**第三章 像素运算与直方图**

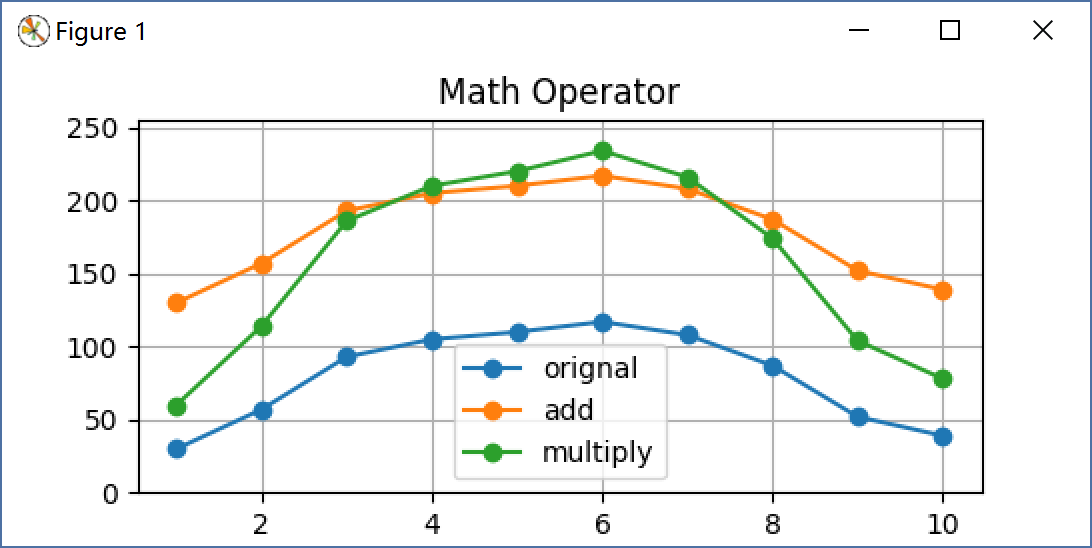
我们已经了解数字图像是由排列在规则网格上的图像元素构成，而这些元素是可以进行运算的，本章我们来讨论图像的数学运算与直方图。如果你不了解数字图像处理，但一定或多或少了解Photoshop，一定听说过色阶，亮度，对比度这些概念，通过本章的学习，我们将对这些概念由一个很好的认识。

**基础运算:**

1. 我们讨论图像维度的时候，曾经举例了一个成都市一年当中的月平均气温，这里我们将再次用一维图像进行讲解。

**$ IBook > Chapter3 Pixel Operate > Orignal Line**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 原始数据 | | | | | | | | | | oriline |
| 30 | 57 | 93 | 105 | 110 | 117 | 108 | 87 | 52 | 39 |
| 加法 y = x + 100 | | | | | | | | | | addline |
| 130 | 157 | 193 | 205 | 210 | 217 | 208 | 187 | 152 | 139 |
| 乘法 y = x \* 2 | | | | | | | | | | mulline |
| 60 | 114 | 186 | 210 | 220 | 234 | 216 | 174 | 104 | 78 |



**观察描述：**通过观察我们发现原始数据的值在（30，117）之间，图像整体偏暗。而加100之后，图像的值位于（130，217）之间，图像整体偏亮。而乘以2之后，像素值位于（60，234）之间，图像亮暗视觉差异更明显。像素值的最大，最小跨度称之为动态范围，一般来说，动态范围越大，视觉冲击力就越强。

**充分利用像素值区间**

依旧以上面的例子来讲解，现在提出这样一个问题，如何用，让原始信号尽可能填满像素值区间。

**问题分析：**现在像素值现在分布在（30，117）之间，而8位灰度图像的可能分布区间在（0，255）之间，我们已经知道，增加动态范围能有效增加图像的视觉对比度，那么如何让原本在（30，117）之间的像素，充满（0，255）呢？

这个问题单纯的加法，乘法都不能实现，所以需要用一个组合运算。其实这使一个一次函数问题，我们其实是要找到一个**映射f，满足f(30)=0, f(117)=255**. 我们知道两点确定一条直线，于是设定解析式 y = k x + b列出方程组:



当然也可以直接采用两点表达式：

(y - 0) / (x - 30) = (255 - 0) / (117 - 30)

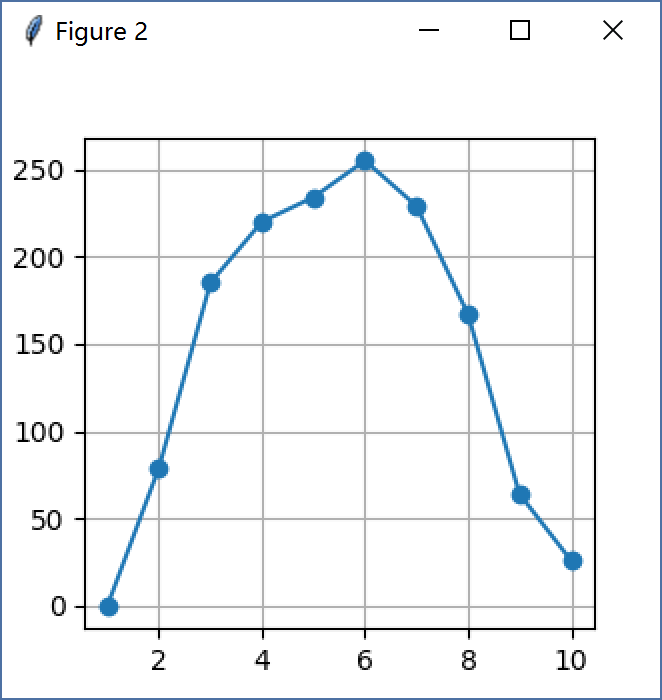
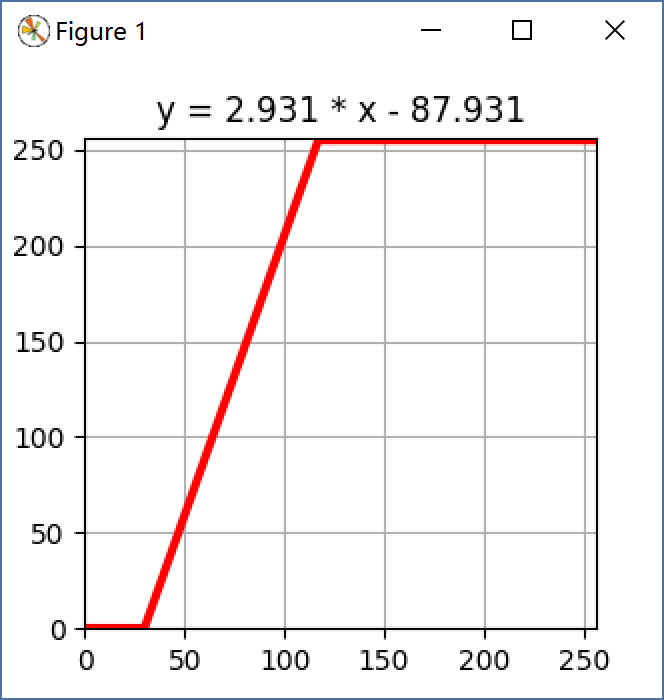
求解得：y = 2.931 x - 87.931

**$ IBook > Chapter3 Pixel Operate > Extend Full**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 变换后数据 | | | | | | | | | | extend |
| 0 | 79 | 185 | 219 | 234 | 255 | 229 | 167 | 64 | 26 |

