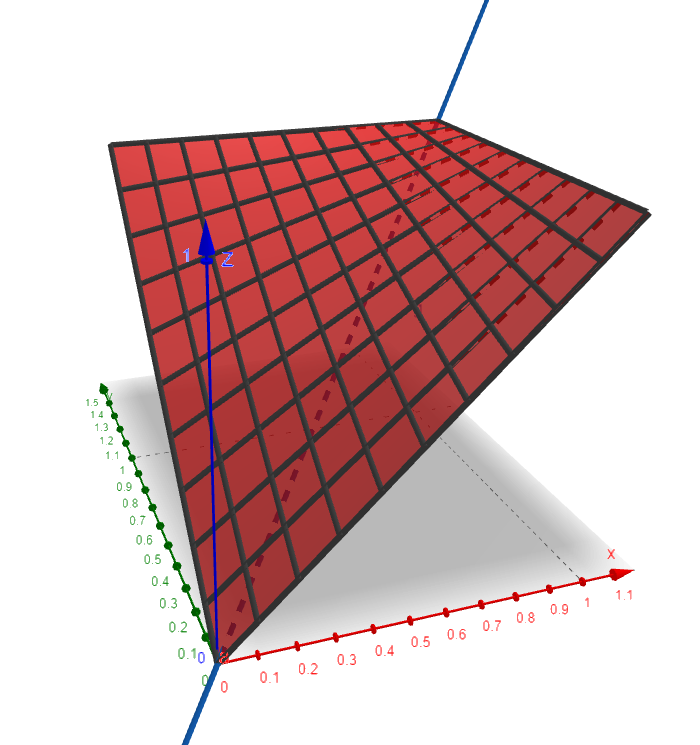
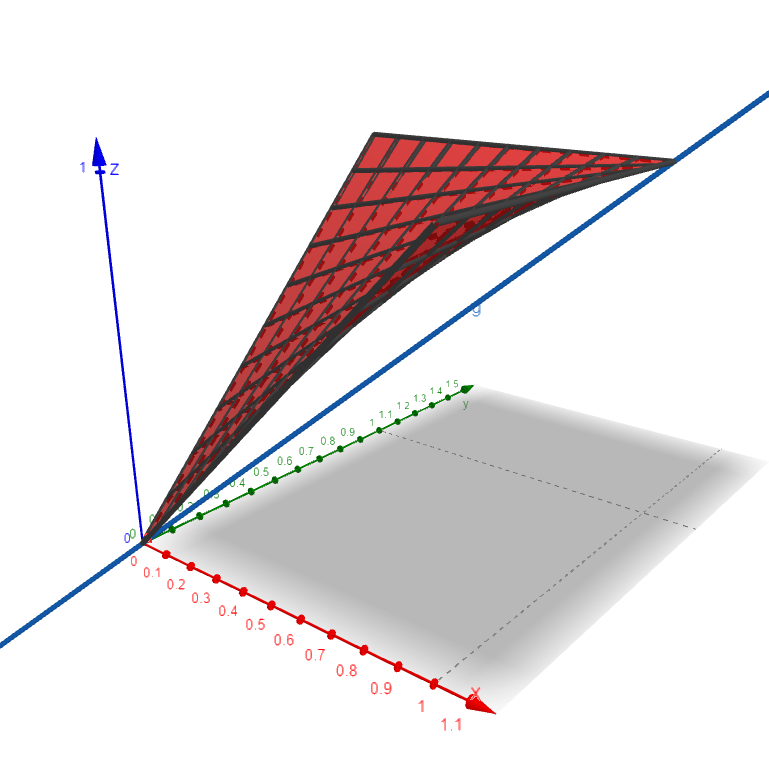


变亮：C=Max（A,B）

**谁多就取谁的粗暴算法。**

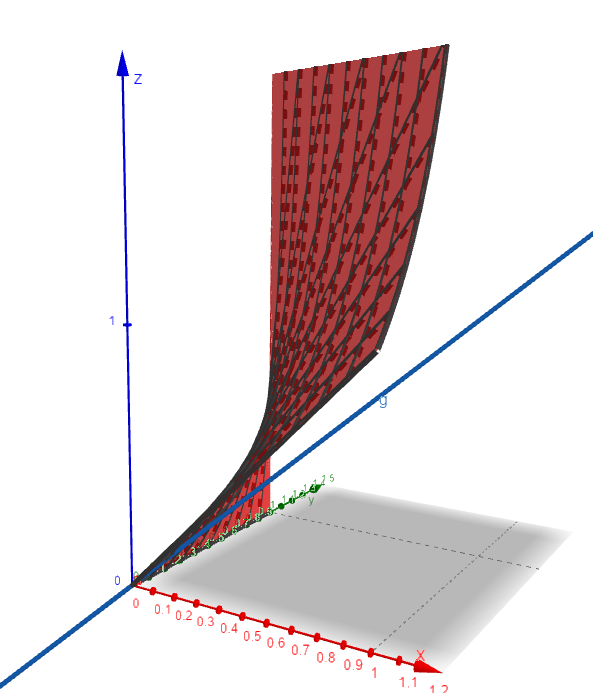
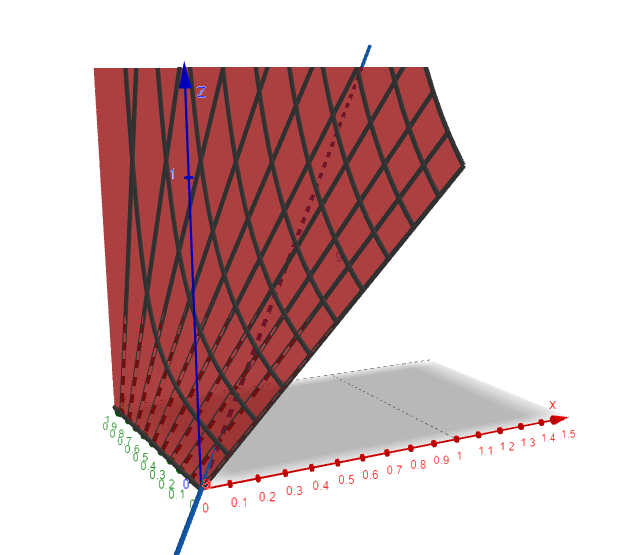
滤色：C=1-(1-A)(1-B)

