**数字图像处理基础及实验**

**前言：**随着计算机软硬件的发展，数字图像处理在遥感，医学，生命科学等科研工作中起到越来越显著的作用。而现实情况是很多非计算机专业的学生，由于理论和编程能力较差，往往不能很好利用图像处理技能解决实际科研问题，本书将试图以友好，感性，可操作的一种方式，向理工科学生和研究者介绍数字图像处理知识以及分享一些实用技巧。

**本书的写作目的**

本书希望以一种友好，感性的方式向读者介绍数字图像处理知识，结合具有启发性的案例，培养读者认识问题，解决问题的能力。因此本书的受众主要是那些需要用到数字图像处理的非计算机专业理工科学生和研究人员作为参考书，同样也适合那些打算深入，系统学习数字图像处理相关知识的朋友作为入门教材。

**阅读本书的预备**

本书对读者的要求是初级的，不需要有太高深的数学基础和计算机编程技能。只要有清晰的逻辑思维，最好对自己所研究的领域有一定的认识，并且大致了解哪些问题可以用数字图像处理来解决。另外本书的案例涉及各个领域，但通过一些简单的背景知识介绍，总能转化成一个图像处理工作，因而，不要被学科差异所辖制，希望读者能够认真的阅读这些案例，不要因为那个领域与自己无关，就匆匆跳过。

**本书的特点**

数字图像处理作为一门理论性很强的学科，往往被划分在应用数学，信号处理，计算机科学等专业中。但其实在遥感影像，医学影像，细胞，生物图像，化学工业等领域有非常重要的应用。而对非数学和计算机专业的研究者，往往数学理论基础，计算机编程能力相对较弱，因而阅读数字图像处理的专业书籍非常困难。而一些行业书籍，往往基于特定的软件，操作性较强，却没有很好建立数字图像处理的知识体系，也忽略了思维训练。

本书将大致按照数字图像处理的知识体系展开，尽可能感性的讲解知识点，避免用代码包括伪代码来描述。进而结合一些应用领域的案例，培养读者思考问题，解决问题的能力。在书的末尾，会简单介绍一些基础的编程知识，为有兴趣在这一领域深入研究的读者抛砖引玉。

**本书的价值排序**

另外，本书不求知识体系的绝对完整性和严密性，甚至有意回避一些对于基础薄弱的人理解困难的知识（比如频谱技术，多尺度理论等）。但注重思维训练，结合具有代表性，启发性的案例。所以在价值排序上本书尊崇如下原则：

1. 不破坏读者积极性，以友好的方式对知识展开叙述。
2. 案例具有启发性，并可以从感性的角度去理解，思考，学会合理运用知识。
3. 注重可操作性，本书的大多数讲解和案例可以用ImagePy软件进行操作实践。
4. 兼顾知识体系的完整性（体系在于自我总结和养成而非灌输）

**在线资源**

书中的多数讲解，案例是用ImagePy开源图像处理工具进行操作的，读者可以在官方网站：[www.imagepy.org](http://www.imagepy.org) 获得书中案例的相关支持。

**致谢**

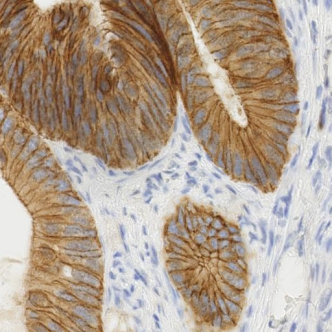
首先感谢家人对我工作的支持与理解，作为公益性的开源项目，以及相关书籍的编写，家人的宽容是非常重要的。其次感谢异步社区给予我这样一个机会，来向广大读者展示，分享自己的思想与技能。另外作为一个开源框架，ImagePy海纳百川，吸纳了scikit-image, opencv等著名的开源项目，也从这些开源组织中得到了一定的支持，得以让项目快速成长。也感谢众多的用户与开发者积极的回馈，激励着我继续前行。最后，感谢救主耶稣，圣灵的充满让我能够排除外界的一切诱惑，在非常艰难的条件下完成项目的开发与图书的编写，我所做的一切，尽为彰显神的伟大。

**目录**

1. 数字图像的基础概念
2. 数字图像的定义
3. 数字图像的特性
4. 数字图像的要素
5. 数字图像的存储
6. 数字图像的分类
7. 数字图像处理的研究领域
8. ImagePy简介
9. ImagePy主界面
10. ImagePy功能简介
11. 打开，保存图像
12. 滤波器运算
13. 交互工具
14. 其他特性
15. 如何利用ImagePy学习本书
16. 关于ImagePy
17. 像素运算与直方图
18. 基础运算
19. 充分利用像素值的区间
20. 二维图像的直方图
21. 亮度对比度色阶
22. 直方图均衡化与直方图匹配
23. Garmma矫正
24. 多图运算
25. 彩色图像的像素运算
26. ImagePy中的像素运算
27. **数字图像处理的基础概念**

**数字图像**

数字图像已经遍布我们的生活当中，数码照片，遥感影像，CT，电镜，这些都属于数字图像，那么究竟什么才是数字图像呢？这个意义并不重要，因为对于初学者来说，精确，抽象的定义并不易于理解，而对于资深专家，也没有必要讨论定义。这里也不打算下定义，这样责任重大，只对数字图像做一些尽可能准确的描述。

truncated  

数码照片 显微镜照片 射电望远镜成像

数字主要是区别于模拟的，模拟是指通过一系列的物理化学过程，将图像信息进行固化得到，例如胶卷相机，是通过胶片上的感光元素曝光，实现对光信息的记录。而数字图像**本质是一段保存了图像信息的数据**。

**图像产生的流程**

无论数字还是模拟，图像都是按照 **采集>(预处理)>存储>(后处理)>展示** 的顺序被我们感知到的。而对于数字和模拟，每个环节有不同的流程，每个环节都是一个很有意义的课题，而本书重点讨论数字图像的后处理过程。

**数字与模拟的关系：**真正意义上的数字图像只存在于计算机当中，但必须通过采集设备得到，并借助显示设备展示。而采集设备一定是模拟的，因为本质上自然界的光是模拟的，通过CCD，CMOS把光转化成电压并记录到存储设备。而显示设备一定也是模拟的，因为人的眼睛感知光线是一个模拟过程，成像到视网膜并以电信号传递给大脑。显示器不过是一个把数字量转化成光的设备，和打印机别无二致。

**数字图像特性**

**网格性：**本书讨论的数字图像是那些排列在规则网格上的图像元素（像素）构成的数据。

**空间离散性：**数字图像由有限数量的像素构成，因此具有空间离散性。

**色彩离散性：**数字图像每个像素能表达的值是有限的，因此具有色彩离散性。

**数字图像的要素**

**维度（对应于网格性）**数字图像一般来说是二维的，但实际上一维，三维，甚至高维的图像是存在的。比如CT断面扫描。

**尺寸（对应于空间离散性）**数字图像每个维度上像素的个数是有限的，所以图像总是有具体的尺寸的。

**深度（对应于色彩离散性）**数字图像上每个像素都是由若干个二进制位构成，以存储信息，位数决定了所能表达颜色的种类多少。

**一些衍生特性**

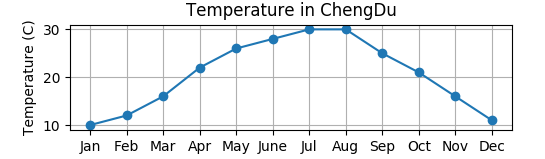
**通道** 是指几张从不同角度描述完全相同的空间内容的图像，比如彩色图像，往往用RGB三通道表示，每个通道描述了某个色彩分量信息。

**分辨率（空间）**我们已经知道图像具有特定的尺寸，而图像描述的现实物体也有真实尺寸，这样就有类似比例尺的概念。我们称之为分辨率，分辨率决定了目标刻画的精度上限。

**成都市某年的各月平均气温 $ IBook > Chapter1 Digital-Image > Chengdu Temperature**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **1月** | **2月** | **3月** | **4月** | **5月** | **6月** | **7月** | **8月** | **9月** | **10月** | **11月** | **12月** |
| **10℃** | **12℃** | **16℃** | **22℃** | **26℃** | **28℃** | **30℃** | **30℃** | **25℃** | **21℃** | **16℃** | **11℃** |

或许你很难把这个想象成图像，但它的确是一段排列在规则网格上的数据，我们试图对其可视化。



用图表来表示成都市一年中的月平均气温

**temperature**

用一行像素表示月平均气温，其亮度值代表温度

**temperaturec**

用一个红黄系的假彩色增强视觉效果

**扩展阅读：**如果我们愿意，我们可以认为一年下来，每个月的平均温度是一幅图像，并且可以以多种方式进行可视化，如果再加上降雨量，甚至是双通道图像，另外这种一维图像还有一个很好的特性，可以与图表对应，非常易于理解，因此本书会由多处采用一维图像进行讲解。通道的概念和维度与深度似乎有一定的冲突，但没必要纠结。通常的灰度照片，我们习惯用8个位来存储，2^8=256，也就是每个像素最多能表示256中颜色，而彩色图像用3个通道的8位来存储，3\*8=24，因此叫24位图像，因为这个分辨率是肉眼分辨的极限，也是多数显示设备的标准，但必要时，我们可以存储更多通道以及精度更高的图像。

**数字图像的存储**

**Raw** 我们已经知道图像是一段规则排列的记录图像信息的数据，因而原始的存储也就是按照规律，把像素的值一个一个的存下来，这就构成了图像的原始信息。

**Bitmap** 由于具备网格规则性，因此也叫作位图，位图的存储接近于原始信息，只是外加了一些尺寸等信息。

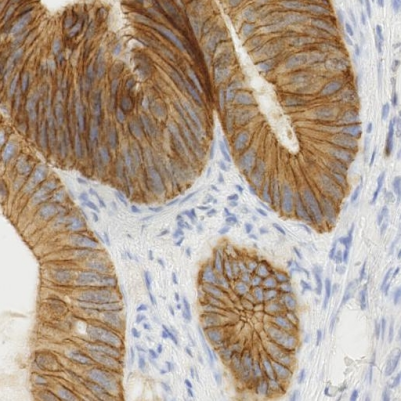
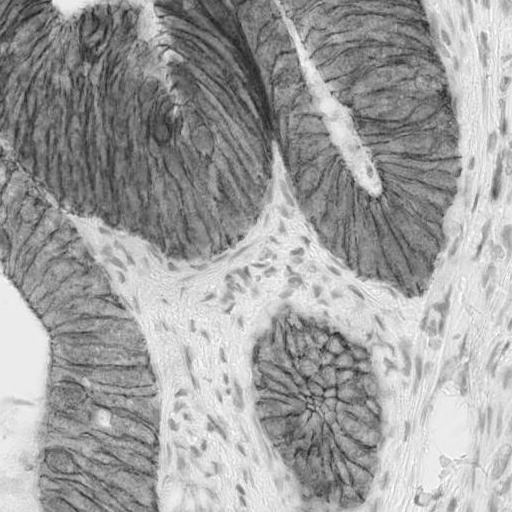
**TIFF** 有些时候我们需要在存储像素信息的同时，也存储一些其它信息，比如遥感影像的坐标参考系，而TIFF就是一种通用的标签图像存储格式。

