# 直方图规定化与直方图均衡化优劣对比

## 实现平台：python

## 作者：刘浩

### **引言**：

对于数字图像的处理，在图像的空域上，往往杂乱无序，难以找出规律，所以我们很难直接在空域上对图像进行处理。然而，一幅图像的频域往往存在特殊的规律与特征，这就为我们从频域上处理图像创造了机会。而直方图均衡化与规定化，就是两种基于频域的增强图片对比度的方法。

### 直方图均衡化

原理

直方图均衡化是图像处理领域中利用图像直方图对对比度进行调整的方法。通过这种方法，亮度可以更好地在直方图上分布。这样就可以用于增强局部的对比度而不影响整体的对比度，直方图均衡化通过有效地扩展常用的亮度来实现这种功能。直方图均衡化通常用来增加许多图像的全局对比度，尤其是当图像的有用数据的对比度相当接近的时候。通过这种方法，亮度可以更好地在直方图上分布。这样就可以用于增强局部的对比度而不影响整体的对比度，直方图均衡化通过有效地扩展常用的亮度来实现这种功能。

优缺点

这种方法对于背景和前景都太亮或者太暗的图像非常有用，这种方法尤其是可以带来X光图像中更好的骨骼结构显示以及曝光过度或者曝光不足照片中更好的细节。这种方法的一个主要优势是它是一个相当直观的技术并且是可逆操作，如果已知均衡化函数，那么就可以恢复原始的直方图，并且计算量也不大。这种方法的一个缺点是它对处理的数据不加选择，它可能会增加背景噪声的对比度并且降低有用信号的对比度。

直方图均衡技术将原始图像的灰度直方图从比较集中的某个灰度区间变成在全部灰度范围内的均匀分布, 由于其算法简单, 无需借助外来因素的参数设置,可以自成系统的运行, 有效地增强图像对比度, 是一种常用的图像增强方法. 一直以来, 直方图均衡技术受到各个领域的重视, 比如在消费类电子产品邻域, 在均衡图像的同时希望保持图像的亮度信息 。

实现过程：

假设一个灰度图像大小为a\*b=n，让ni表示灰度i出现的次数，这样图像中灰度为i 的像素的出现概率是pi=ni/(x\*y)