参考正片叠底，A的负片乘上B的负片，最后的结果再负片一次就是滤色，**二者互为描述画面变量程度的图层。**

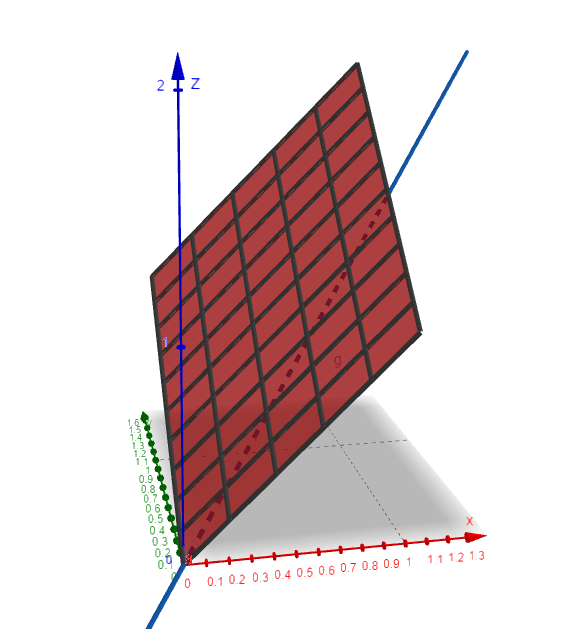
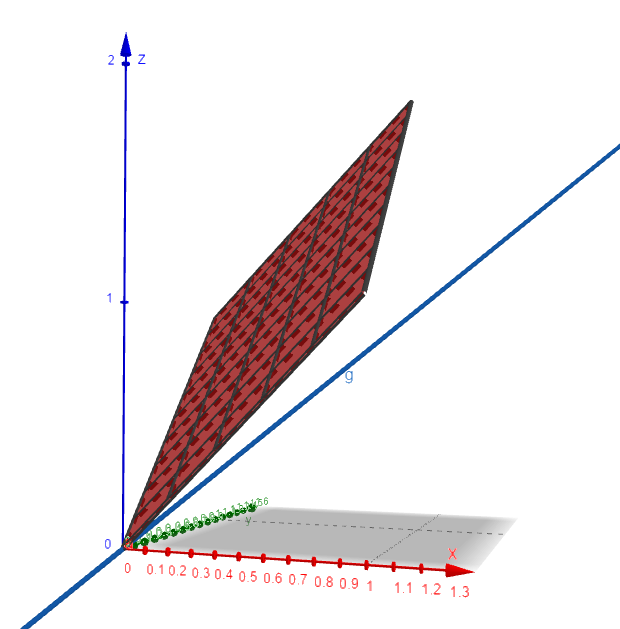
颜色减淡：C=A+AB/(1-B)=A/(1-B)

参考颜色加深，**当A+B的总量大于1（即公式结果大于1），画面会变为纯白，**其余情况均是变亮，**可以如此理解：B是用作描述变亮程度的图层，变亮程度是一个二次方程，B越大，影响越强。**



线性减淡（添加）：C=A+B

简单的加法，即便过曝也不去抑制，**常用于效果图叠加光照。**

浅色：C= Max [A(rgb),B(rgb)]