**JPG** 有经验的平面设计师都会知道BMP图像存储城JPG所占磁盘空间就会大大减少，而这里面的原因是，JPG本质是是一种压缩数据格式，但注意这种压缩不是没有代价的，会对图像的高频信息造成一定的损失。

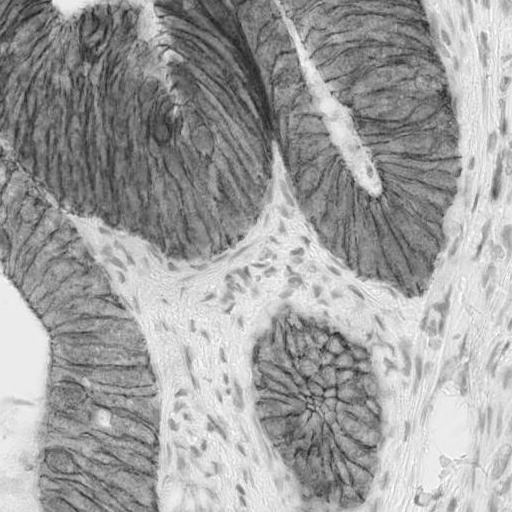
**PNG** 网络便携格式，同样是对BMP的一种压缩格式，但是用到的压缩原理与JPG不同，并且这种压缩是无损的。

**特殊格式** 以上介绍了一些常见的图像存储格式，这些格式可以满足我们日常拍照，上网等一些需求，但对于一些行业，往往有特定的需求，因而又有很多行业内的特殊格式，比如医学上使用的DICOM数据，电镜中使用的DMC，MRC等格式，这些格式主要是为了满足一些硬件设施，或者存储一些行业内有用的相关数据，但就其图像属性来说，与其他的格式区别不大。

需要指出的是，各种压缩格式往往是为了存储，但是当其读取到内存，进行分析的时候，往往需要重新转成规则网格的位图格式，以便各种图形处理操作。

1024\*1024 rgb bmp 3M 1024\*1024 8-bit bmp 1M 1024\*1024 8-bit jpg 327K

1024\*1024 rgb png 736K 1024\*1024 8-bit bmp 392K 1024\*1024 8-bit png 106K

**BMP图像的大小：**由于BMP图像主要是存储的图像原始信息，因而其大小是可以精确计算的，具体计算方式如下： Size = width \* height \* deepth (bit)，比如一幅1024\*1024的8位图像，其大小就是:

1024\*1024\*8(bit)= 1024\*1024（byte）= 1024 (K) = 1M，而相同大小的彩色图像就是3M。

**压缩格式简介：**相比之下JPG，PNG图像的大小就无法精确估计大小，因为他们采用的压缩算法不仅取决于像素的个数，而且与其内容有关，比如一幅风景画和一张白纸所包含的信息量是不同的。JPG是采用频谱技术进行压缩的，所以当图像以高频为主，比如文稿扫描件，那最好不要用JPG格式存储，简单说，一幅色彩多变，但相对柔和的风景画，JPG往往可以得到比较好的压缩效果，而对颜色相对单一的矢量图，PNG是一个好的选择。

**数字图像的分类**

这是一个很泛的问题，从不同的角度，可以有多种分类方法

**按获取方式分类** 数码相机照片，电镜，CT片，卫星影像，雷达成像，射电望远镜成像等。

**按通道和深度分** 二值图像，灰度图像，RGB三通道彩色，单精度浮点图像，多通道图像等。

**按应用领域分** 网络图像，生活照片，遥感图像，医学图像，生物影像，天文影像等。

**数字图像的研究领域**

我们已经讨论了图像产生的流程，这里简单介绍一下每个部分的研究内容

**采集和前处理**

这一部分是成像领域所研究的，主要是研究光学，传感器，信号放大，增益等。这一部分需要国家基础工业做支撑，比如我们日常使用的单反相机，或天文望远镜的制造。

**存储**

存储的研究课题是图像压缩，以及如何更稳定，更快速地存储大量的信息，我们知道图像信息往往是非常巨大的。（但有些时候这个过程是在内存中完成的，就是我们并没有永久性地存储，而是获取之后，及时分析，就舍弃，然后处理下一帧）

**后处理和展示**

这一块主要研究对成像后的图像进行处理并展示信息，这种处理分为三种形式：

**图像增强：**输出图片，使得里面表达的内容更加清晰，比如重新曝光照片，或增强对比度。

**信息提取：**输出从图像中获取到的信息，比如显微镜中细胞的个数，航片中中河流的长度。

**模式识别：**输出图像中更高级的信息，比如解读这是什么字，图中的人是谁，甚至回答图中有什么，这意味着什么。

其实这样划分也并不是非常严谨，很多时候问题是交叉的，一般来说增强，提取，识别在复杂度上有递进关系。近年来比较热门的机器学习很多都是基于图像的模式识别领域的，而本书重点将会讨论图像增强和信息提取部分。

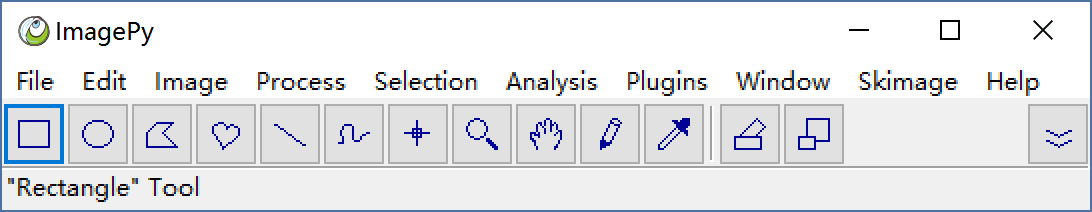
1. **ImagePy 简介**

本书包含大量的案例，虽然本书中的案例尽可能以生动，形象，感性的方式进行讲解，但必要的实践，操作是必不可少的，书中的所有案例都是基于ImagePy的。ImagePy是作者本人研发的开源，免费图像处理软件，读者可以在http://imagepy.org免费下载。

**介绍**

想必大家听说过Image-Pro Plus或ImageJ，如果你没有听说过，那一定也听说过大名鼎鼎的Photoshop，他们都是图像处理软件，所不同的是，Photoshop侧重于图像的视觉效果，而Image-Pro Plus重点是图像的分析。ImagePy是一款类似的软件，采用Python开发，很大程度上借鉴了ImageJ，下面我们来简单认识一下ImagePy，在此仅仅做基本了解，目的是让读者能够很好地使用ImagePy去操作本书中的案例，另外具体的操作说明会在本书附录中展现。

**主界面**



主要由标题栏，菜单，工具栏，状态栏构成。

**功能简介：**

ImagePy可以打开常见的图像类型，对图像进行直方图统计，数学运算，可以进行常见的滤波器运算，几何运算。

**File：**各种文件打开，保存

**Edit：**剪切，拷贝，粘贴，撤销等常用操作

**Image：**图像色阶调整，克隆，几何变换，调整画布尺寸等。

**Process：**各种像素操作，数学运算，滤波器运算等

**Selection：**选区的几何运算

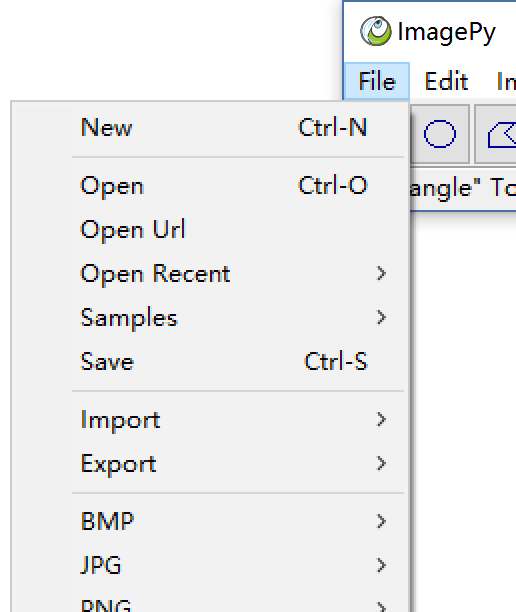
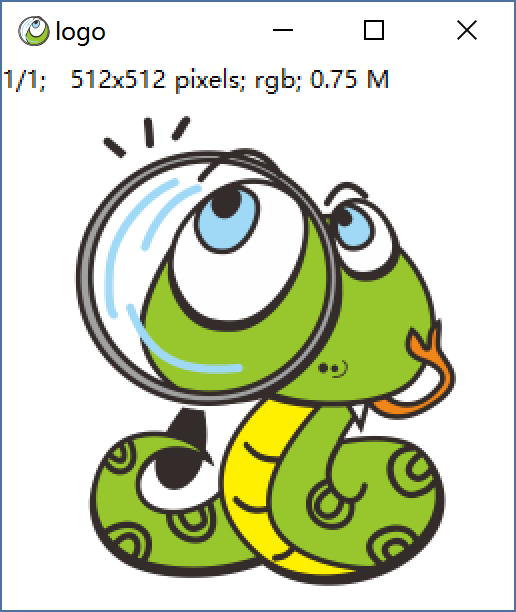
**Analysis：**像素统计，区域分析等

**Plugin：**宏录制，插件管理，扩展功能管理等

**Window：**窗口管理

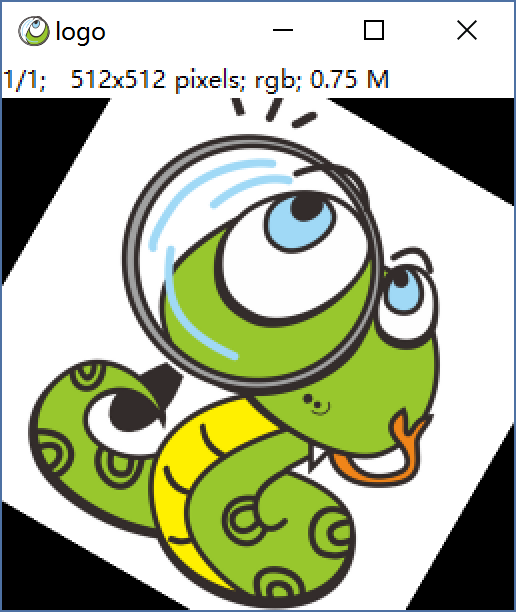
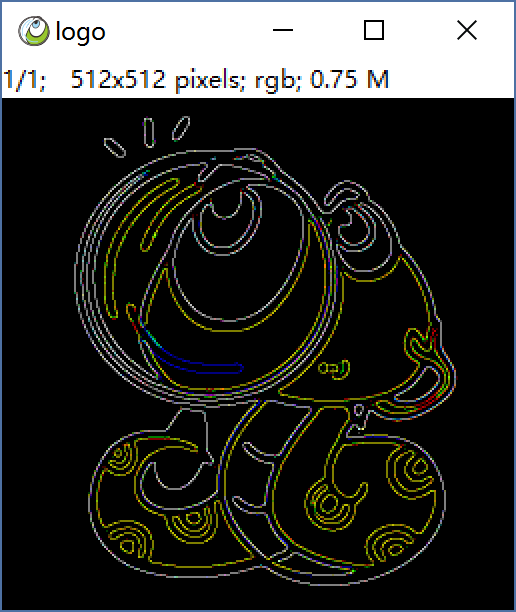
**Help：**主题帮主