变换后数据 变换后图像

**观察描述：**经过上述变换之后，我们可以看到数据最小值使0，最大值成了255，这样正好充满了一个8位灰度图像的可行域，而观察其对应的图像，此刻达到了黑白分明的效果，给我们更好的视觉冲击力。

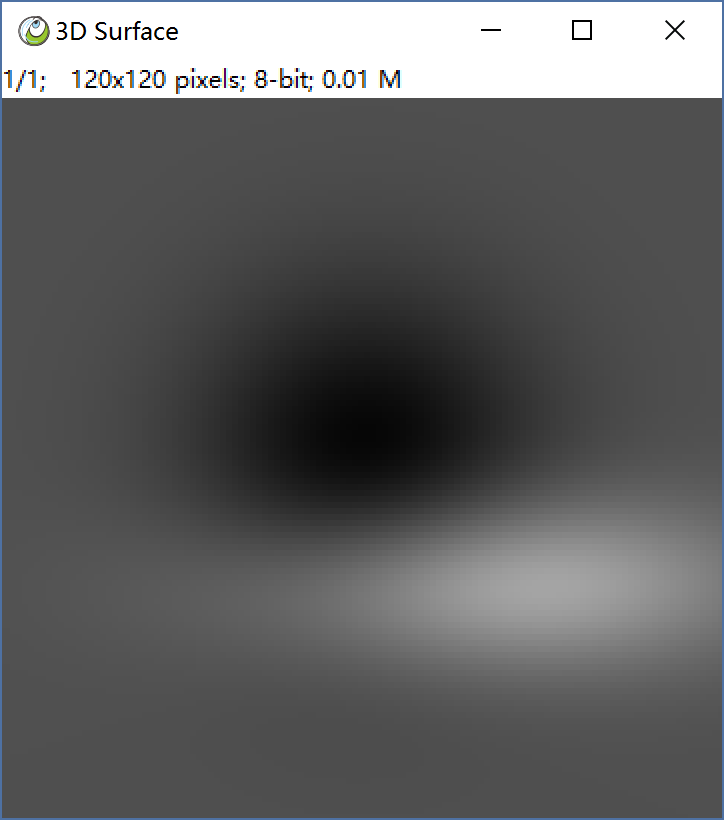
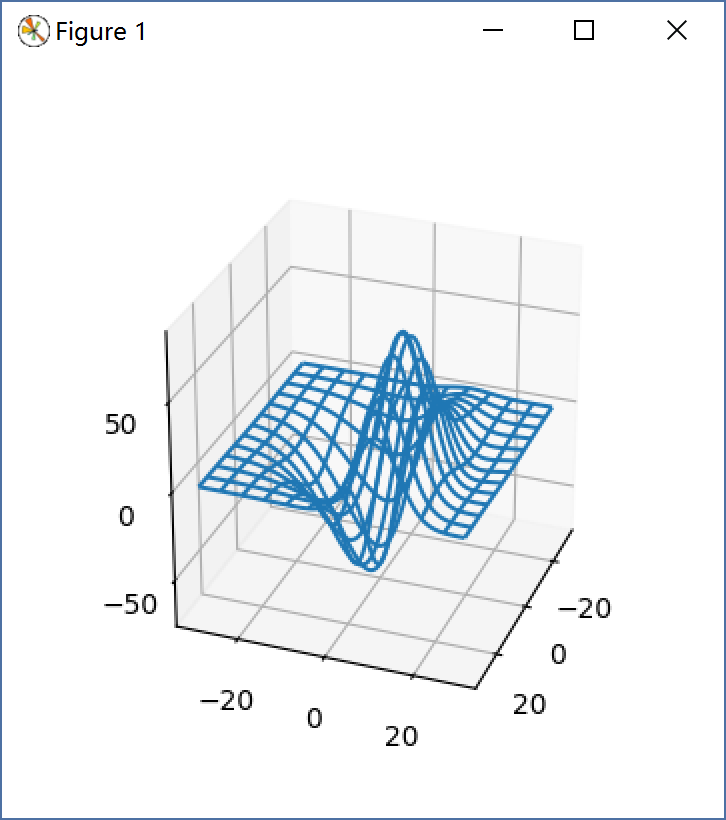
变换后图表 映射函数

**观察描述：**此时数据图表上的效果也充满了坐标系，而观察映射函数的图像，经过（30，0），（117，255）点，其实这个图像告诉我们如何根据原图的像素值计算出新值，比如我们观察图像大概经过（100，200）这个点，意味着如果原图中某个像素值是100，那么在新图上对应位置像素值会是200。

**二维图像与直方图**

对于以上一维的少数点，我们可以用图表绘制出来，而对于二维的图像，我们很难用一张图表将其展现，当然如果一定要展现是可以的，你可以把图像想象成一个起伏的地表，亮度代表高程，事实上这种模型在GIS中很常用，（DEM）数字高程模型，数字高程模型是对地表信息的完全表达，而纸质地图上，我们更常见的是一种简化了的表达，等高线。我们这里并不是着重介绍某个专业，但DEM的概念的确对理解数字图像有很大帮助，所以我们在此给出，就像下面这样。

**$ IBook > Chapter2 Math-Operator > 3D Surface**

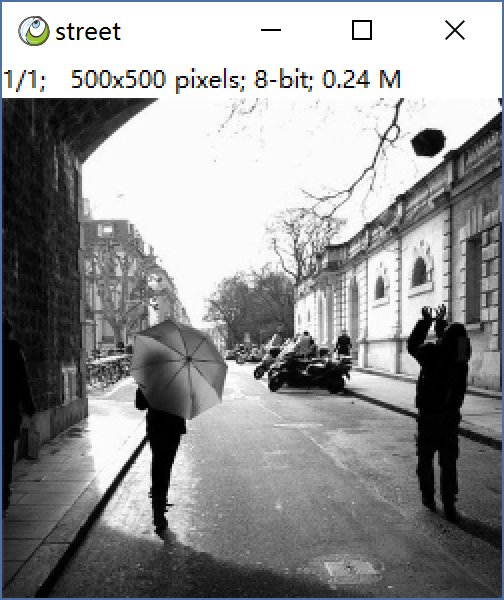
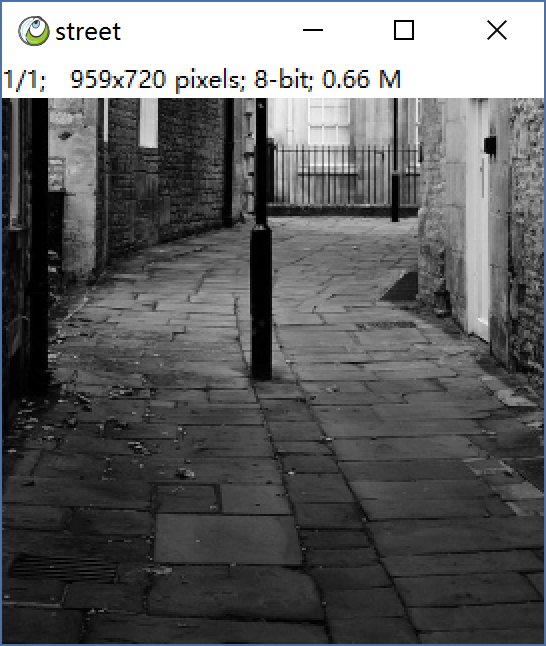
 

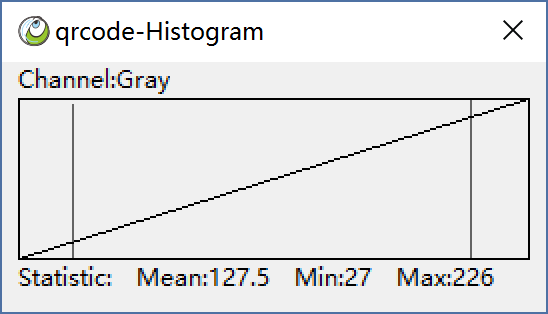
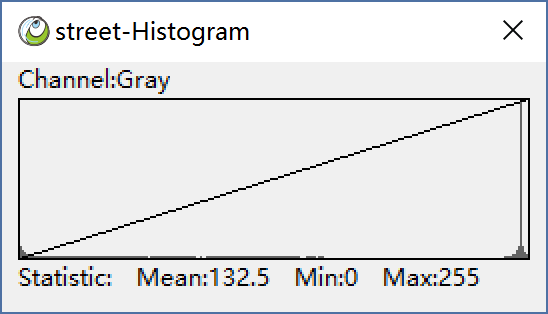
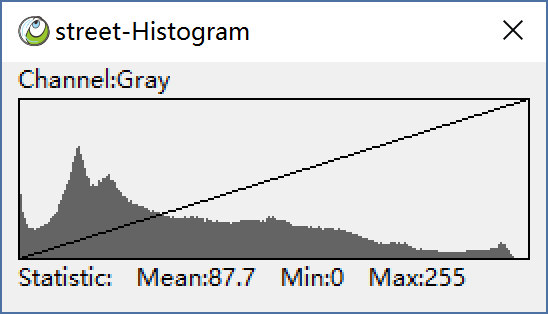
一副数字高程图像 对其进行表面绘制

**像素分布直方图**

不过三位表面图有它的局限性，因为绘制太复杂，并且复杂情况，肉眼判读也不是很容易，所以我们经常对其进行频率统计，得到像素分布直方图。其实和我们初中学习统计时的直方图别无二致，只是这里有256个投票箱，每个像素根据自己的像素值投票给对应的格子。

**从直方猜测图像整体效果 $ IBook > Chapter3 Pixel Operate > Histogram Of GrayImage**

二维码 道路 石板街道

直方图描述了图像的整体亮度分布，上方的二维码，所有颜色集中在黑，白，并且比例大致相等，因此得到两条分立亮线。第二幅图，道路，观察图像，从暗到灰像素分布比较均匀，但在亮部出现了一个孤立峰值，可以猜测那是大片的背景天空形成的，而石板街道，图像整体偏暗，但各个亮度都有一定的分布，因而直方图相对平滑，但偏向于左边。