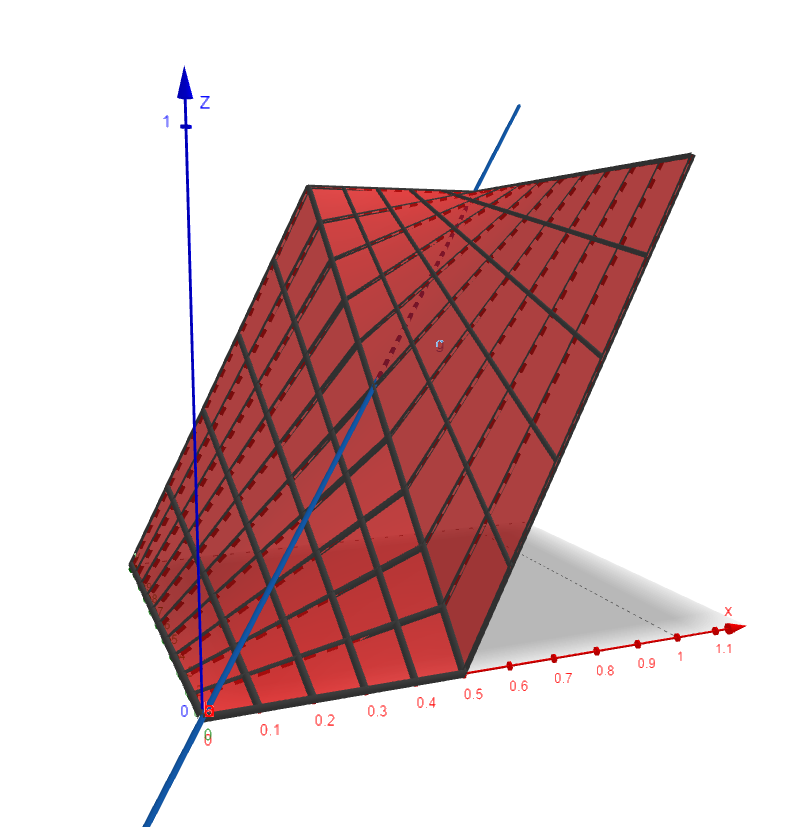
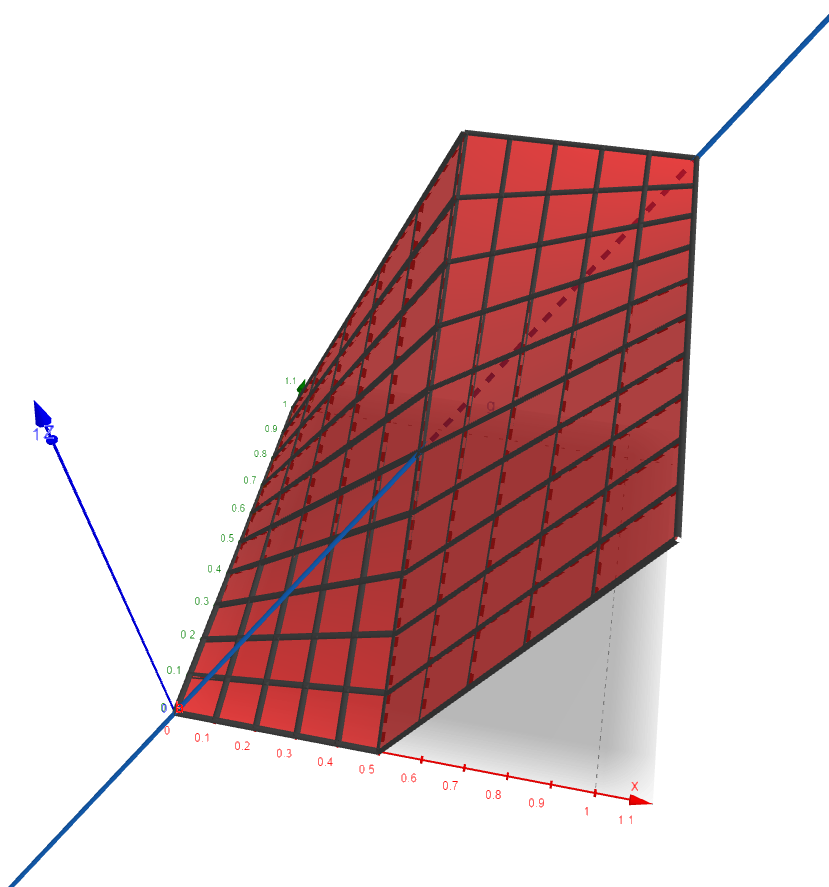
RGB通道相加算出总量，并显示总量更多的，少的不显示。

和深色一样采用RGB to Gray算法转换，详见上图。

叠加：if A≤0.5，C=2\*A\*B

If A>0.5，C=1-2\*(1-A)(1-B)