**打开，保存图像**

ImagePy可以打开，保存BMP，JPG，PNG等常用的图像格式，通过扩展插件也可以打开DICOM等专用格式。

**滤波器操作**

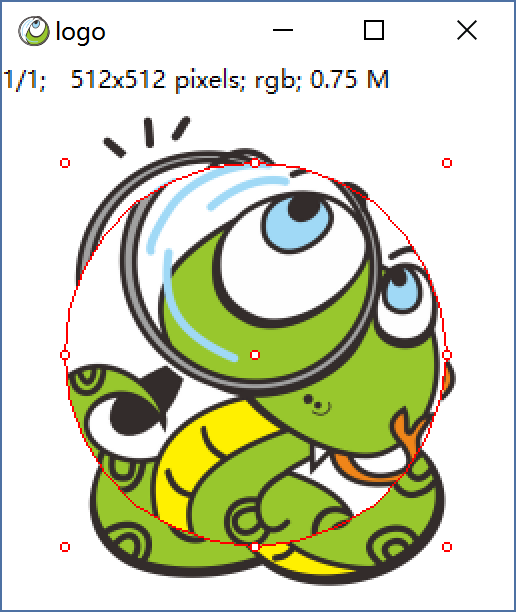
gaussian rotate canny

ImagePy包含大量的常规滤波器操作，多数滤波器操作都会以一个参数对话框形式出现，如果勾上Preview则可以实时预览效果。并且操作后，有单步撤销功能。但注意，有些滤波器不需要参数，也不是所有功能都支持预览，不是每个功能都支持撤销。

**交互工具**



[ 选区类工具 ][ 浏览 ][ 绘图 ][ 变换 ] [ 更多 ]

rotate with roi draw with pen distance measure

这里是ImagePy的工具栏，工具多是借助鼠完成一些操作。比如框选，绘图，做图像变换，测量等。工具栏分两部分组成，左边是常规工具，是最常用的，而右侧是专用工具，可以从最右侧下拉菜单中选择专用工具集，比如绘图工具集，测量工具集等。一些工具双击会弹出属性配置对话框，比如画笔可以配置线宽度等。

**其他特性**

**支持批量处理：**允许用户把一系列尺寸，格式相同的图片批量导入到ImagePy中，然后任何操作会自动批量处理每一张图片。

**支持宏录制：**宏本质上是一个记录了一系列操作的文本文件，这使得一些常规，固定的操作可以用宏来大大简化。ImagePy允许用户自己录制宏。

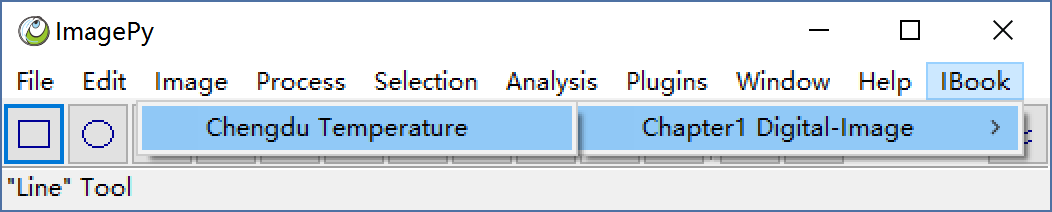
**可扩展插件式框架：**ImagePy不仅是一款图像处理工具，更是一个轻量级，可扩展的图像处理框架，由于这涉及到编程，因而在基础理论部分和实践操作部分我们都不会涉及，而在最后的编程基础部分，做一定的介绍，有兴趣深入学习的朋友可以仔细阅读。

**如何结合ImagePy学习本书**

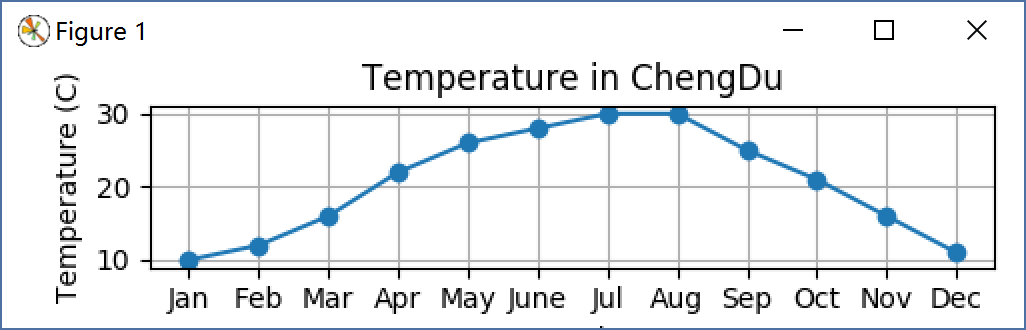
以上我们简单介绍了ImagePy，本书的案例都将以ImagePy进行操作，其中涉及到操作部分，都会尽量写明在哪个菜单下的哪个功能，此外，为了配合本书的学习，作者写了一个针对性的扩展包，也就是最后的IBook菜单，里面配备了本书案例中使用到的数据，并用宏的方式做了一系列的功能演示插件。读者请结合实践来阅读本书，加深理解。

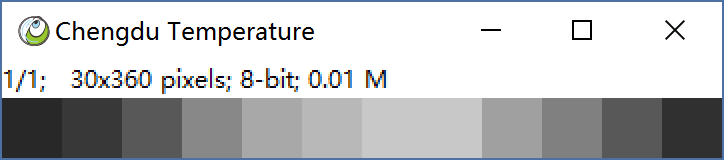
比如前一章节讨论的，一年内成都市的月平均气温，我们可以用ImagePy查看演示：

**$ IBook > Chapter1 Digital Image > Chengdu Temerature** 你将看到如下：



**本书用 $ 后面的部分是表示操作，用>表示菜单层级关系。**





**关于ImagePy**

ImagePy是作者本人基于Python开发的开源图像处理软件。从软件的最初设计，开发，调试，完善耗费了大量的精力，至此基本是一款拿得出手的产品，但依旧有很多功能有待完善，因此软件会定期更新，读者都在官网http://imagepy.org随时下载到最新版本。如果暂时还不是那么完美，也请多理解，如果在使用过程中如果遇到问题，请主动联系我，我会及时修正，尽我所能确保软件能茁长成长，关于使用技巧方面的交流，可以加QQ群，596310256，或搜索ImagePy进行学习，沟通。

1. **像素运算与直方图**

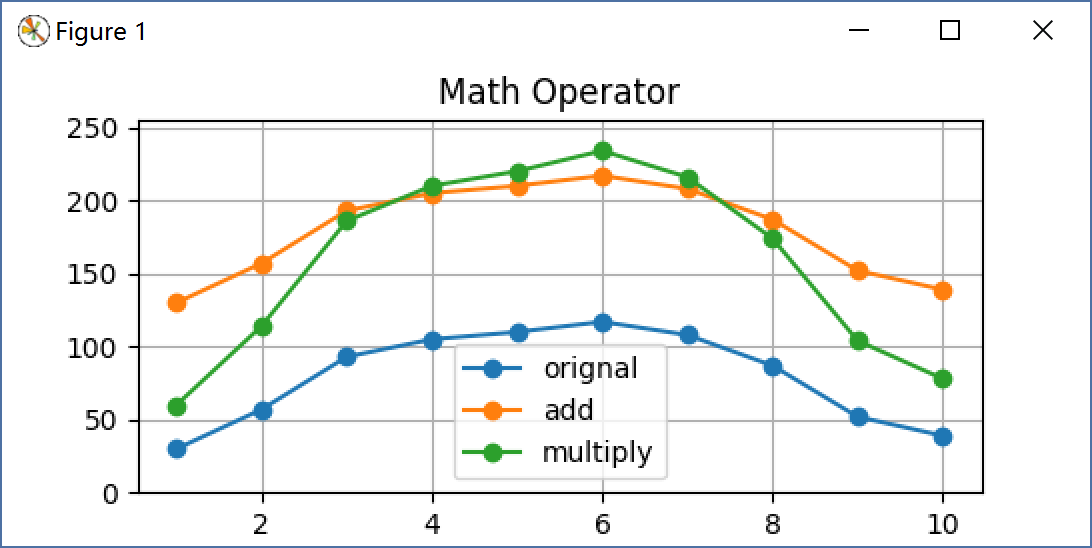
我们已经了解数字图像是由排列在规则网格上的图像元素构成，而这些元素是可以进行运算的，本章我们来讨论图像的数学运算与直方图。如果你不了解数字图像处理，但一定或多或少了解Photoshop，一定听说过色阶，亮度，对比度这些概念，通过本章的学习，我们将对这些概念由一个很好的认识。

**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)

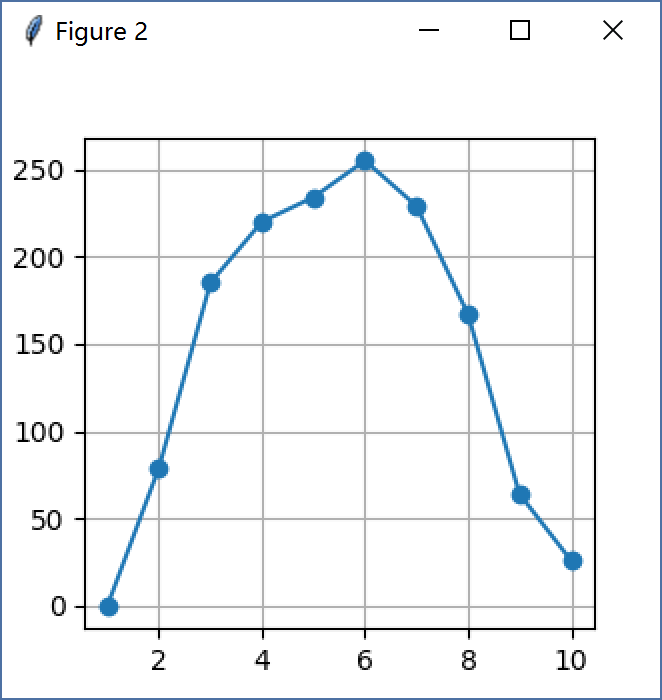
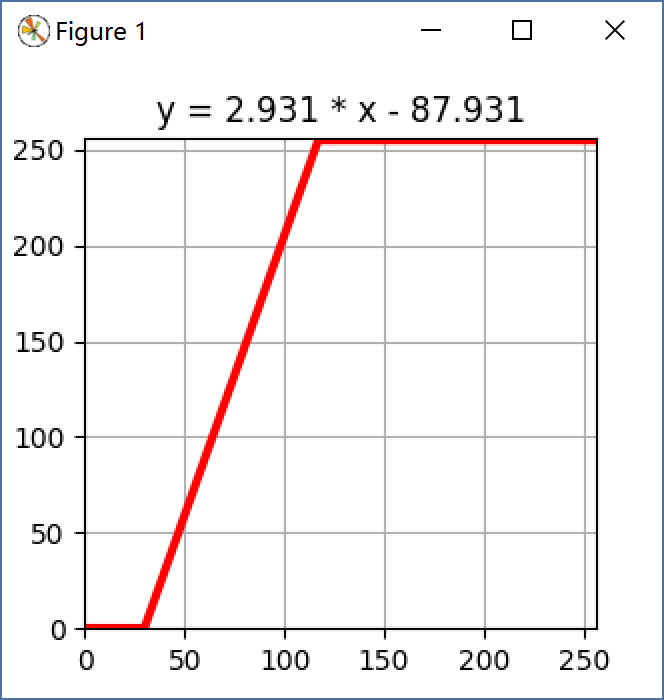
求解得：y = 2.931 x - 87.931