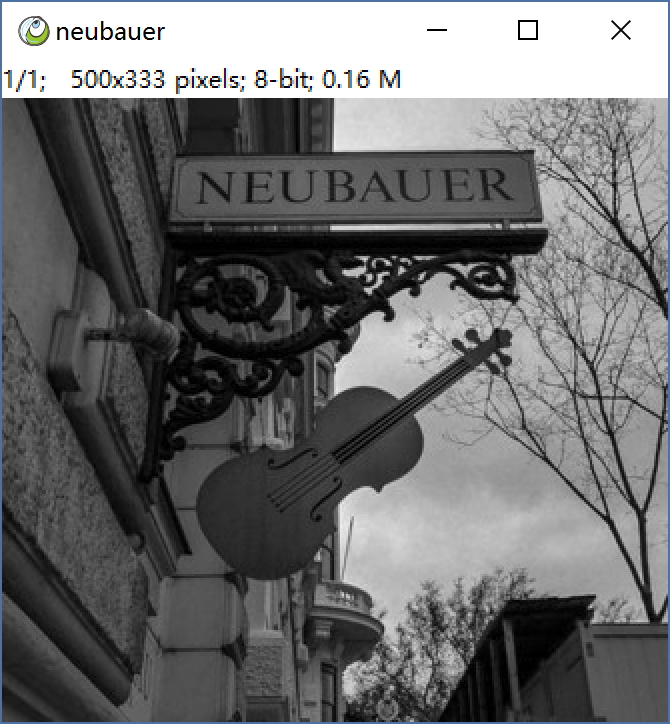
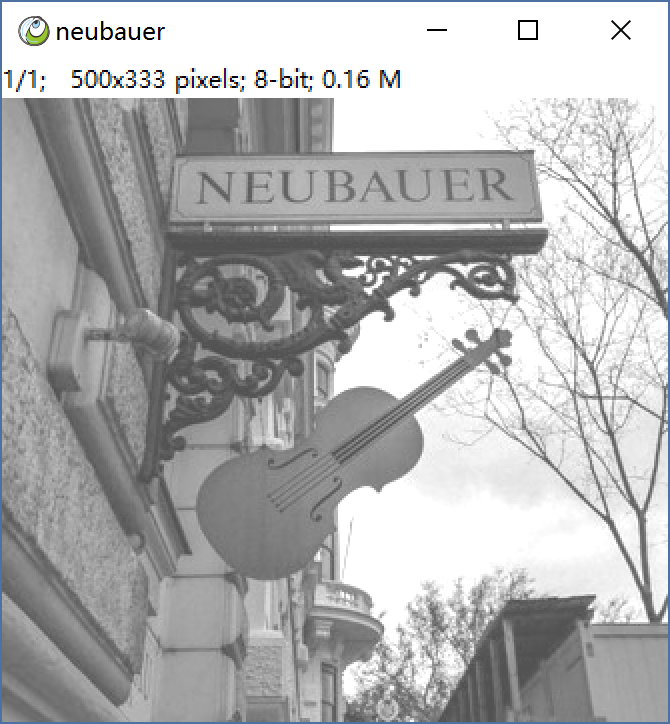
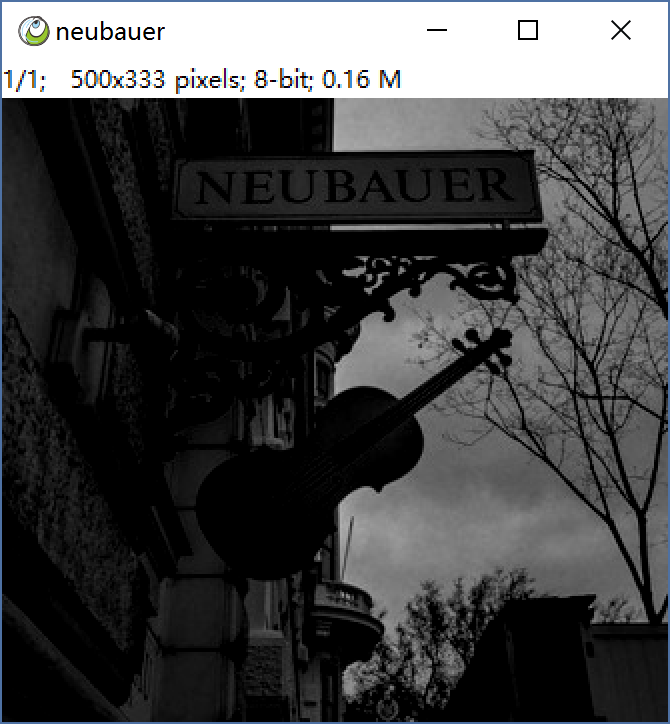
**亮度对比度色阶**

如果你熟悉Photoshop，那你一定很熟悉亮度，对比度，色阶这些词语，而通过本章的学习，我们将对亮度，对比度，色阶有一个较为深刻的认识。

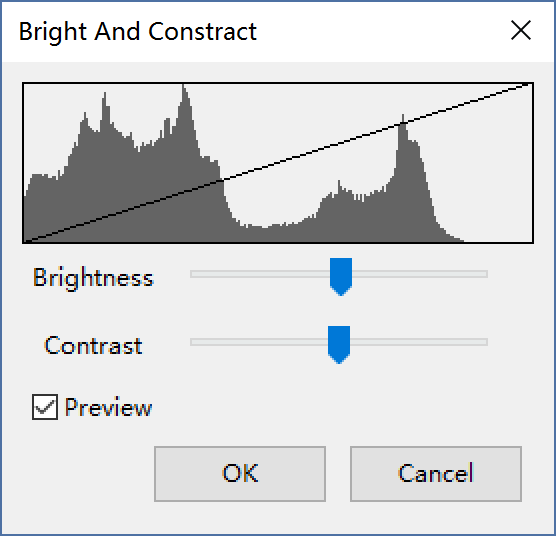
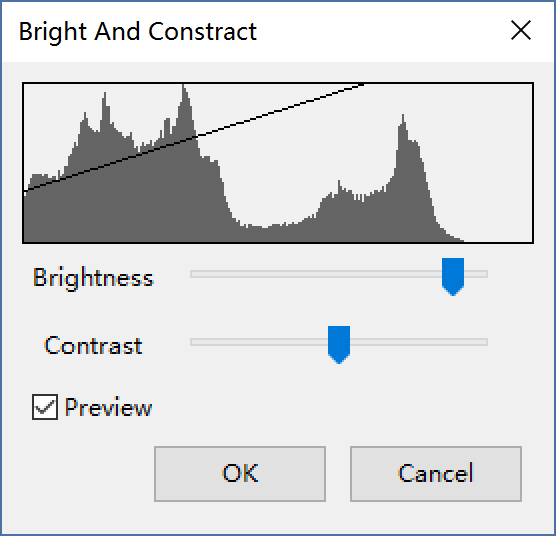
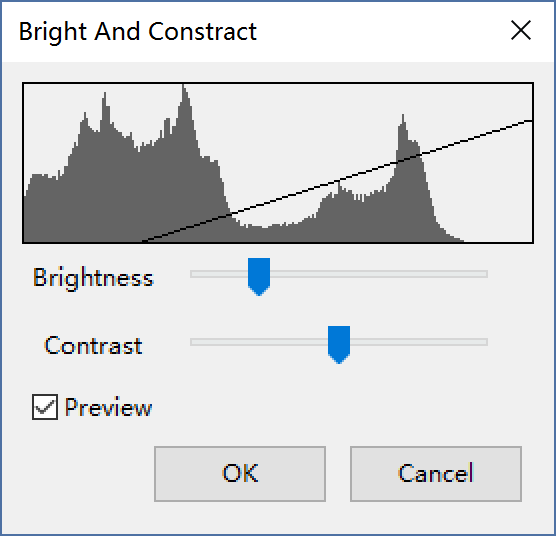
**亮度调整**

**$ IBook > Image Referenced > neubauer**

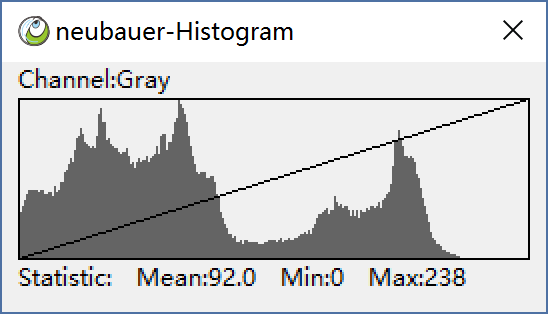
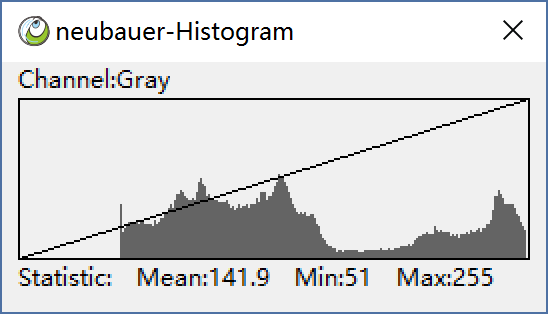
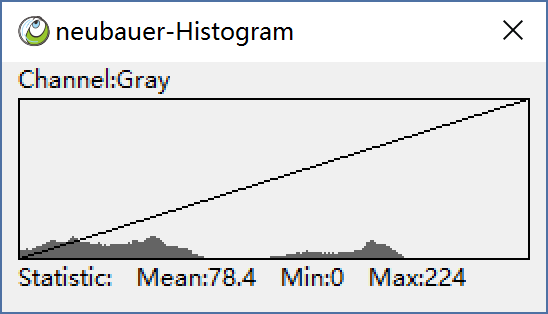
**$ Image > Adjust > Bright And Constract**