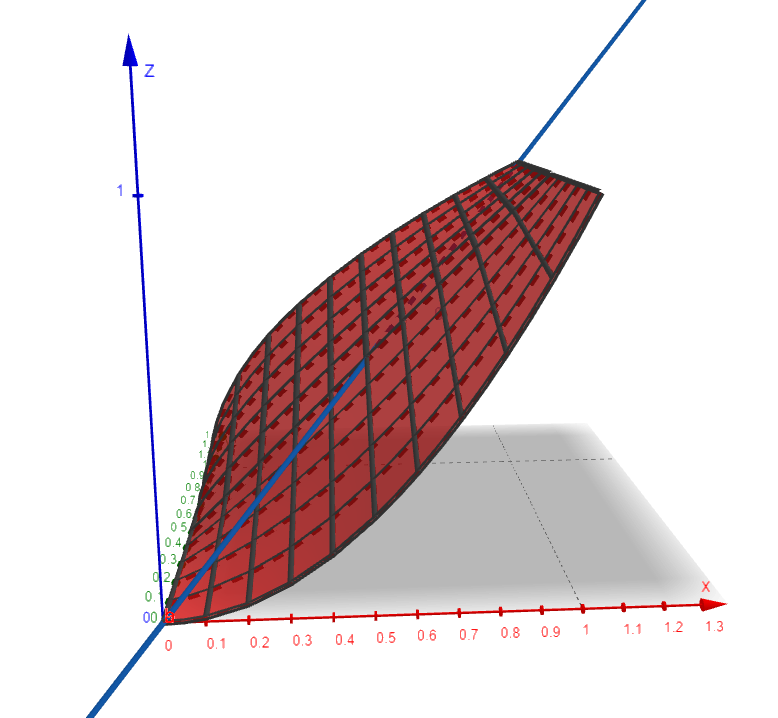
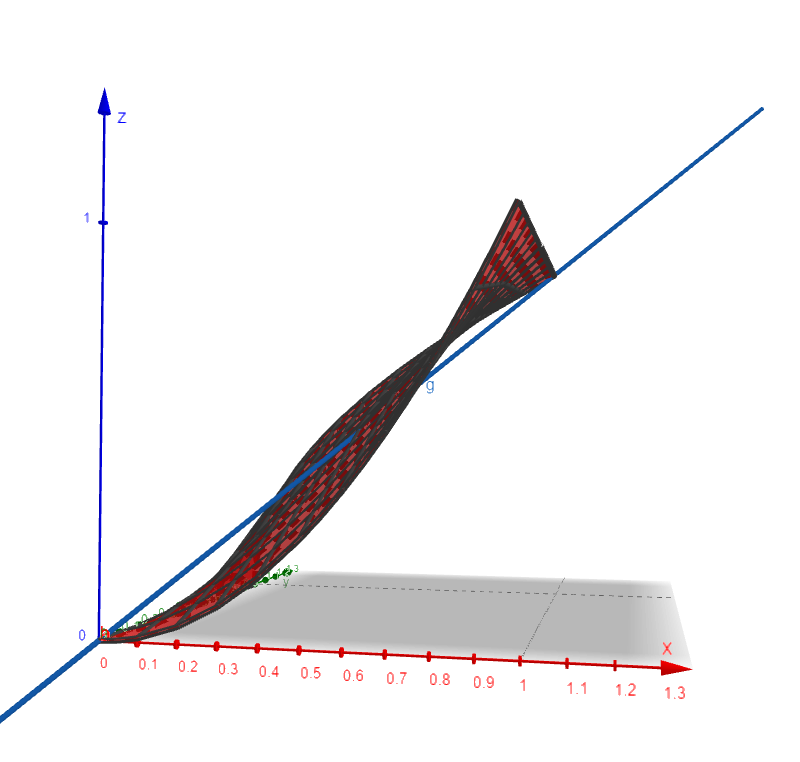
**暗更暗，亮更亮，A在中性灰前后对画面的影响力发生陡变，B对于画面的影响一直是线性的**。