**$ 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。

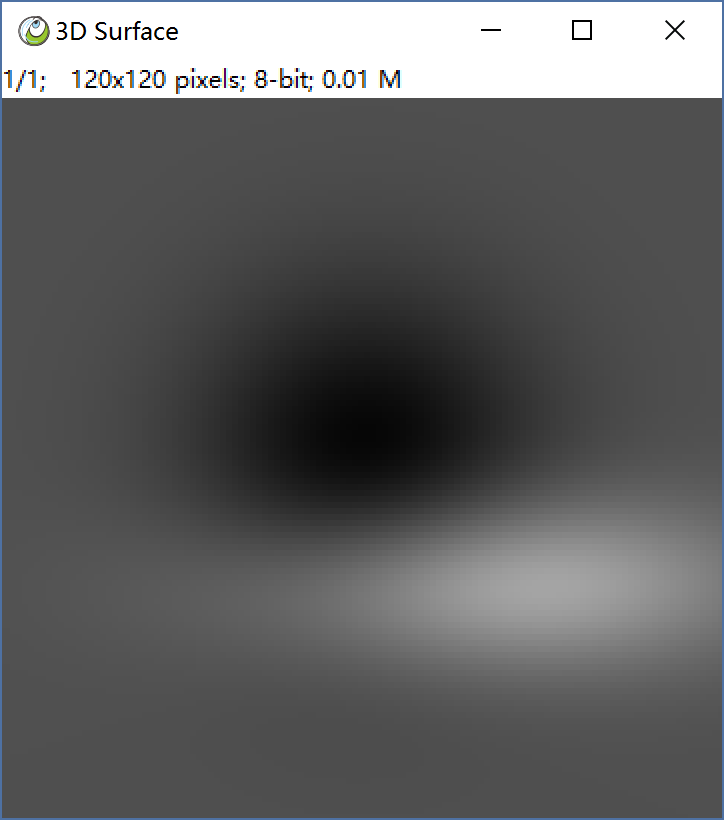
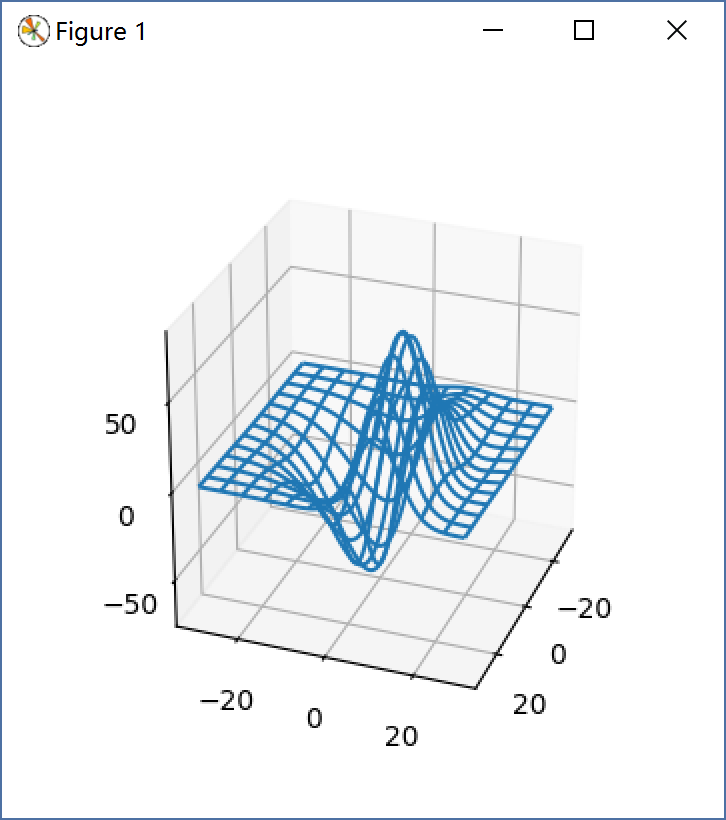
变换后图表 映射函数

**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**

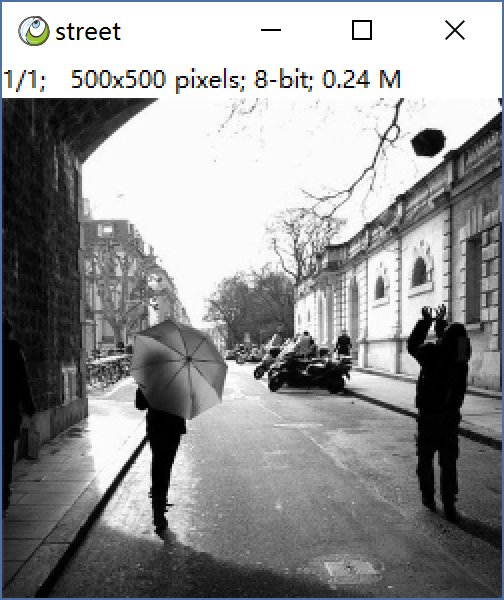
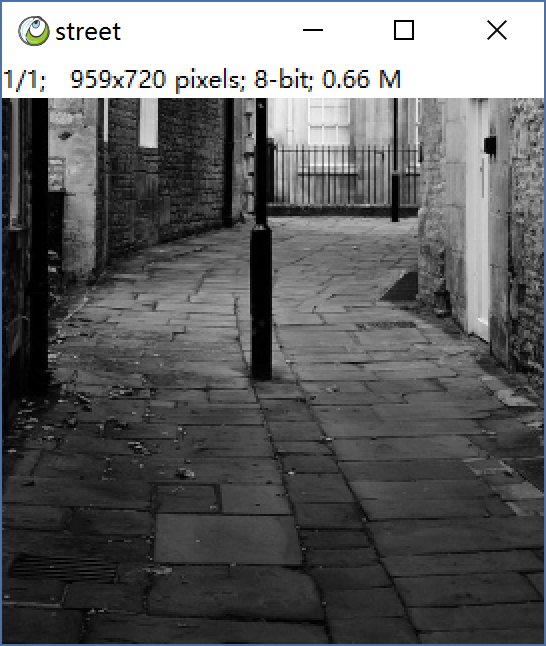
 

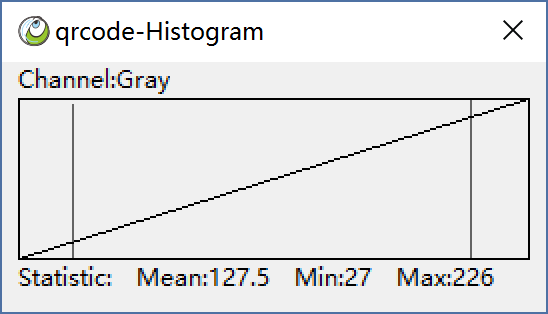
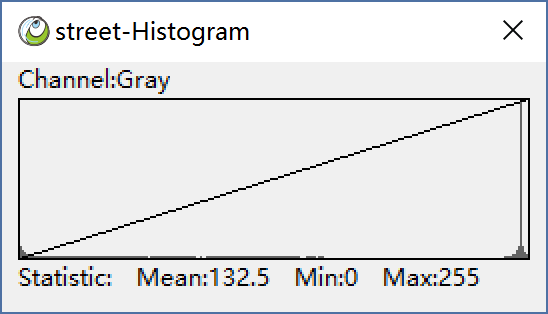
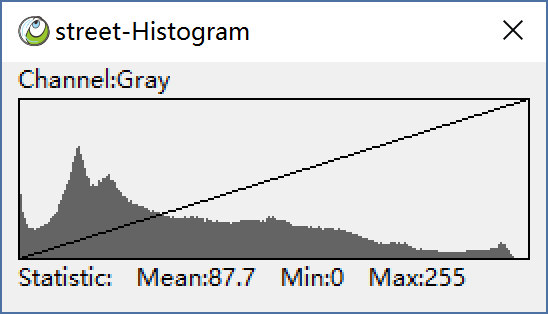
一副数字高程图像 对其进行表面绘制