参数对话框

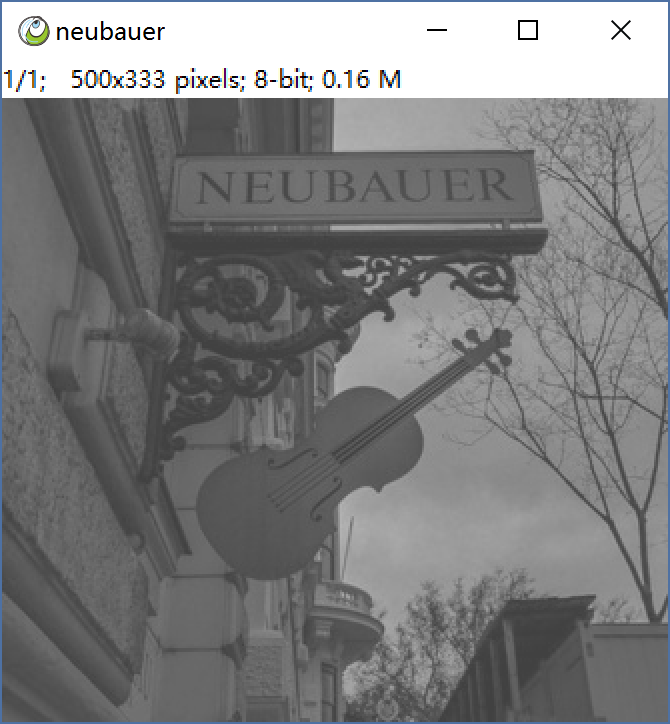
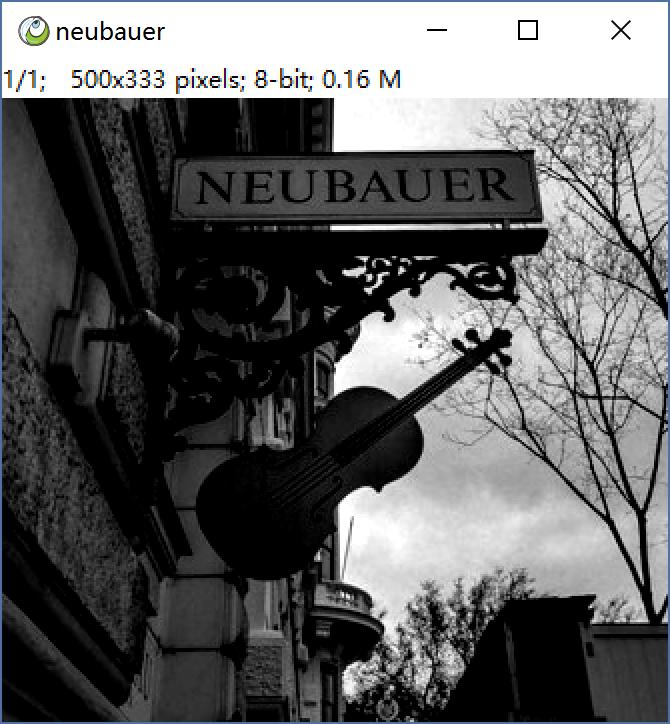
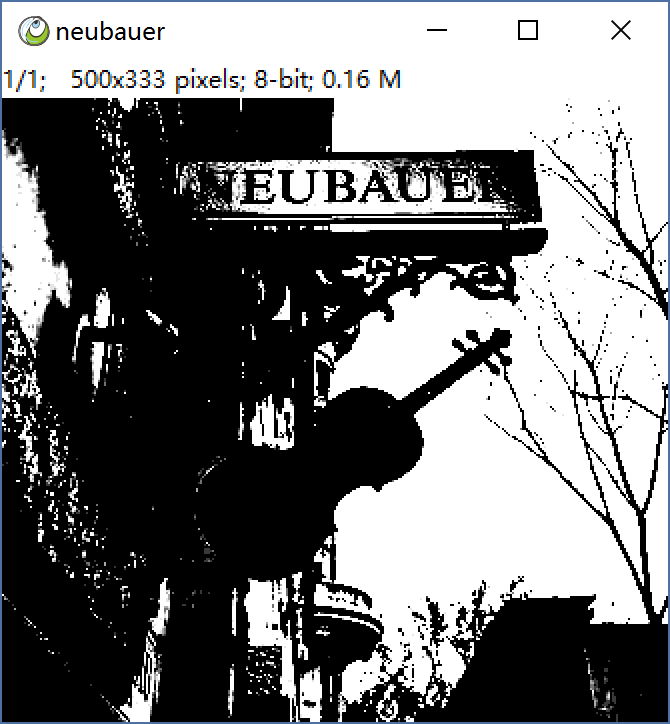
调整后直方图

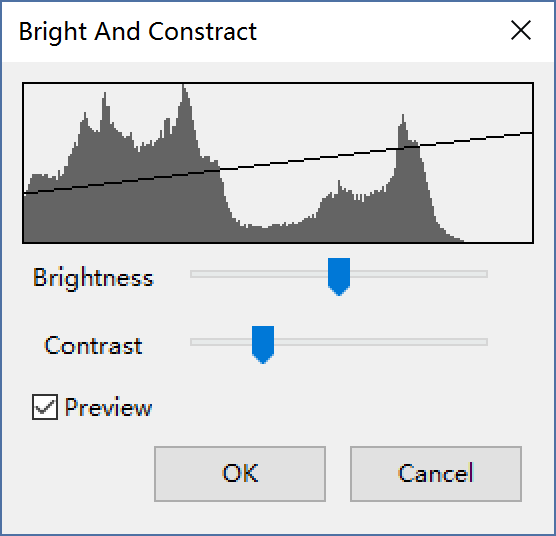
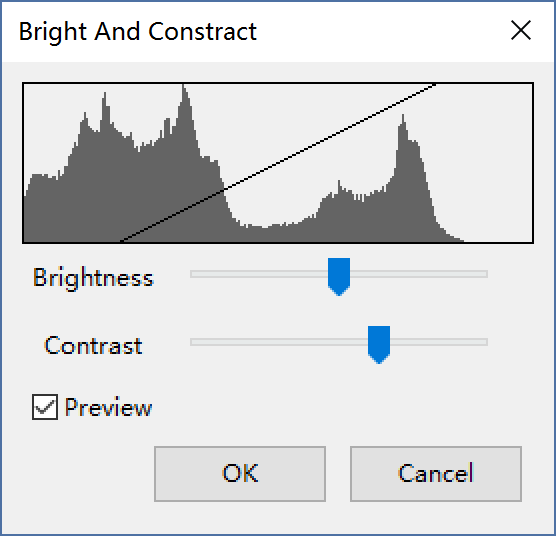
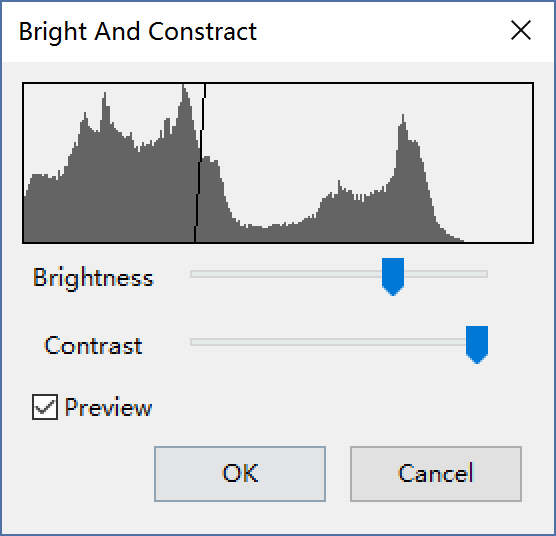
  

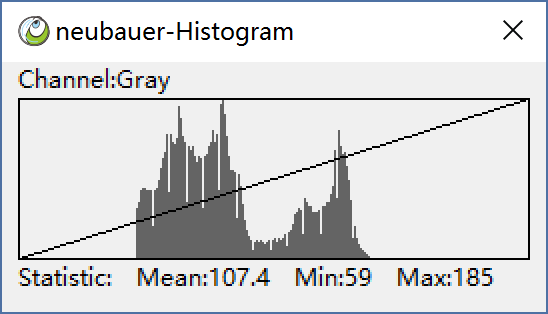
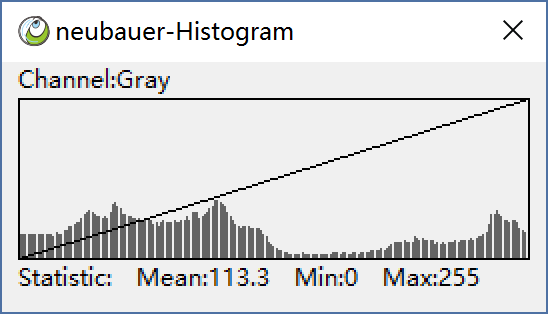
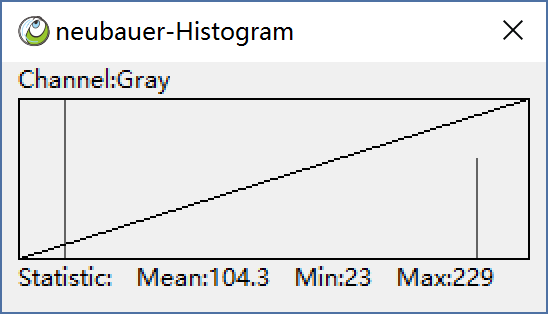
原图 提高亮度 降低亮度