柔光：if B≤0.5，C=2\*A\*B+A²（1-2B）

if B>0.5，C= 2 A(1-B)+ √A \* (2 B-1) √为根号

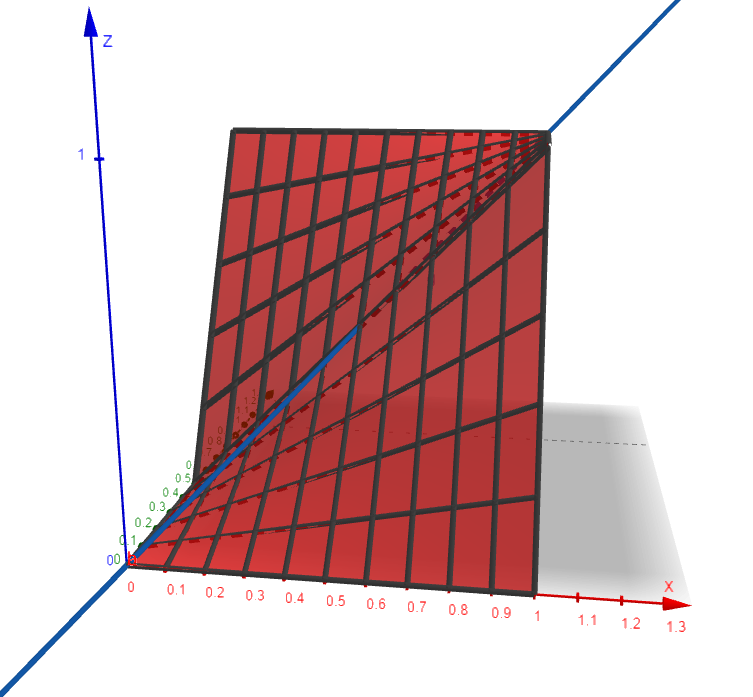
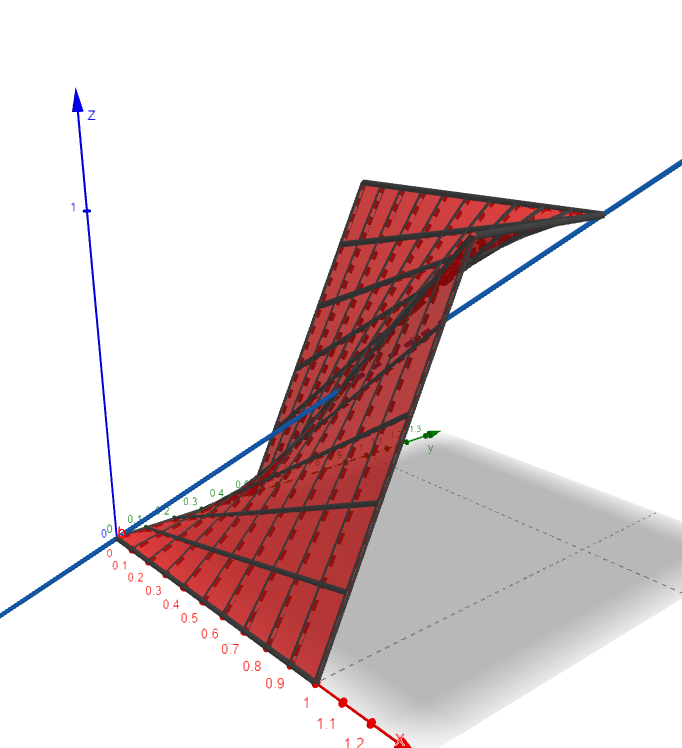
从图中可以看出，**柔光效果同叠加，但效果更平滑**。

强光：if B≤0.5，C=2\*A\*B

if B>0.5，C=1-2\*(1-A)(1-B)

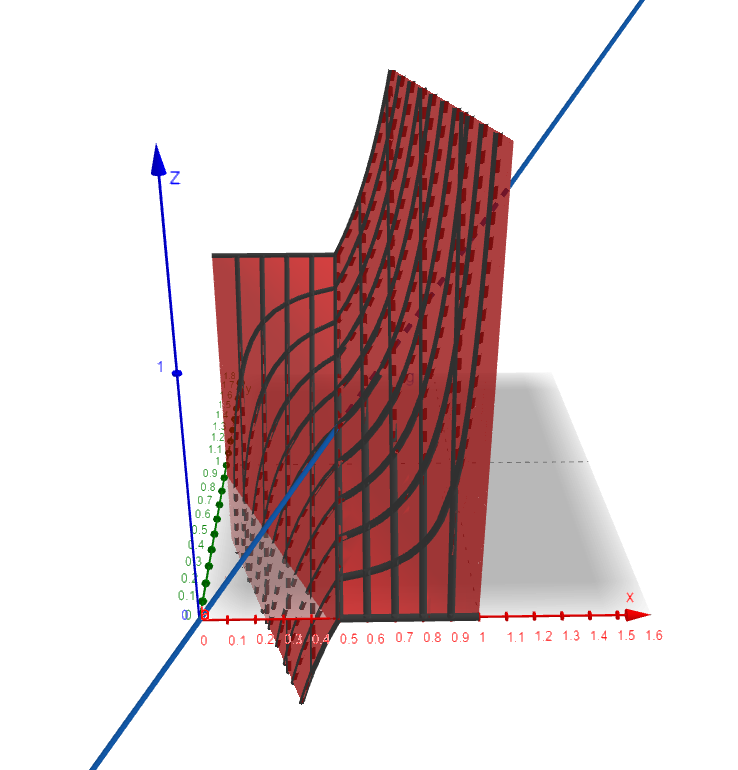
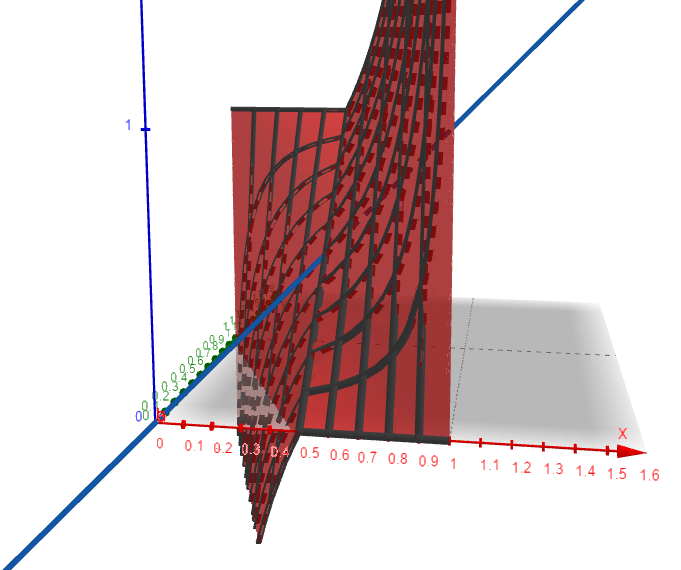
从公式也不难看出**，强光的公式和叠加一模一样，区别只在于叠加的判定是A的取值，而强光的判定是B的取值，B在中性灰区域前后的影响力发生陡变，可以理解为强光和叠加不过是A、B换了位置**。数据模型如下。