**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。

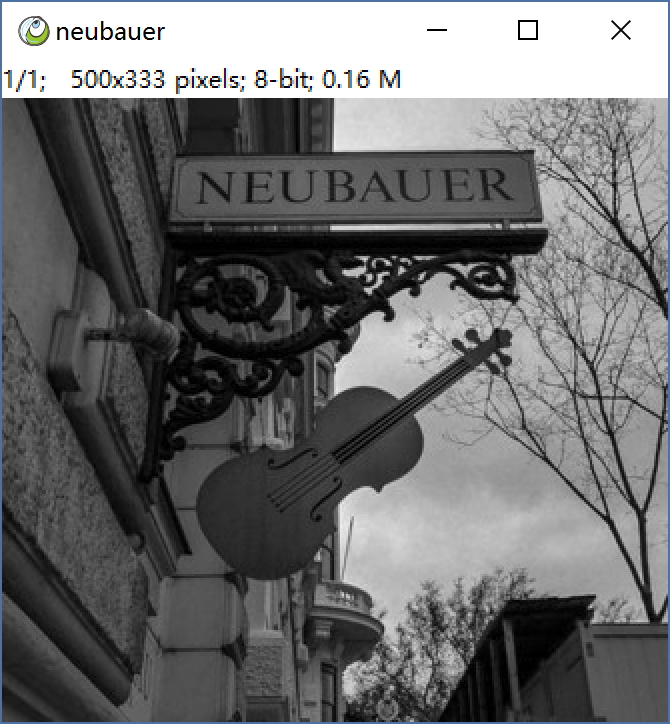
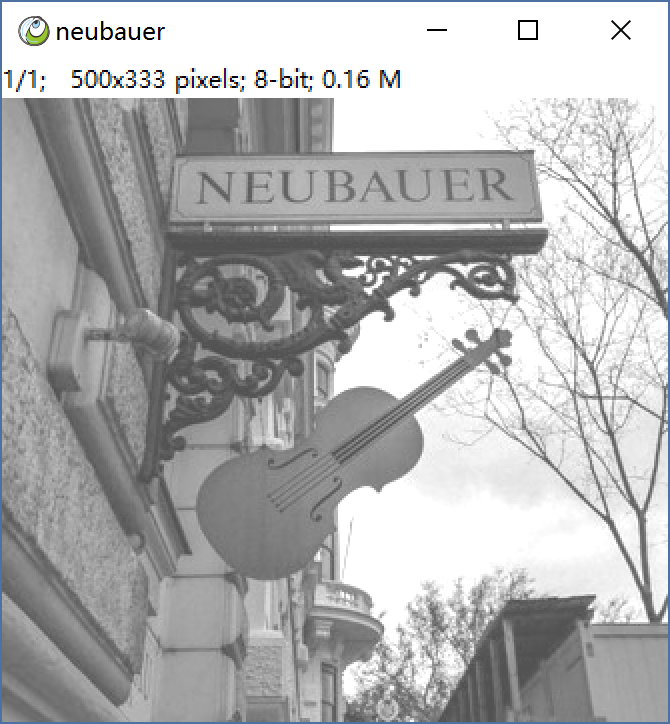
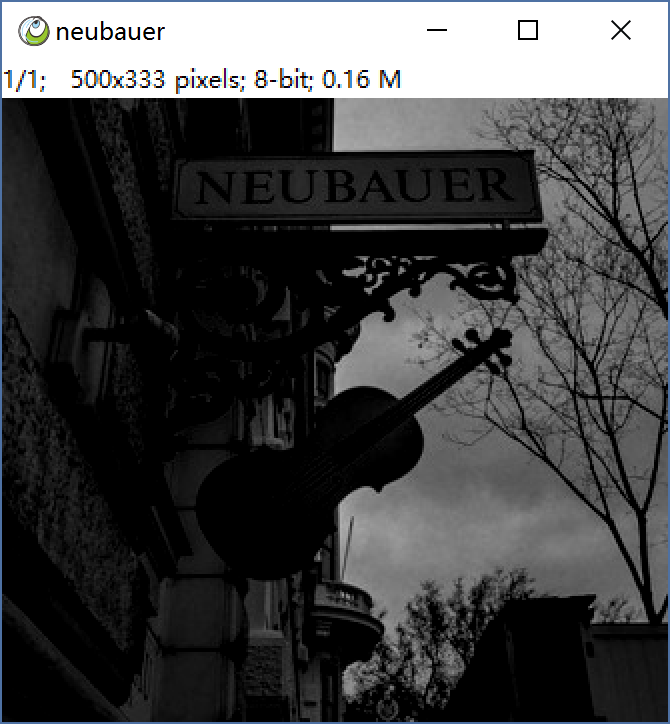
**亮度对比度色阶**

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。

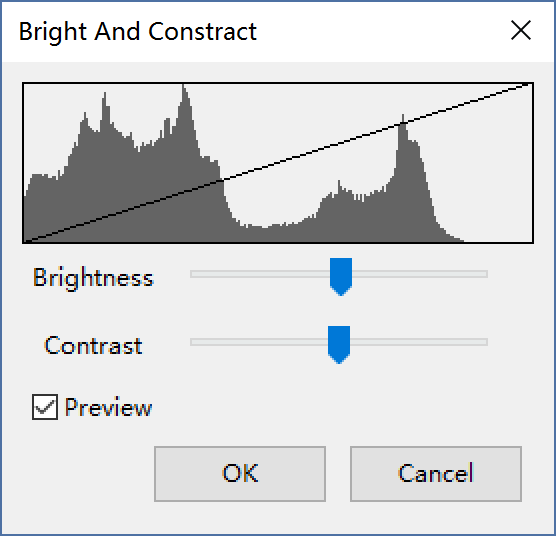
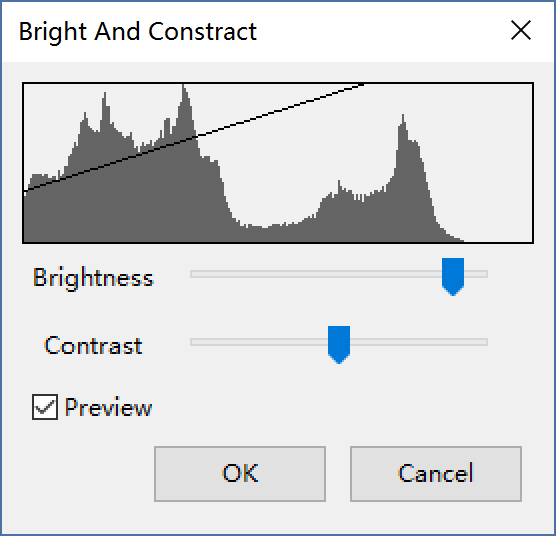
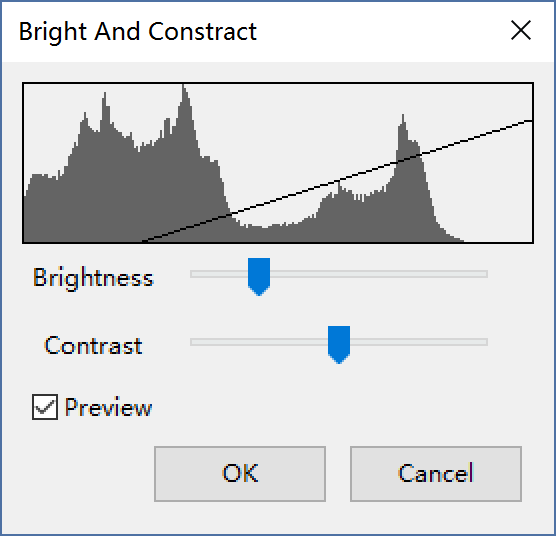
**亮度调整**

**$ IBook > Image Referenced > neubauer**

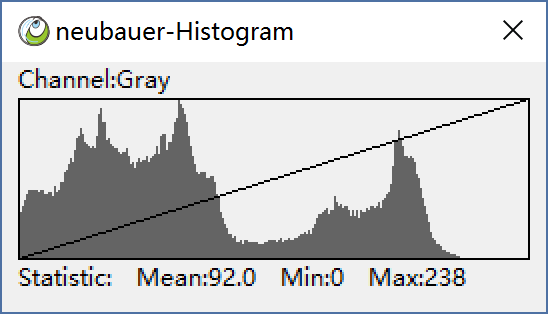
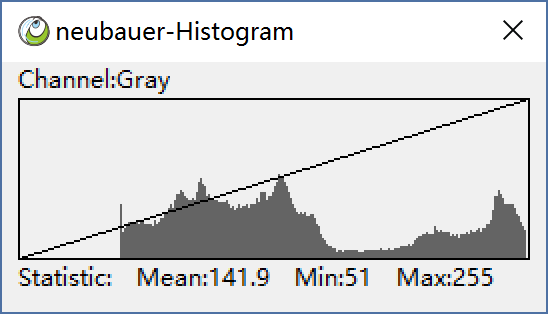
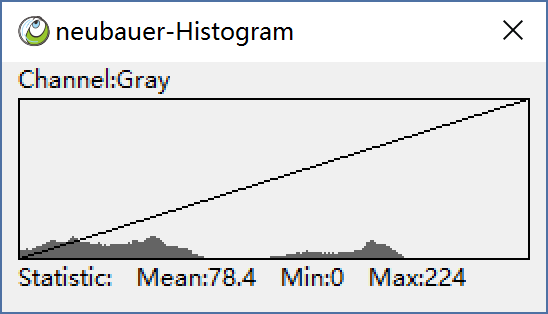
**$ Image > Adjust > Bright And Constract**

参数对话框

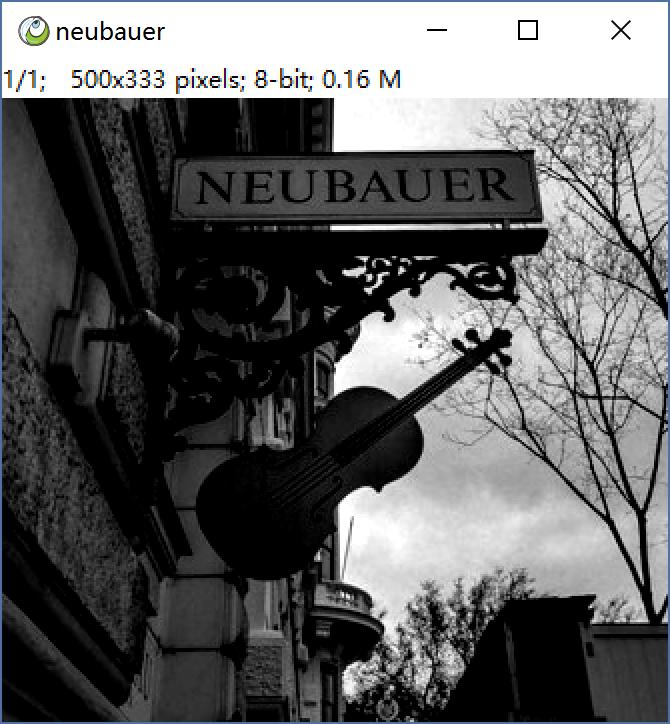
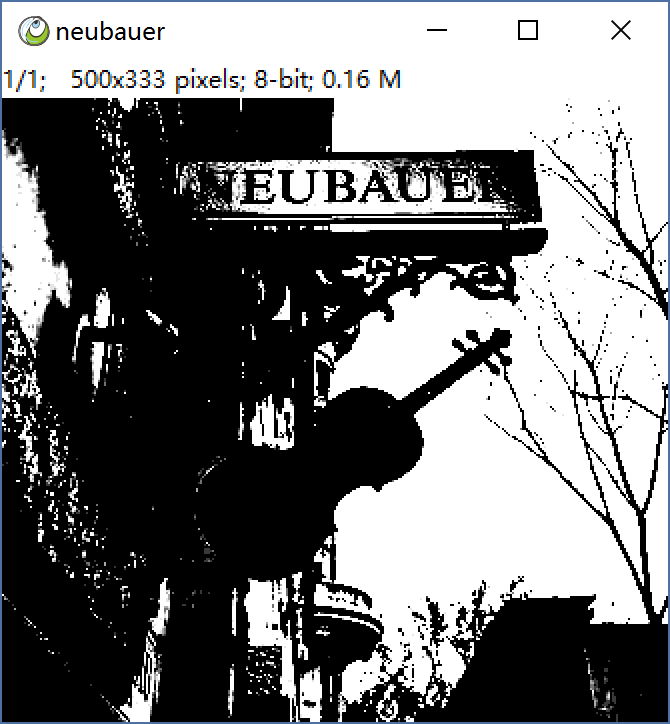
调整后直方图