**调整亮度的本质**是所有像素亮度的提高，就是我们之前学习的图像加法，降低亮度的本质就是图像减法。我们观察提高亮度后的直方图，大体上时原直方图向右移动了一定举例，然而整体似乎被压低了，这其中的原因是，由于8位图像最大值是255，当亮度增加，超出255的值就都成了255，导致255处出现累计峰值，直方图根据峰值自动缩放，导致整体被拉低。同理，降低亮度也被压低，甚至更严重。

**对比度调整**

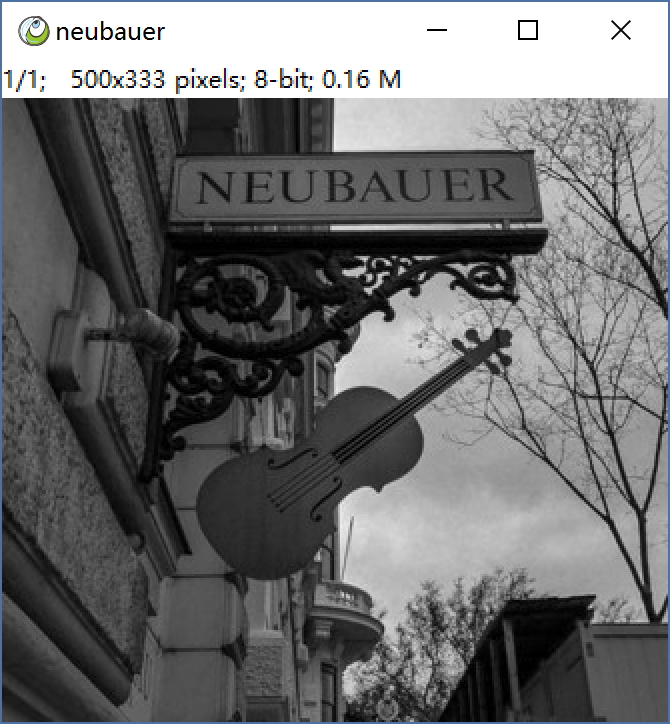
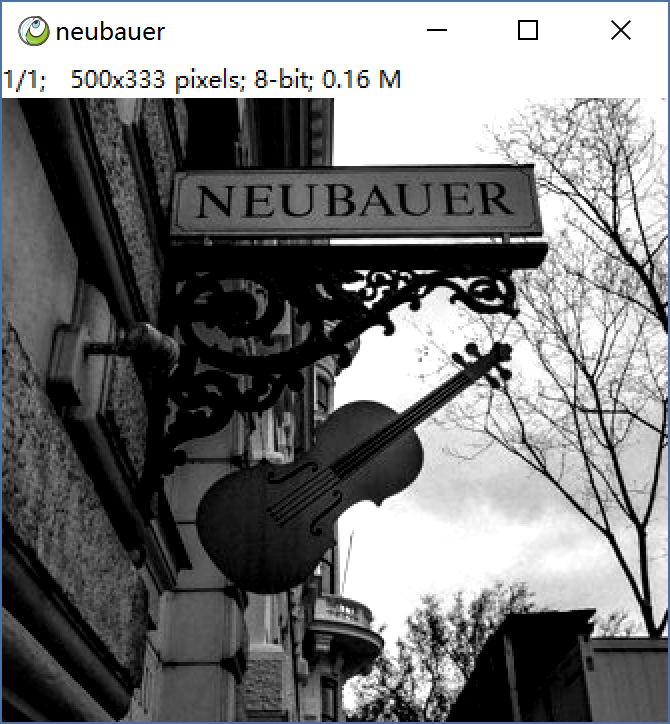
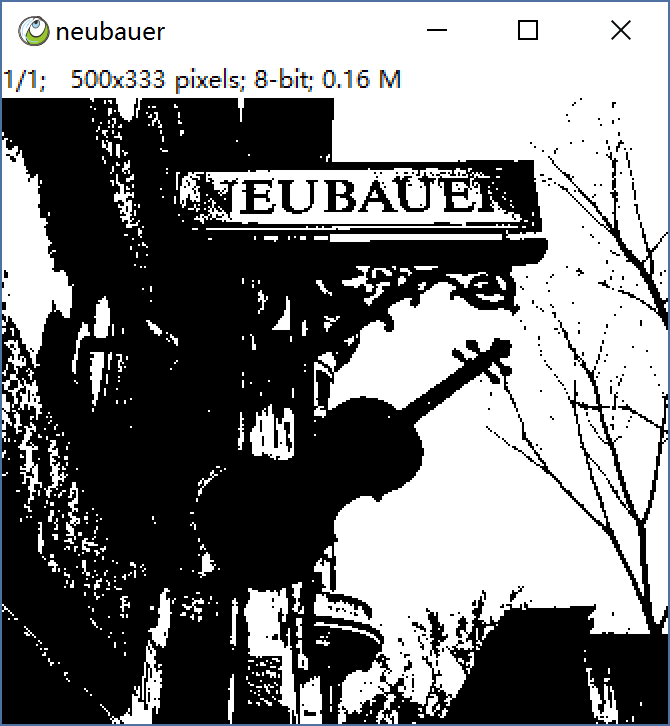
降低对比 提高对比 最高对比

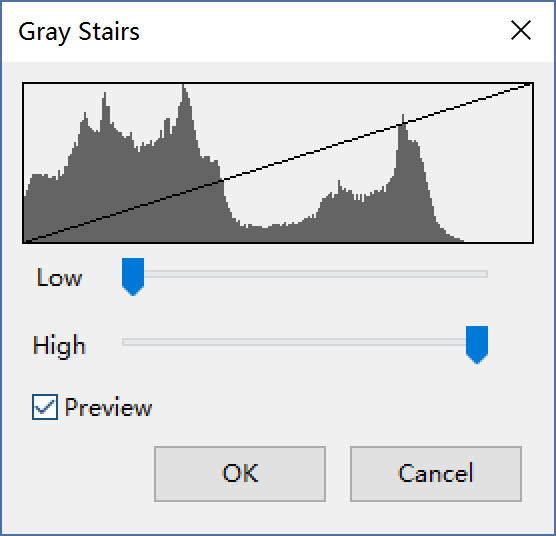
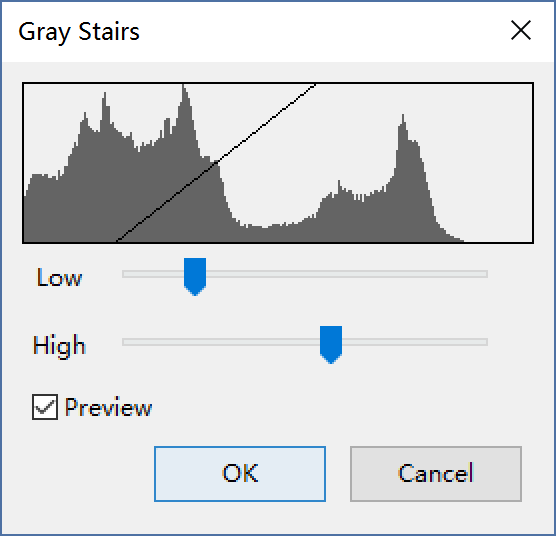
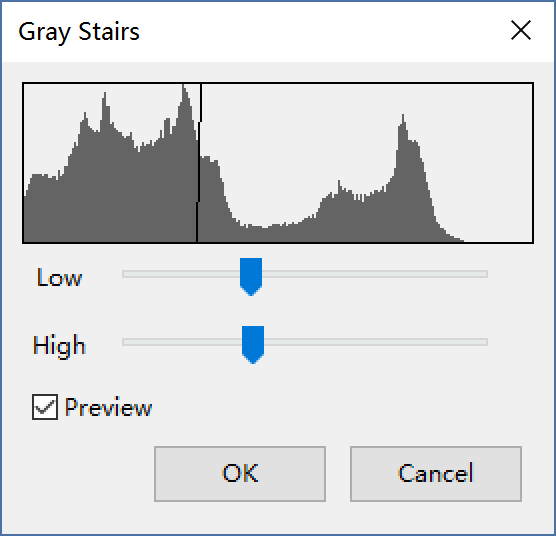
**调整对比度的本质**是像素乘法，当系数大于1时，对比度提高，当系数小于1时对比度降低，其实顾名思义，对比提高即增加差异性，我们看第一幅图，对比度降低，直方图向中间靠拢，图像的动态范围降低，因此视觉效果对比度下降。而对比度增加之后，我们观察直方图向两边扩张，并且中间出现了缝隙，其原因时当像素乘以大于1的数之后，四舍五入，自然灰形成空隙。而最后一幅图，对比度极端化之后，函数曲线几乎垂直，而结果直方图也分立成两个值（原本应时0，255，但这里为了展示清晰，做了其他处理）