亮光：if A≤0.5，C=1-(1-B)/(2A)

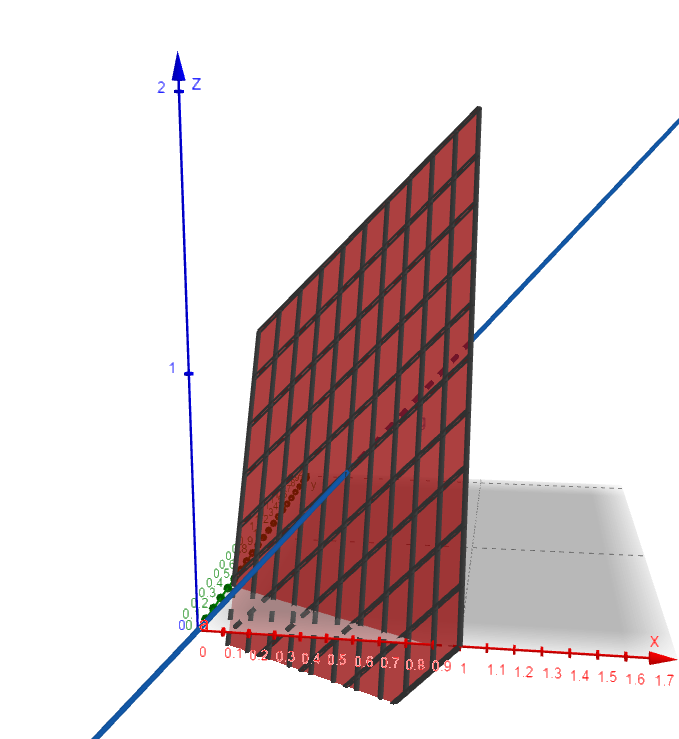
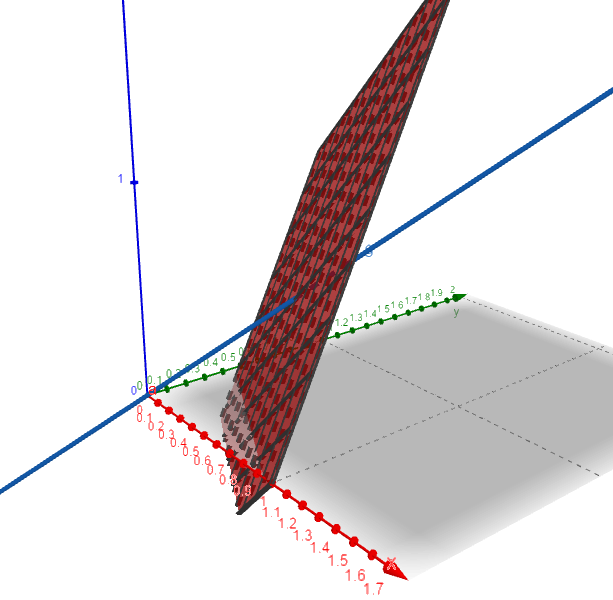
If A>0.5，C=B/(2(1-A))

同样是更暗更亮的算法，由公式得出，**B+2A<1时画面会变为纯黑，而这个叠加模式中，A离中性灰越远处影响力度越大，且过渡不平滑，B的变化则是线性的，B可以将画面压为纯黑或提亮为纯白**  。

线性光：C=A+2B-1

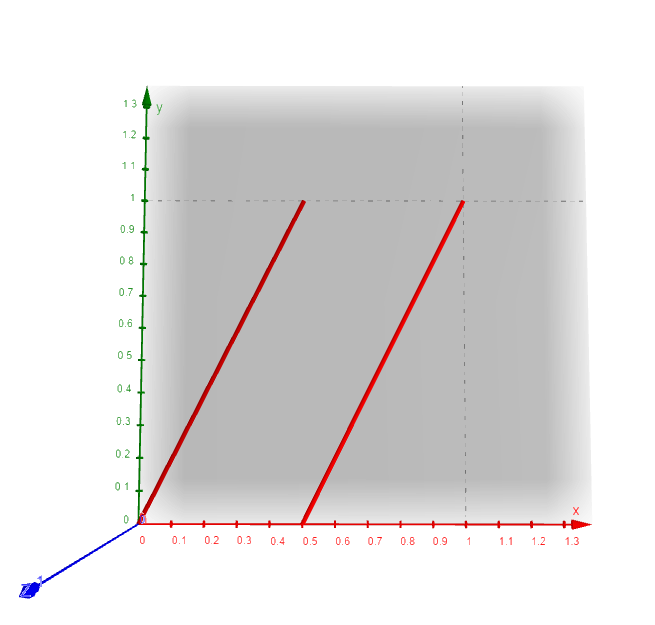
**理解为B贡献了两倍光照的线性加深**，A+2B<1时纯黑，大于1时纯白，保证了B在中性灰时不对画面产生影响。