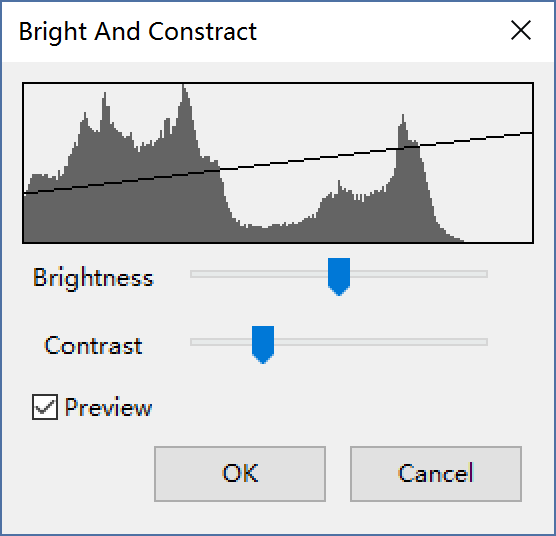
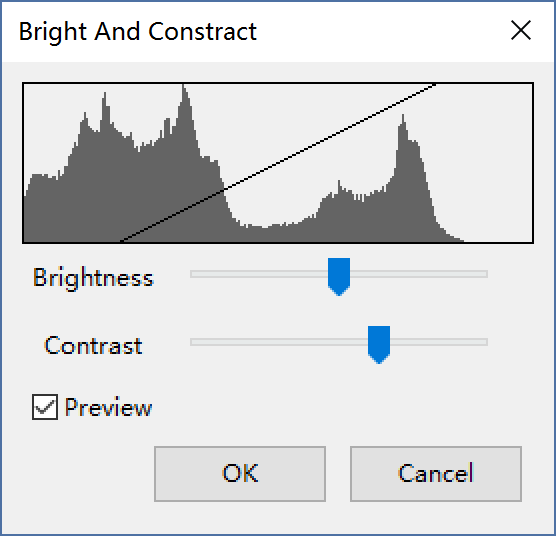
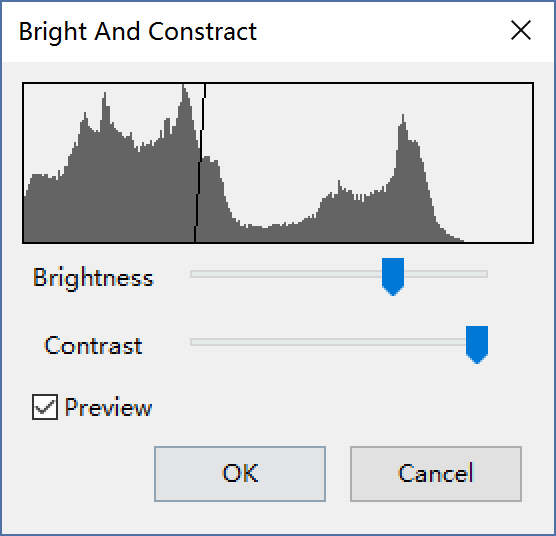
  

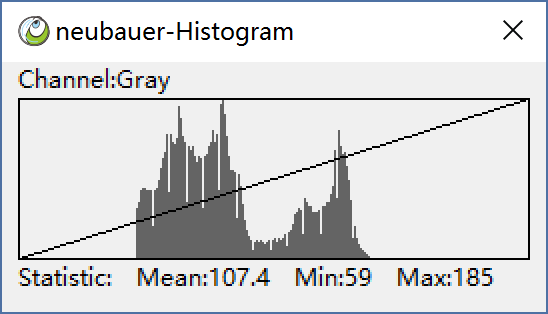
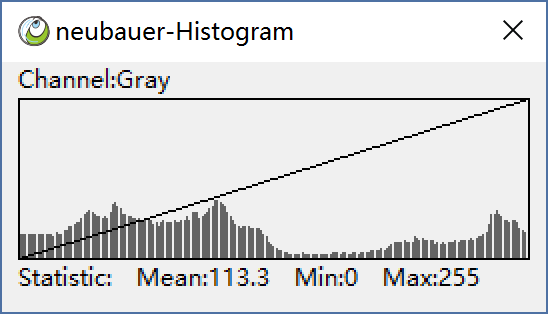
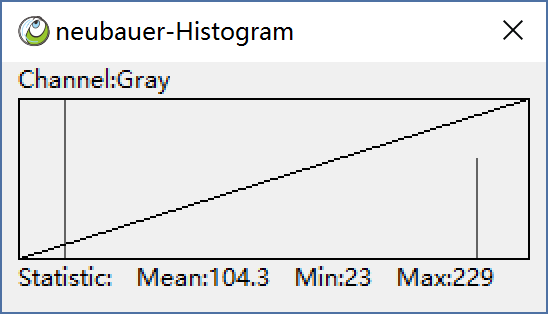
原图 提高亮度 降低亮度

**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**

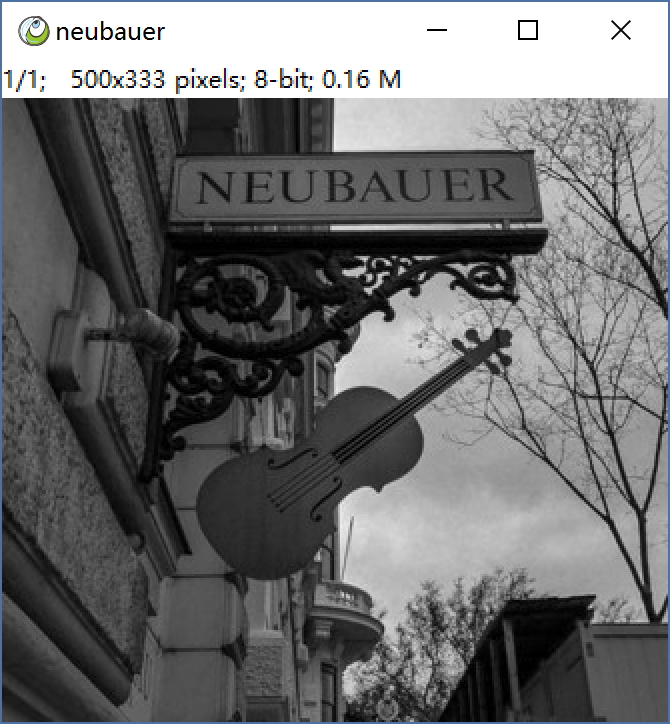
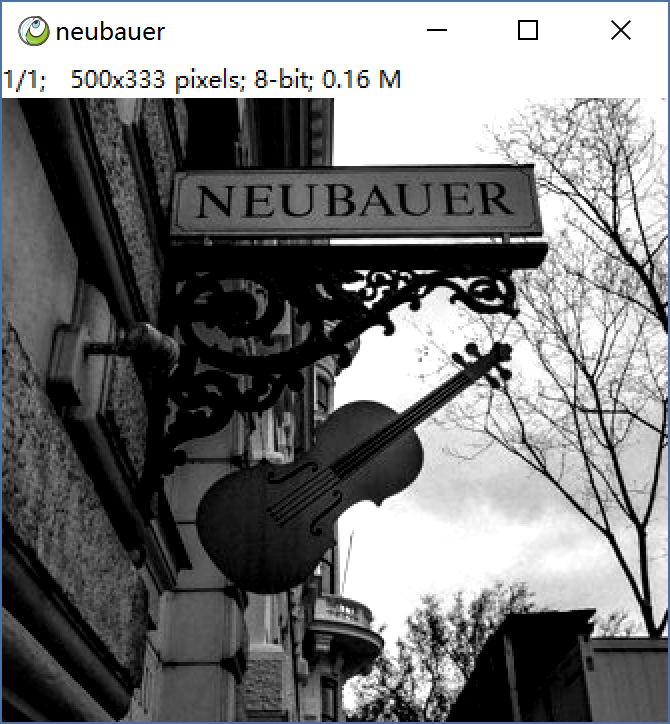
降低对比 提高对比 最高对比

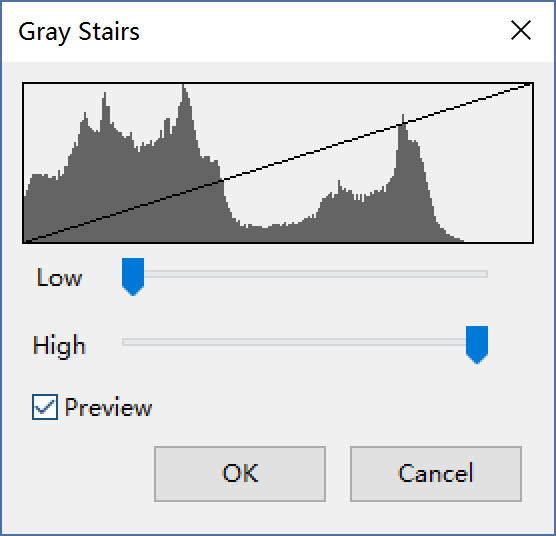
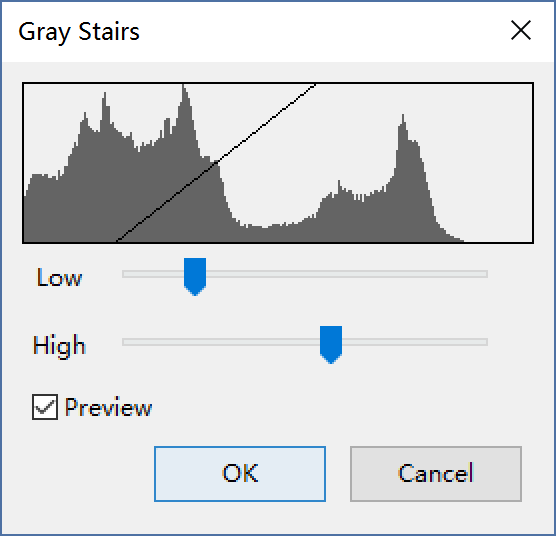
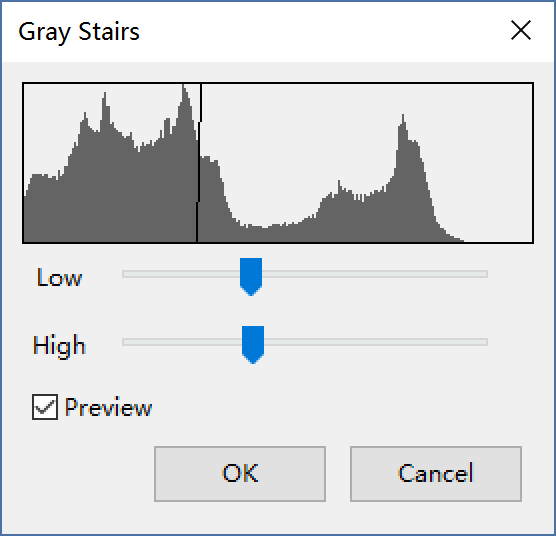
**调整对比度的本质**是像素乘法，当系数大于1时，对比度提高，当系数小于1时对比度降低，其实顾名思义，对比提高即增加差异性，我们看第一幅图，对比度降低，直方图向中间靠拢，图像的动态范围降低，因此视觉效果对比度下降。而对比度增加之后，我们观察直方图向两边扩张，并且中间出现了缝隙，其原因时当像素乘以大于1的数之后，四舍五入，自然灰形成空隙。而最后一幅图，对比度极端化之后，函数曲线几乎垂直，而结果直方图也分立成两个值（原本应时0，255，但这里为了展示清晰，做了其他处理）