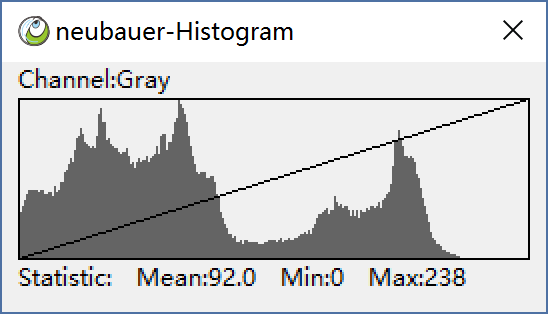
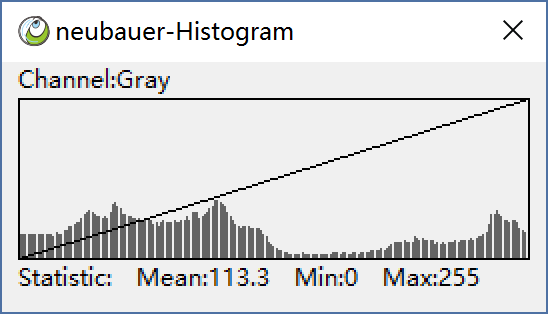
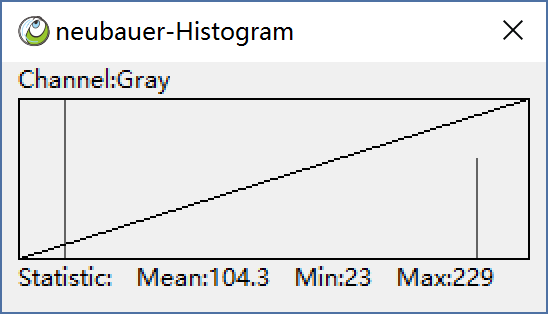
**色阶调整**

**$ IBook > Image Referenced > neubauer**

**$ Image > Adjust > Gray Stairs**

原图 色阶调整 极限情况

**色阶的本质**很类似于我们之前让像素值充满可行域的例子，只是这里由用户指定上限和下限，然后进行像素值的线性映射。

**总结**

经过以上学习，我们对亮度，对比度，色阶有了一个理论上的认识，其实一定程度上，亮度对比度和色阶是可以相互等价的，本质都只是一个 y = k x + b 形式的像素映射。简单说，b决定了亮度，k决定了对比度。

**直方图均衡化与直方图匹配**

我们以上进行的亮度，对比度，色阶都是基础线性表达式，接下来我们介绍的直方图均衡化，是一个相对复杂的运算，目的是让直方图呈现出我们希望的样子。

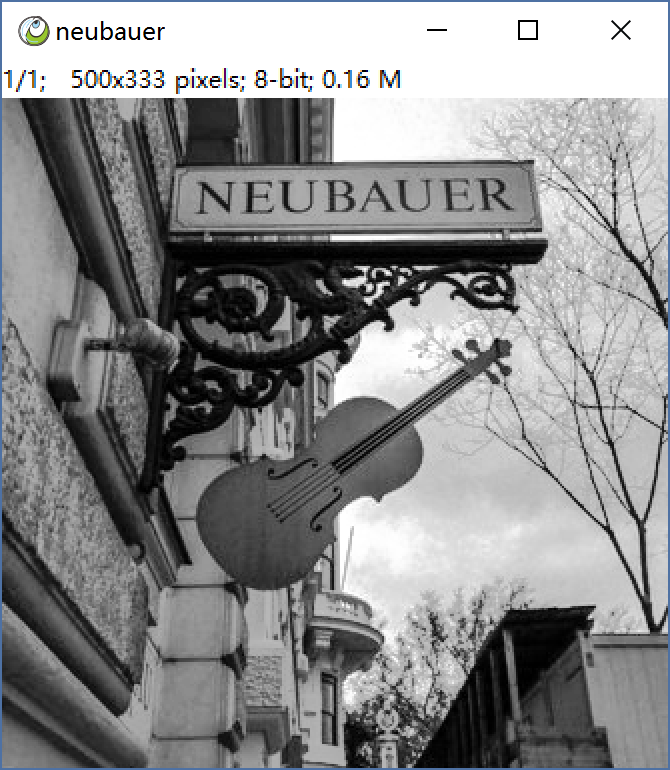
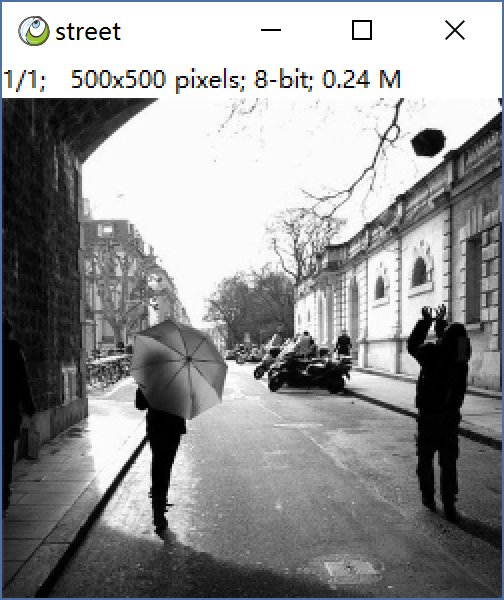
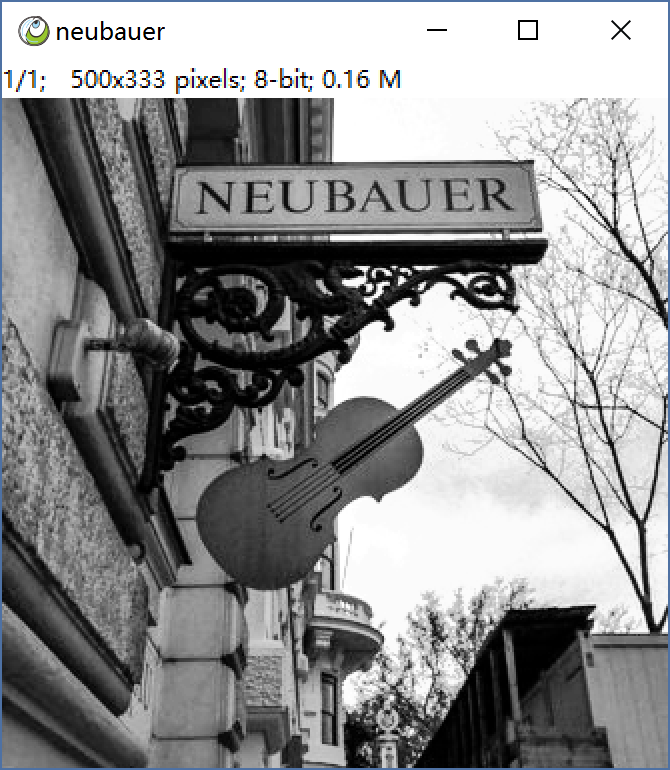
**直方图均衡化：**

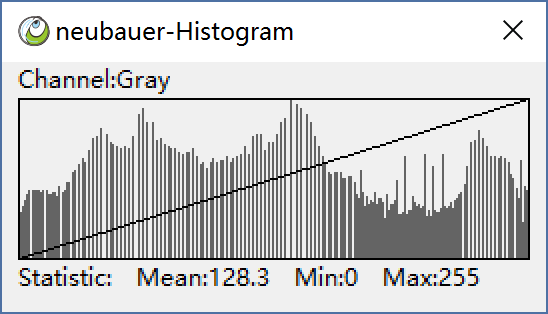
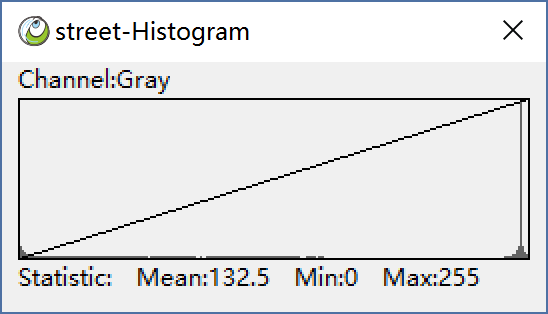
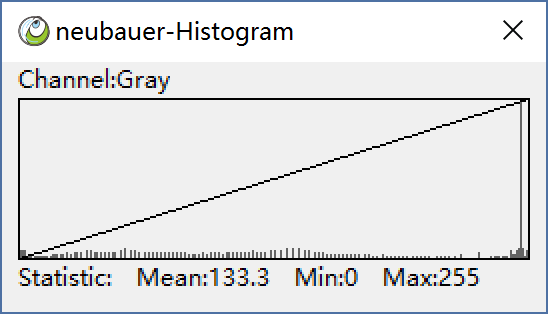
**$ Image> Adjust > Histogram Normalize**