点光：if B≤0.5，C=Min （A,2B）

if B>0.5，C=Max （A,2B-1）

纯粹的比大小,在下图中，y轴是A（同时也是输出轴C），x轴是B,**A的值确定后，B小于0.5和大于0.5会采用两种取值法，但总体而言还是加亮和压暗的逻辑，**这种模式的结果会存在强烈的断层。



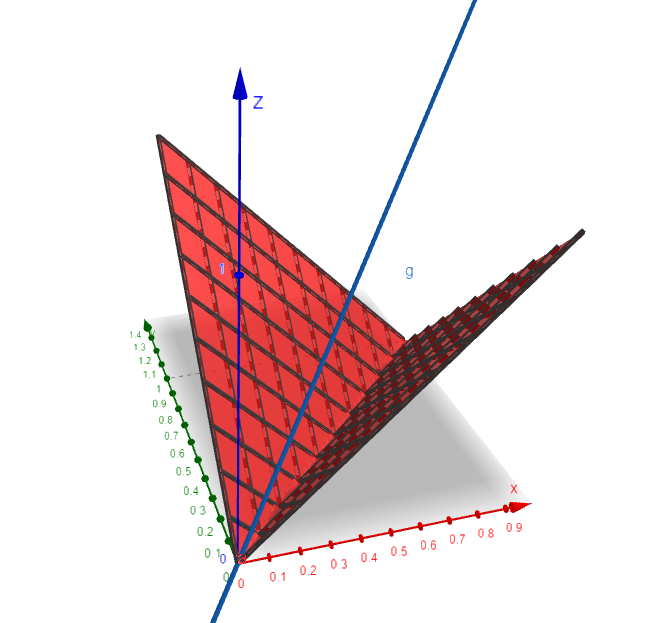
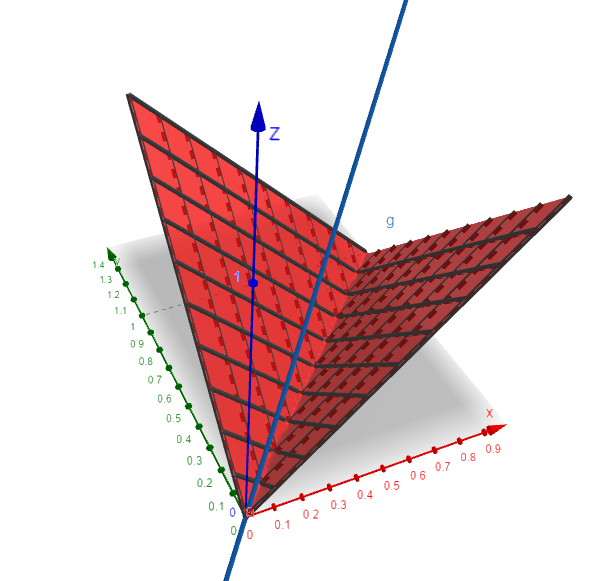
实色混光：if A+B≥1，C=1

if A+B<1，C=0

典型走极端的混合模式，**取决于A、B总和是不是大于1，输出结果不是1就是0**。

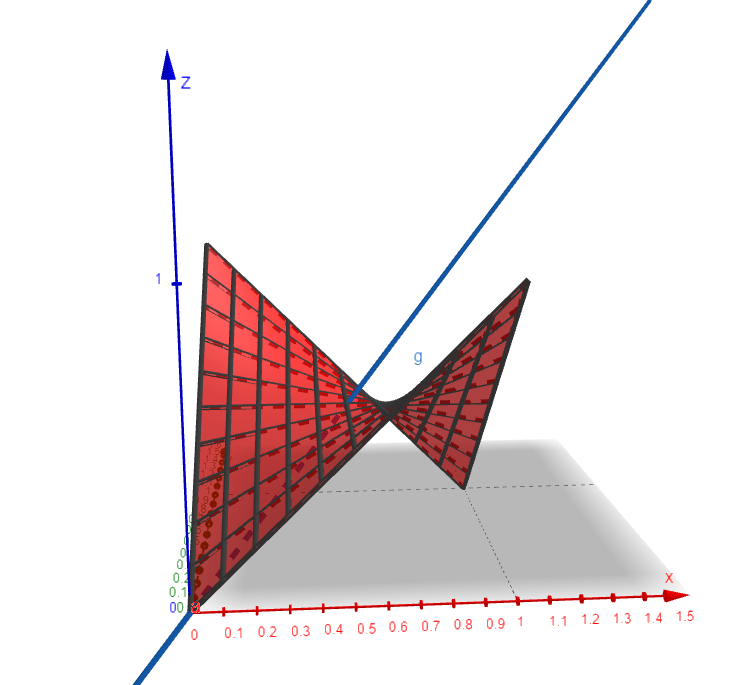
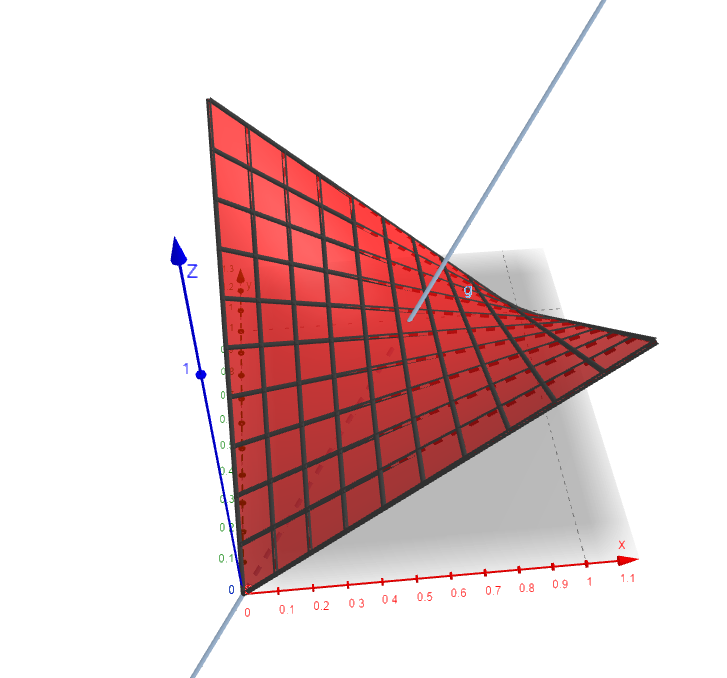
差值：C=|A-B|