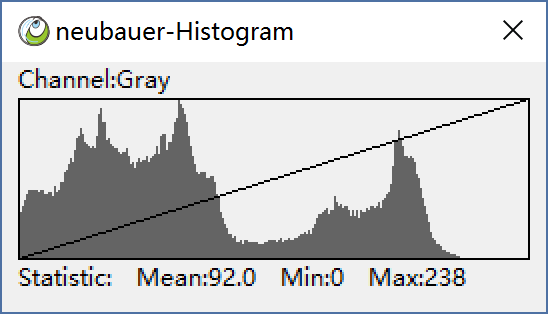
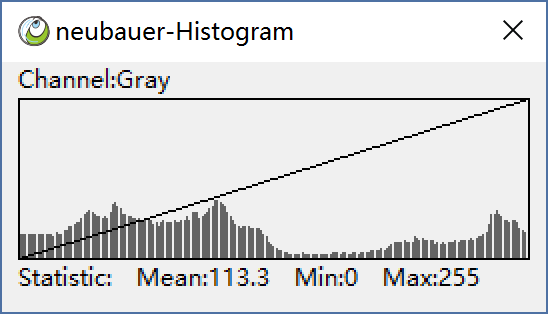
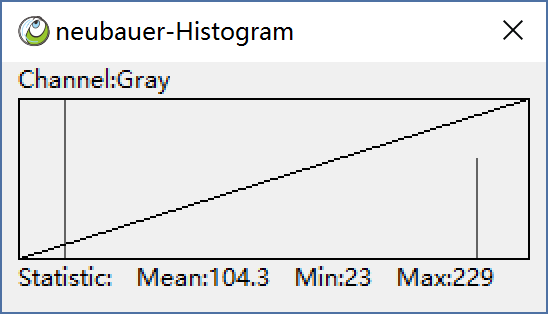
**色阶调整**

**$ 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。

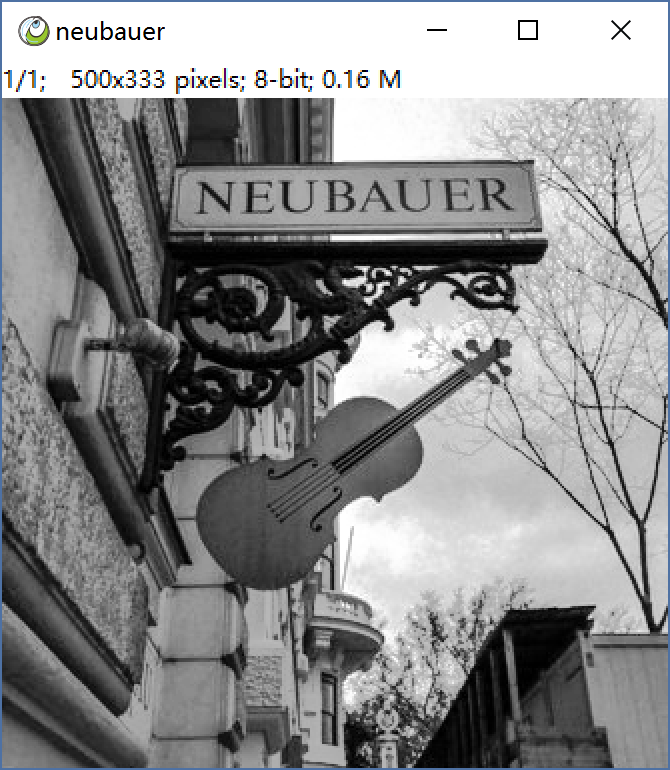
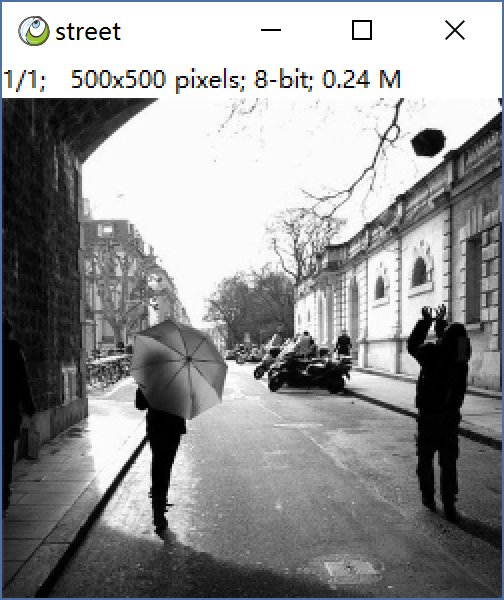
**直方图均衡化：**

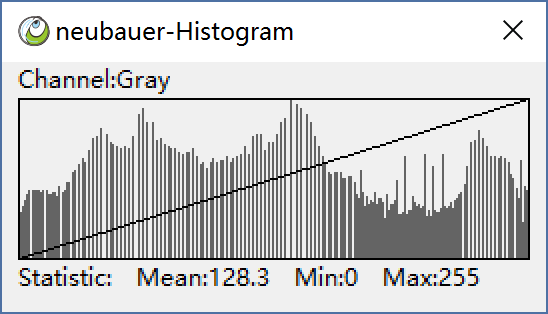
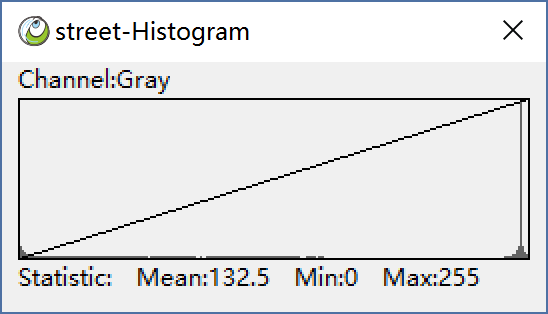
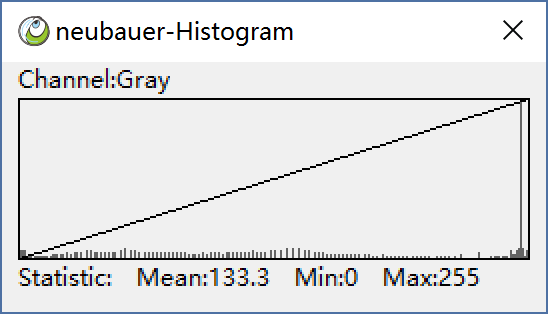
**$ Image> Adjust > Histogram Normalize**

**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

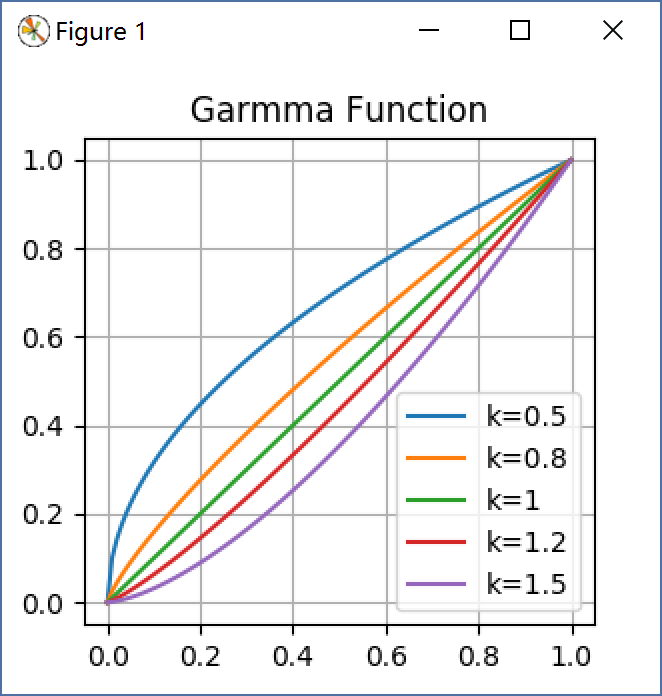
直方图均衡化可以使图像的像素在各个亮度区间分布均匀，因为我们只能移动与合并原直方图，而无法分裂，所以结果使一种概率意义下的均衡，高出的地方灰稀疏，低洼的地方会密集。而直方图匹配的结果是使调整后的图像的直方图尽可能接近目标直方图，这样视觉效果上会让两幅图给人的视觉体验相似。

**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**

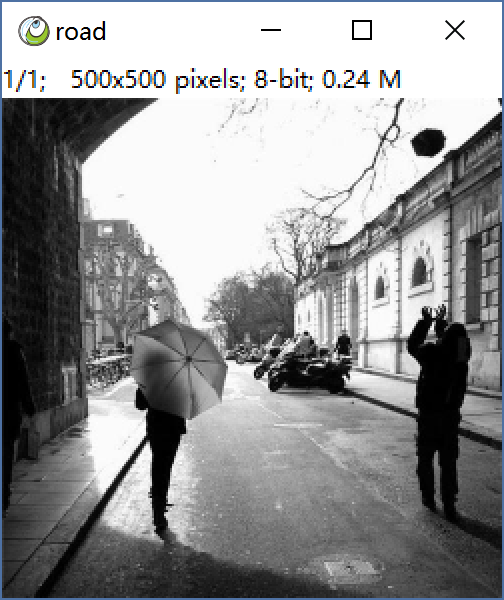
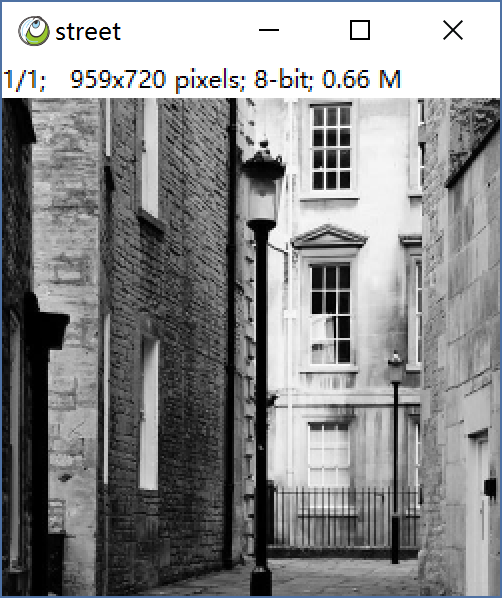
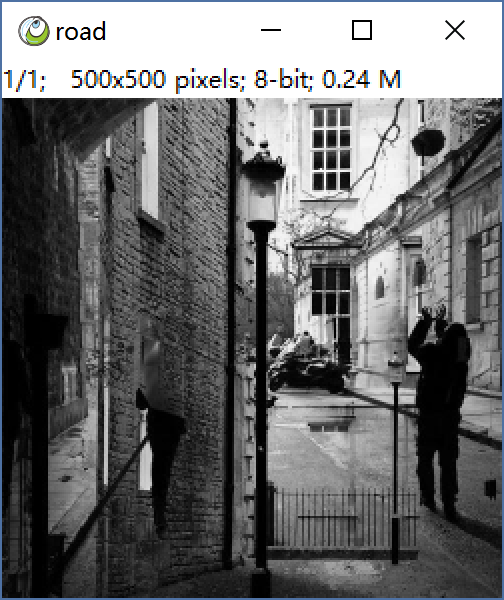
  