**直方图匹配：**

**$ Image> Adjust > Histogram Match**

**演示 $ IBook > Chapter3 Pixel Operate > Histogram Normalize And Match**

均衡化 模板直方图 匹配后直方图

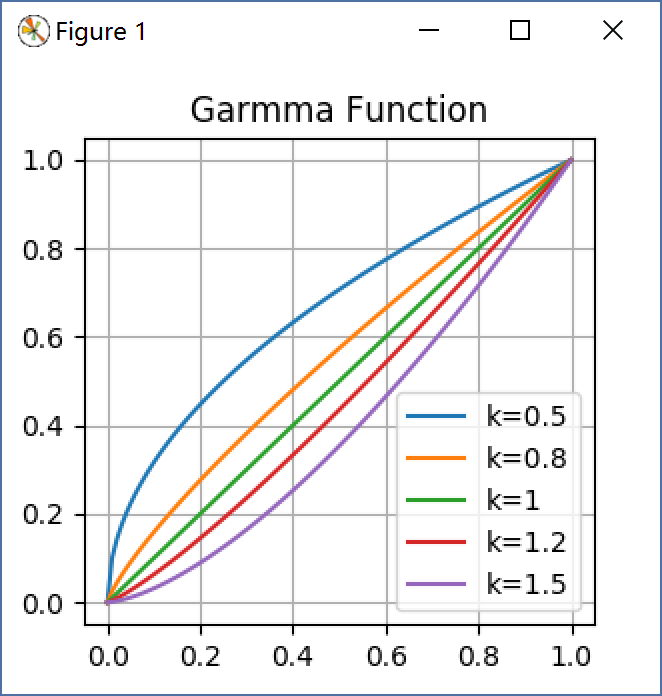
直方图均衡化可以使图像的像素在各个亮度区间分布均匀，因为我们只能移动与合并原直方图，而无法分裂，所以结果使一种概率意义下的均衡，高出的地方灰稀疏，低洼的地方会密集。而直方图匹配的结果是使调整后的图像的直方图尽可能接近目标直方图，这样视觉效果上会让两幅图给人的视觉体验相似。

**Garmma矫正**

**$ Process > Math > Garmma**

我们已经讨论过很多数学运算，这里把Garmma矫正单独拿出来是因为其在信号处理中有非常重要的作用。Garmma矫正是利用一个指数函数在0-1的区间内的图像对图像进行数学运算，由于其定义在0-1之间，因此需要先将图像的像素缩放到0-1之间，进行变换，再复原。对于8位灰度图像是如下关系：y = (x/255) ^γ\* 255

**演示：$ IBook > Chapter3 Pixel Operate > Garmma Curve**

Garmma 函数曲线 γ= 0.5 γ= 1.2

Garmma矫正是一个非常重要的数学运算，因为再成像的物理过程中，或者设备的显示过程中，很多时候由于电子元器件的特性，似的最终传感器的结果与原始信号成指数关系。而Garmma矫正通常被用于修正这种误差。

**多图运算**

我们之前介绍的处理都是单张图进行的，（直方图匹配虽然是两张图，但模板图的作用仅仅在于贡献出自己的直方图以做参考，所以不算真正意义上的多图运算）这里我们介绍的是多图运算，是两幅图对应位置的像素进行运算，可以使加，减运算，也可以是最大值，最小值以及差异的绝对值运算。

**$ Process > Image Calculator**

**演示： $ IBook > Chapter3 Pixel Operate > Image Merge**

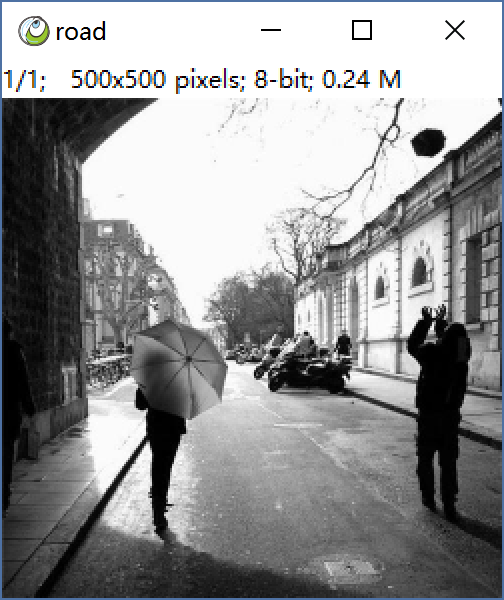
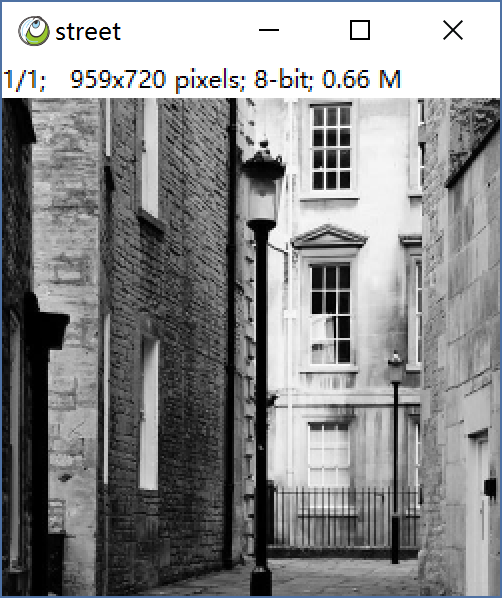
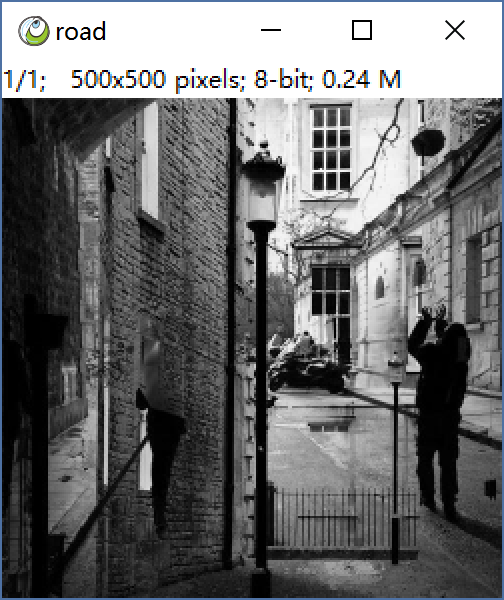
  

图1 图2 融合图像

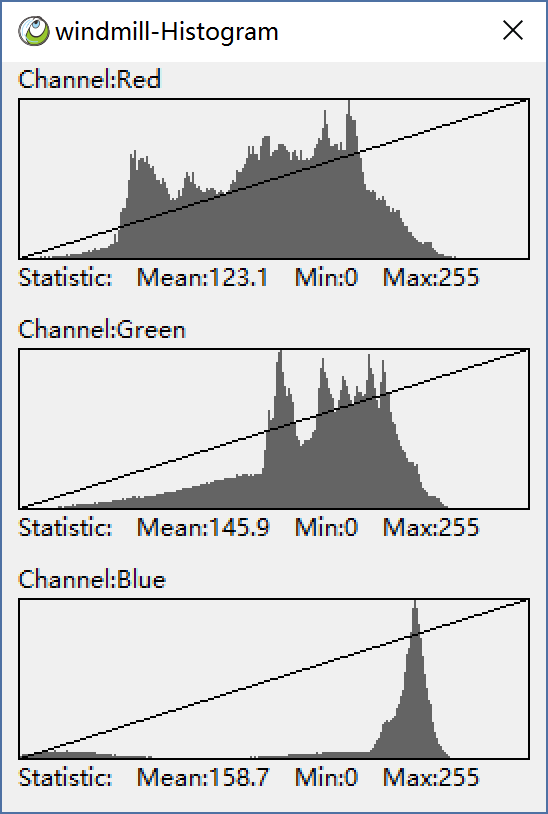
这里我们选用min运算，即两幅图对应位置像素做操作，保留较小的，因此我们看到的是两幅图中较暗的像素被保留下来。当然一般来说两幅图直接融合的结果会很不自然。这个功能更多是用于从原图进行一定的信息提取（比如轮廓）得到了新的图像，然后叠加回原图做效果展示，这个在后续的学习中回经常用到。

**彩色图像的数学运算**

我们讨论了很多的数学运算，不过截至目前，我们讨论的所有问题都仅限于单通道，那么对于彩色图像该如何处理呢？在本章最后，我们讨论彩色图像的直方图与数学运算。在此我们仅以RGB图像为例，讲解如何处理彩色图像。