从公式就能理解，**二者差值越大，颜色越亮。**

排除：C=A+B-2AB

对比上图的差值，我们不难理解，**排除是转折更平缓的一种差值**，但这种表述并不准确，因为相较于差值，**排除中A、B的色彩越接近中性灰，结果也会接近**，而差值则不受这种影响。

减去：C=A-B

**纯粹的减法，差值会有有绝对值修正负数，而减去没有**。

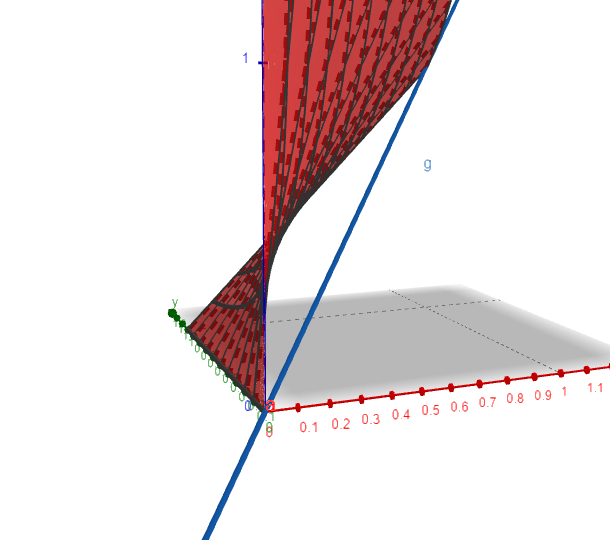
划分：C=A/B

纯粹的除法，这里有3种情况，

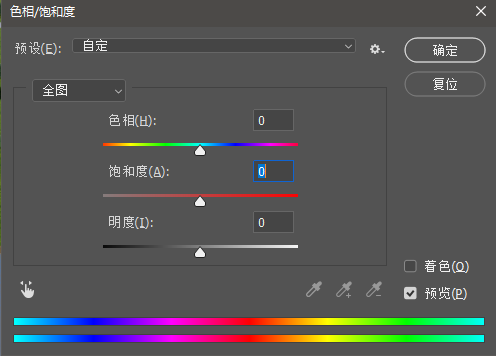
**1.A>B，则输出白色**

**2.A<B，则输出结果介于A、B之间**

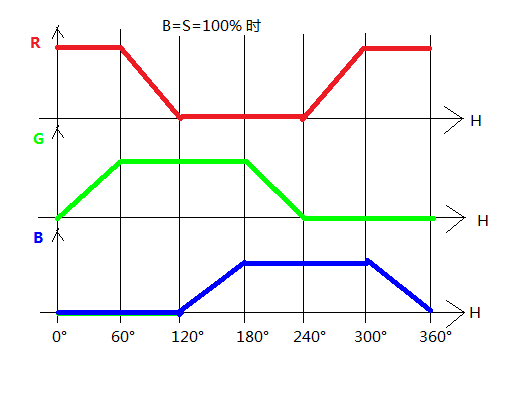
3.B为0，输出白色



在色彩组里面不再单纯以RGB为计算标准，而是以HSL为标准，（即色相Hue、饱和度Saturation、明度Luminosity）



从本质上讲，HSL和RGB完全可以相互转化，但这里不在我们的讨论范围内。



色相：只改变色相不改变饱和度、明度

饱和度：只改变饱和度不改变色相、明度

颜色：同时改变色相和饱和度、明度

明度：只改变明度不改变色相、饱和度