图1 图2 融合图像

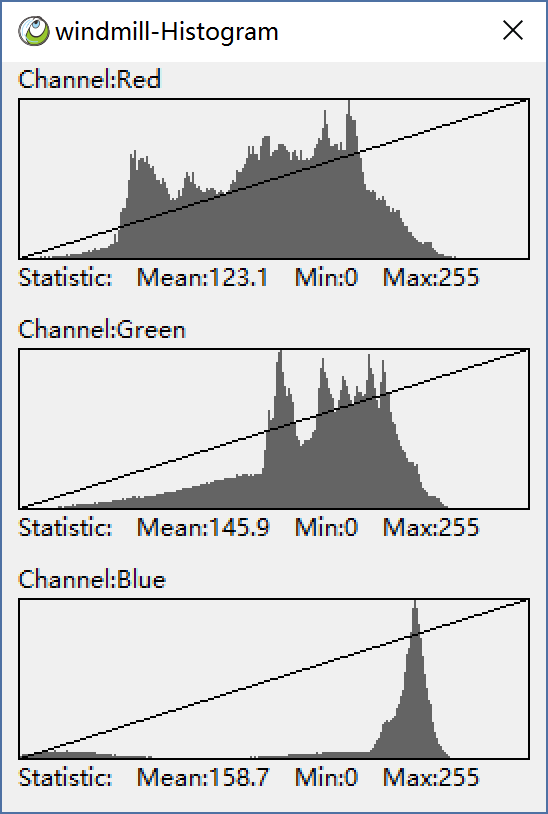
这里我们选用min运算，即两幅图对应位置像素做操作，保留较小的，因此我们看到的是两幅图中较暗的像素被保留下来。当然一般来说两幅图直接融合的结果会很不自然。这个功能更多是用于从原图进行一定的信息提取（比如轮廓）得到了新的图像，然后叠加回原图做效果展示，这个在后续的学习中回经常用到。

**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。

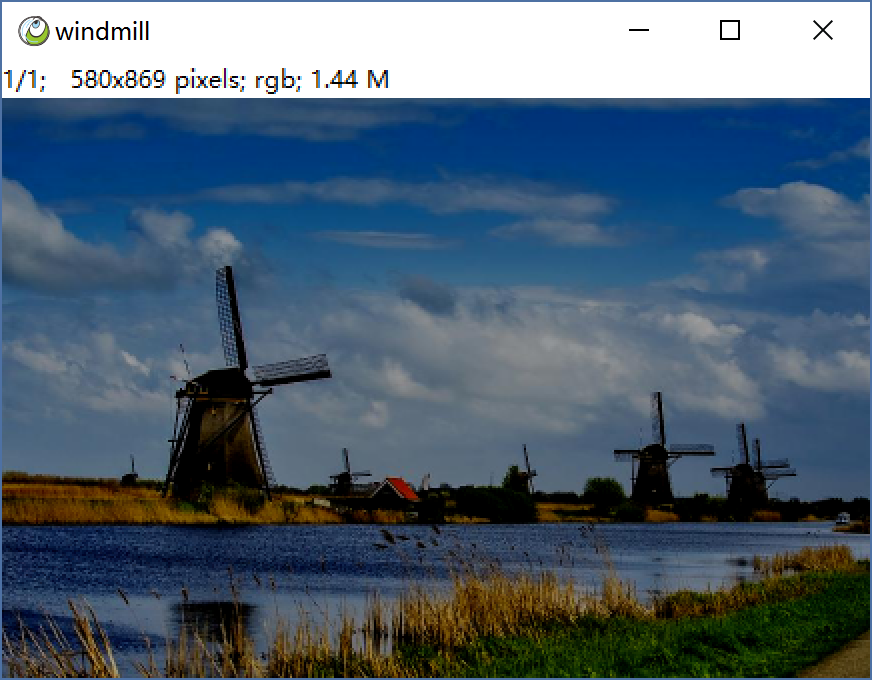
 

一副彩色照片 三个通道的直方图

**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。

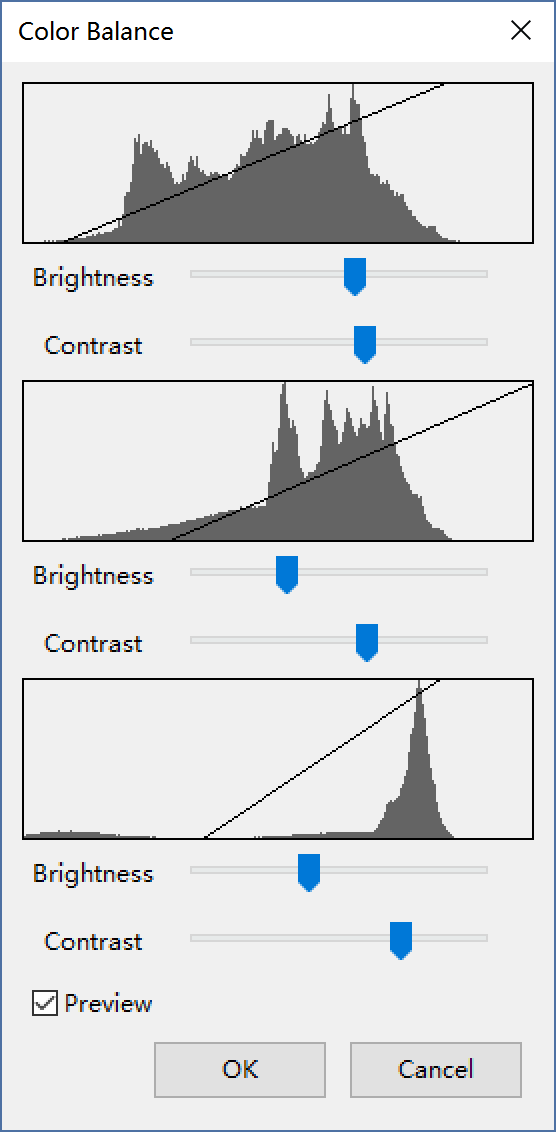
**三个通道分别处理**

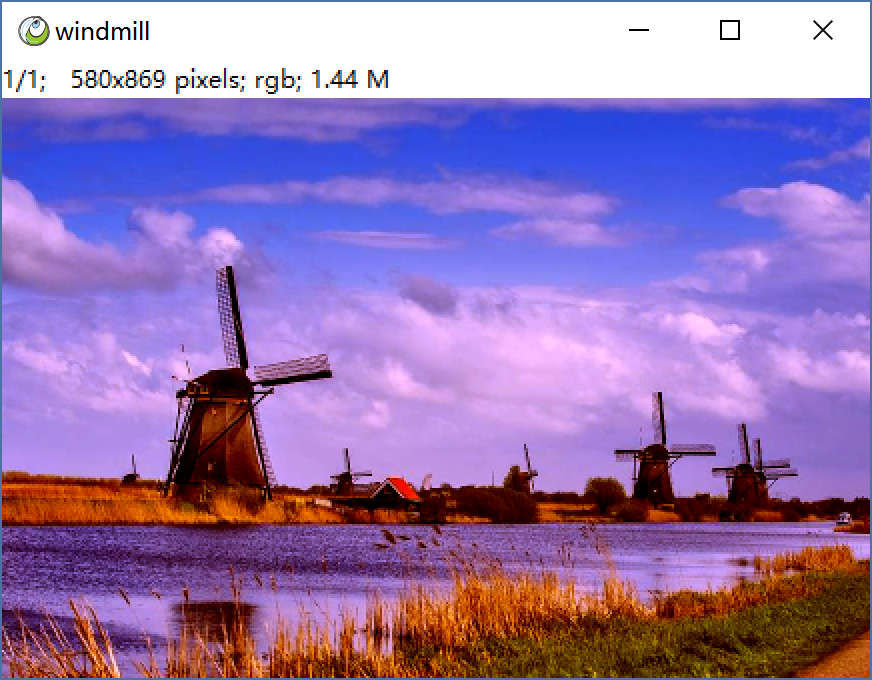
与单通道的亮度对比度，色阶相似，又有多通道的亮度对比度，色阶。不过由于分别调整各通道亮度，会导致分量比例变化，直接体现是图像的颜色发生变化，因此多通道调整亮度对比度在一些图像处理软件中叫做色彩平衡。

**$ 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

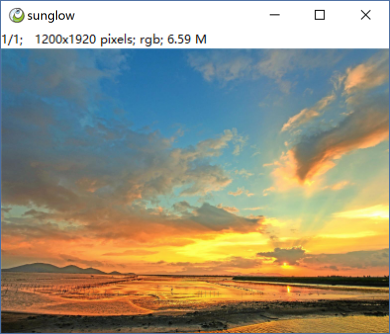
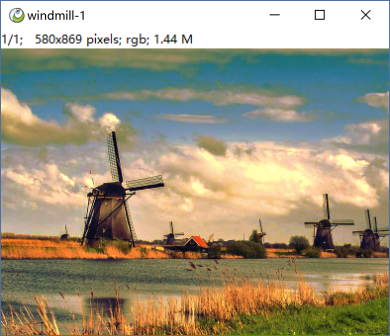
**调色技巧：**对于很多初学者，甚至是有美术基础的人，一个很大的困惑是调不出自己想要的效果，当然这需要一定的经验，不过大体上还是有技巧的，比如上图左边的参数，对应于右下方的暖色调调整。红色通道亮度增加，意味着整幅画面更加偏红，而绿色通道亮度降低，蓝色通道亮度有所下降，但由于对比度增强，因此天空更蓝，蓝色与红色混合后会使画面偏洋红色。我们知道红色给人以温暖，绿色给人以冷清，这样调整后，画面就更加温暖了。在后续我们学习了色彩模型之后，大家会有进一步认识。

**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。