**彩色图像的直方图 $ Analysis > Histogram**

严格意义上彩色图像的直方图应该是一个三维空间中的累计频率分布立方体，但我们更多时候是采用分通道处理的方法。

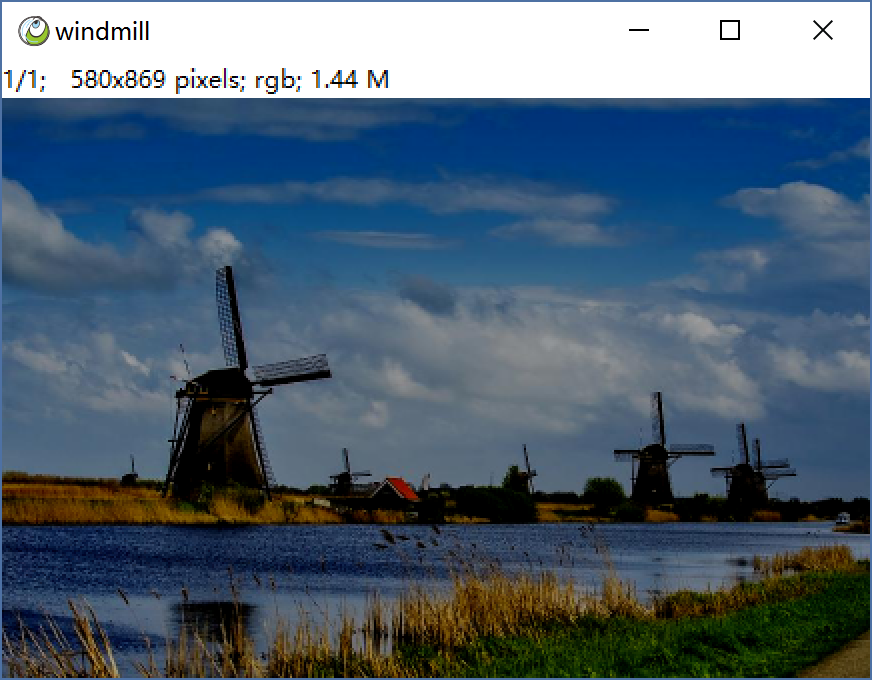
 

一副彩色照片 三个通道的直方图

**观察分析：**红色通道的直方图分布相对均匀，绿色通道则相对靠近亮部，而蓝色通道在亮部集中，因而可以大致分析出图像以蓝色调为主（天空，湖水）

**三通道统一处理 $ Image > Adjust > Bright And Constract**

如同单一通道，我们用相同的方式依次处理RGB三个通道。

降低亮度 增强对比度

之前针对灰度处理的亮度，对比度，色阶功能对于彩色图像都依然适用，他们将会作用在rgb三个通道上。

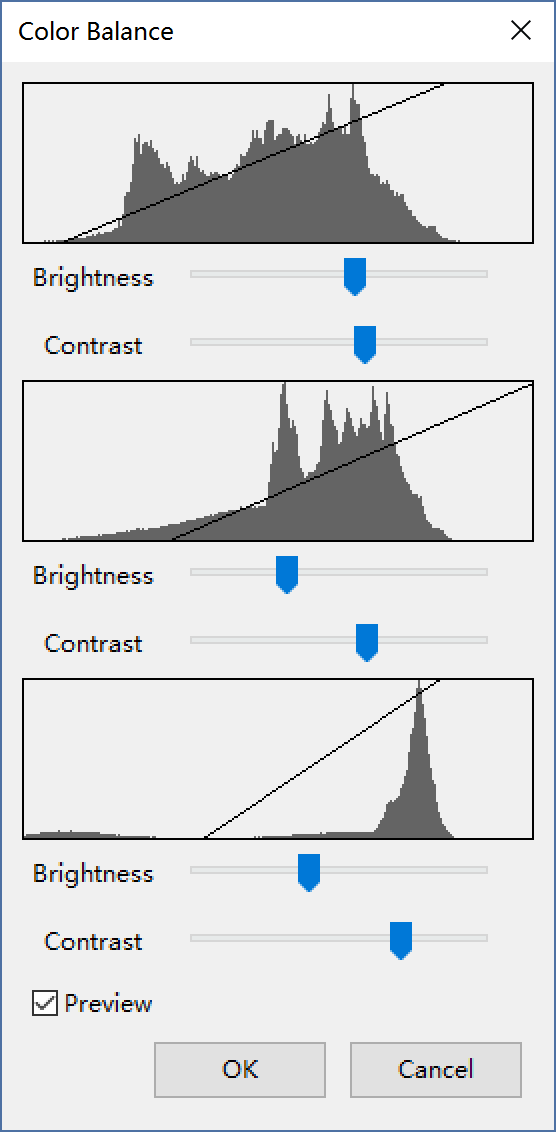
**三个通道分别处理**

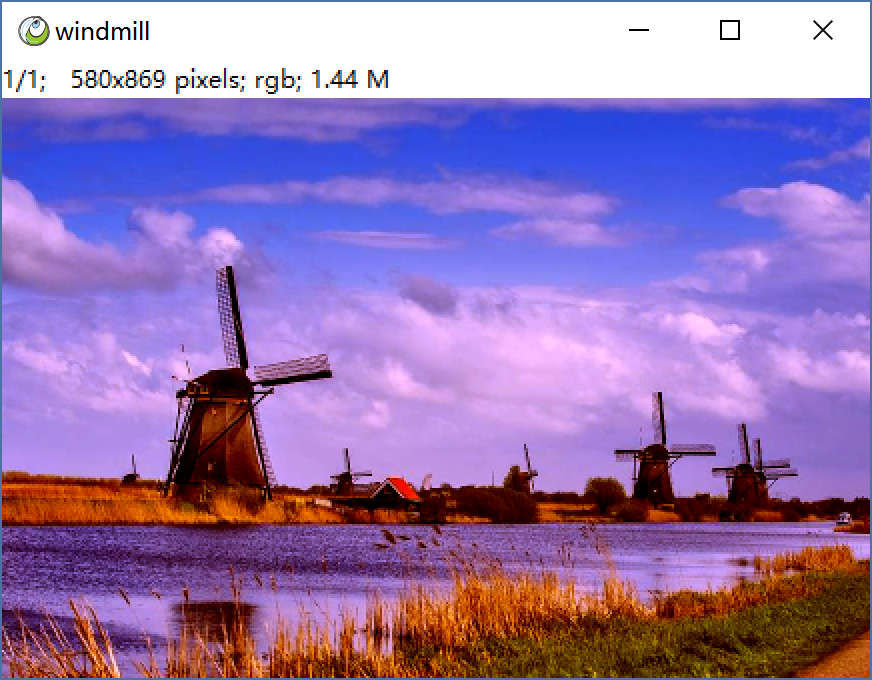
与单通道的亮度对比度，色阶相似，又有多通道的亮度对比度，色阶。不过由于分别调整各通道亮度，会导致分量比例变化，直接体现是图像的颜色发生变化，因此多通道调整亮度对比度在一些图像处理软件中叫做色彩平衡。

**$ Image > Adjust > Color Balance**

**$ Image > Adjust > Color Stairs**

**Color Balance演示： $ IBook > Chapter3 Pixel Operate> Color Balance Demo**





分通道调整亮度对比度 冷色调/暖色调 效果

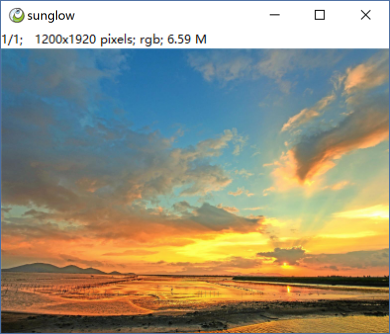
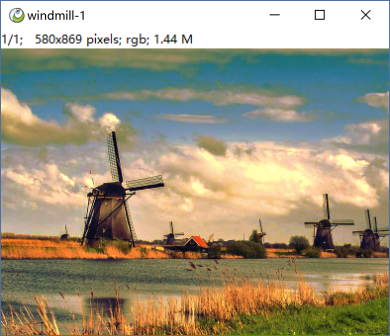
**调色技巧：**对于很多初学者，甚至是有美术基础的人，一个很大的困惑是调不出自己想要的效果，当然这需要一定的经验，不过大体上还是有技巧的，比如上图左边的参数，对应于右下方的暖色调调整。红色通道亮度增加，意味着整幅画面更加偏红，而绿色通道亮度降低，蓝色通道亮度有所下降，但由于对比度增强，因此天空更蓝，蓝色与红色混合后会使画面偏洋红色。我们知道红色给人以温暖，绿色给人以冷清，这样调整后，画面就更加温暖了。在后续我们学习了色彩模型之后，大家会有进一步认识。

**彩色图像的直方图均衡化与直方图匹配**

**$ Image> Adjust > Histogram Normalize**

**$ Image> Adjust > Histogram Match**

**演示 $ IBook > Chapter3 Pixel Operate > Histogram Normalize And Match**

均衡化 模板图像 调整后图像

**观察描述：**彩色图像均衡化之后往往伴随着色彩平衡的改变，因此图像中的颜色可能会发生较大变化。适用模板直方图进行匹配，得到的结果视觉体验上和模板图像类似，当然由于是彩色图像，可能由于通道变换不统一而发生色彩改变，并且在色彩越单调的图像上越容易发生。当我们对图像进行整体感知时，其实首先时感知整体色彩分布，而描述整体色彩分布的恰是直方图，模板图象在这里的作用仅仅在于贡献出自己的直方图，如果我们愿意，可以仅仅留下他的直方图，并且收集不同风格的直方图，然后选择性的取进行匹配。（许多诸如美图秀秀的修图软件里都有风格化，可以把图片变成鲜艳，淡雅，怀旧等风格，其实这些风格本质就是预定义的一套直方图）

**ImagePy中的像素运算**

ImagePy 有关数学运算的功能主要在两个地方，一个是 Image > Adjust 菜单下，也就是本章我们学习过的亮度，对比度，色阶的调整，以及彩色图像的色彩平衡，还有直方图的均衡化与直方图匹配。另一个是在 Process > Math 下面，这里定义了加法，乘法，最大值，最小值，开根号与Garmma矫正。另外有一个直方图视图，位于 Analysis > Histogram下，单纯弹出一个窗口显示图像的直方图。还有一个多图计算功能位于Process > Image Calculater用于做两幅图对应像素的运算。

**本章小结：**

本章我们讨论了数字图像的像素运算，加减乘除运算，线性运算，学会用直方图指导进行图像的像素调整，学习了亮度，对比度的本质，以及直方图均衡化，直方图匹配。进行了基础的多图运算，并学会对彩色图像进行像素运算，得到了一些神奇的结果。