**颜色分级效果**

**后曝光**

将颜色值和后曝光属性相乘。用于调整场景的整体曝光度，而不影响其它效果（如Bloom）。

float3 ColorGradePostExposure (float3 color)   
{  
 return color \* \_ColorAdjustments.x;  
}

**对比度**

通过颜色从中减去均匀中灰色，然后按对比度缩放，添加中灰色来应用它，使用ACEScc\_MIDGRAY作为灰色，其中ACEScc是ACES颜色空间的对数子集，中灰色值为0.4135884。为了获得最好的效果，此计算是在Log C空间中完成的，而不是线性色彩空间。我们使用源码库文件的LinearToLogC方法将颜色从线性空间转换到Log C空间，计算完成后再调用LogCToLinear方法将颜色值转回线性空间。

float3 ColorGradingContrast (float3 color)   
{  
 color = LinearToLogC(color);  
 color = (color - ACEScc\_MIDGRAY) \* \_ColorAdjustments.y + ACEScc\_MIDGRAY;  
 return LogCToLinear(color);  
}

**颜色滤镜**

接下来是颜色滤镜，只需将其与颜色相乘即可。

float3 ColorGradeColorFilter (float3 color)   
{  
 return color \* \_ColorFilter.rgb;  
}

**色调偏移**

​ 颜色的色调通过RgbToHsv方法将颜色格式从RGB转换为HSV，将色调偏移添加到H，并通过HsvToRgb转换回来。因为色调是在0～1的颜色滚轮上定义的，如果颜色超出范围则使用RotateHue方法进行限制。

float3 ColorGradingHueShift (float3 color)   
{  
 color = RgbToHsv(color);  
 float hue = color.x + \_ColorAdjustments.z;  
 color.x = RotateHue(hue, 0.0, 1.0);  
 return HsvToRgb(color);  
}

**饱和度**

最后我们调整饱和度，首先通过Luminance方法获得颜色的亮度，然后计算方式跟对比度一样，只不过使用颜色亮度而不是中间灰度值参与计算，也不用转到Log C空间。

float3 ColorGradingSaturation (float3 color)   
{  
 float luminance = Luminance(color);  
 return (color - luminance) \* \_ColorAdjustments.w + luminance;  
}

**更多控制(代码见原文)**

**白平衡（White Balance）**

白平衡工具使调整图像的感知温度成为可能。它有两个范围在[−100，100]的属性调节滑块。第一个属性是色温，这里的参数以0为基准，可以相应地调整白平衡的冷暖偏向，小于0则色调变冷，大于0色调变暖。第二个属性是色调，用于调整温度变化后的颜色，如果说色温的色彩偏向是黄、蓝，那么这个参数的色彩偏向绿和品红，小于0时色调偏绿，大于0时色调偏品红。

**色调分离**

色调分离工具可以根据亮度值对图像的不同区域着色，可以用它给场景的阴影和高光添加不同的色调，一个典型的例子是将阴影色调调成蓝色，将高光的色调调成暖橙色。

**通道混合器（Channel Mixer）**

支持的另一个工具是通道混合器，它允许你组合输入RGB值以创建新的RGB值。例如你可以交换R和G，从G中减去B，或将G添加到R中，将绿色推向黄色。通俗来说通道混合器修改每个输入颜色通道对输出通道的总体混合的影响。例如增加绿色通道受到红色通道的整体混合的影响，最终图像的所有区域的绿色（包括中性/单色）色调会变成更红的色调。混合器本质上是一个3\*3转换矩阵，默认矩阵是单位矩阵，我们可以使用3个Vector3类型的值，用于红色、绿色和蓝色配置。

**Shadows Midtones Highlights**

我们将支持的最后一个工具是Shadows Midtones Highlights。它的工作原理类似于色调分离，不同之处在于它还可以控制中间色调，可以分别控制阴影，中间色调和高光区域。

**ACES色彩空间**

使用ACES色调映射时，Unity在ACES色彩空间而不是线性色空间中执行大多数颜色分级来产生更好的结果，让我们也这样做。