L 是图像中所有的灰度数，n 是图像中所有的像素数, Ci表示灰度i的累计概率即

定义一个转换函数T：

由此建立了一个灰度级映射，原图像经过灰度级映射后即实现了直方图均衡化。

效果对比：

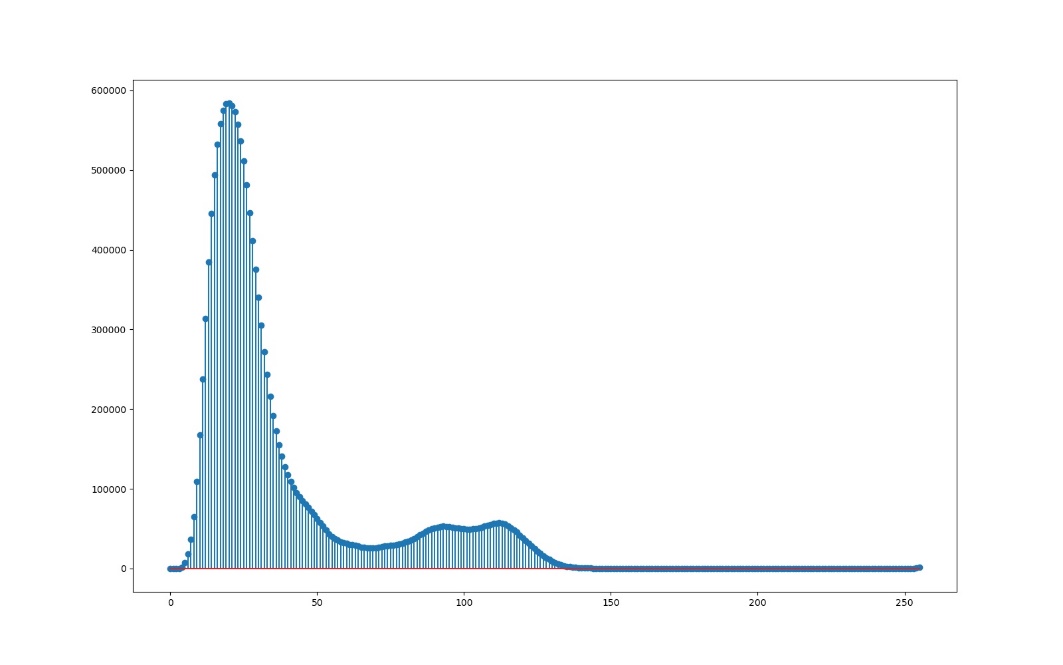


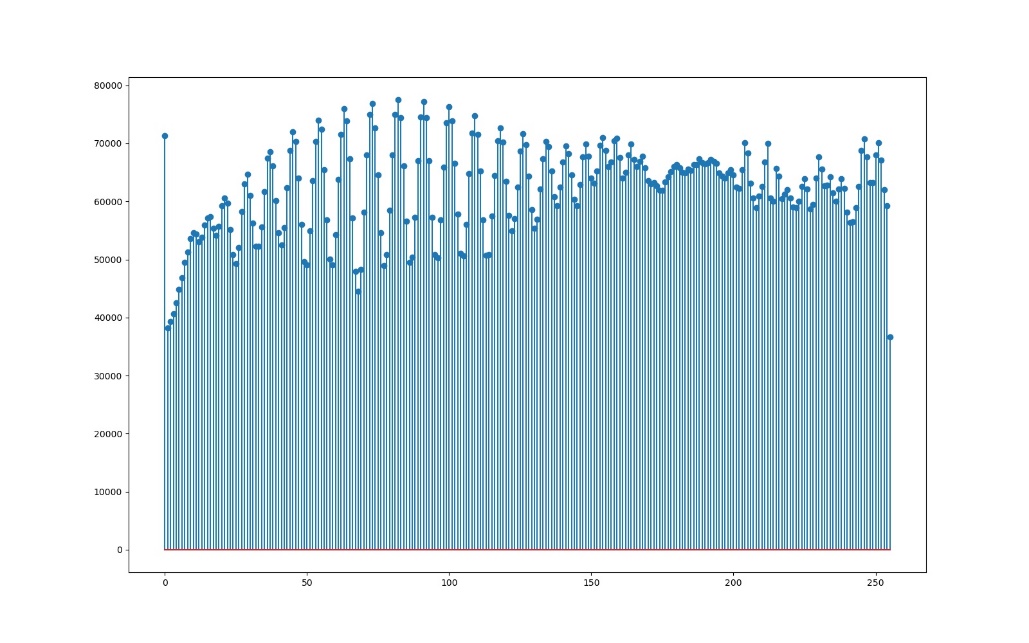
（a1）原图 (a2)均衡化后



(b1)原图 (b2)均衡化后

可以发现，均衡化后的图像亮度更高，可以看清楚暗处更多的细节，但是相对的多了许多白色噪声，我们可以通过频率直方图来观察发生了哪些改变。

 （b）原图频率分布直方图



（b）均衡化后频率分布直方图

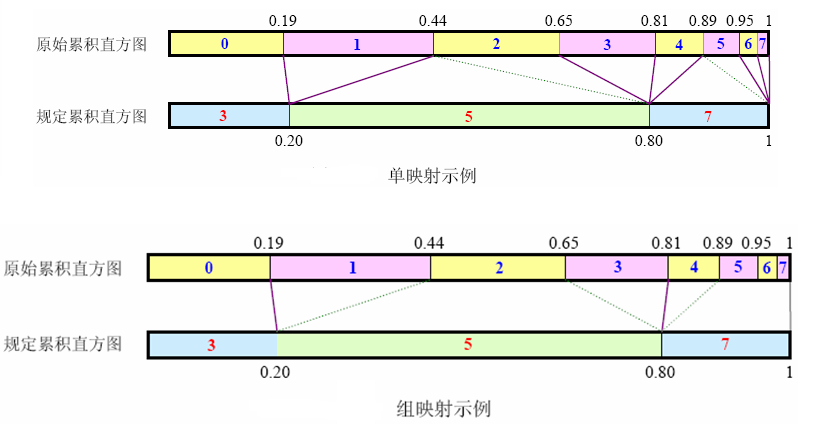
在频域上我们可以看见，原先的图像，像素主要分布在0-80的区间，在其他区间几乎没有分布，在均衡化后，频率近乎平均的分布在0-255区间。这就不难解释，图像的对比度为什么提高了。直方图均衡化在全图范围内，对分布集中的频率区间，放大了其差异，从而实现了图像全局增强。

### 直方图规定化原理

原理

直方图规定化的目的就是调整原始图像的直方图使之符合某一规定直方图的要求.设 Pr（r）和Pz（z）分别表示原始灰度图像和目标图像的灰度分布概率密度函数.根据直方图规 定化的特点与要求，应使原始图像的直方图具有Pz（z）所表示的形状。

由此我们可以建立GML与SML两种映射，达到这样的效果

实现过程

对于SML映射，每一个原始累计概率需要映射到最接近的规定累积概率，由此实现。

对于GML映射，每一个规定累计概率需要映射到最接近的原始累计概率，两个映射间的灰度级映射到对应的规定灰度级。

效果对比

SML基于不同规定累计概率直方图的变换



**1,10,20,40,80,160,255----0.1，0.2，0.3，0.4，0.5，0.6，1**

****

1,11,21,31,41,80,160,255----**0.1，0.2，0.3，0.4，0.5，0.6，0.8，1**



1,11,21,31,41,80,160,220,255----**0.1，0.2，0.3，0.4，0.5，0.6，0.7，0.9，1**

GML基于不同规定累计概率直方图的变换

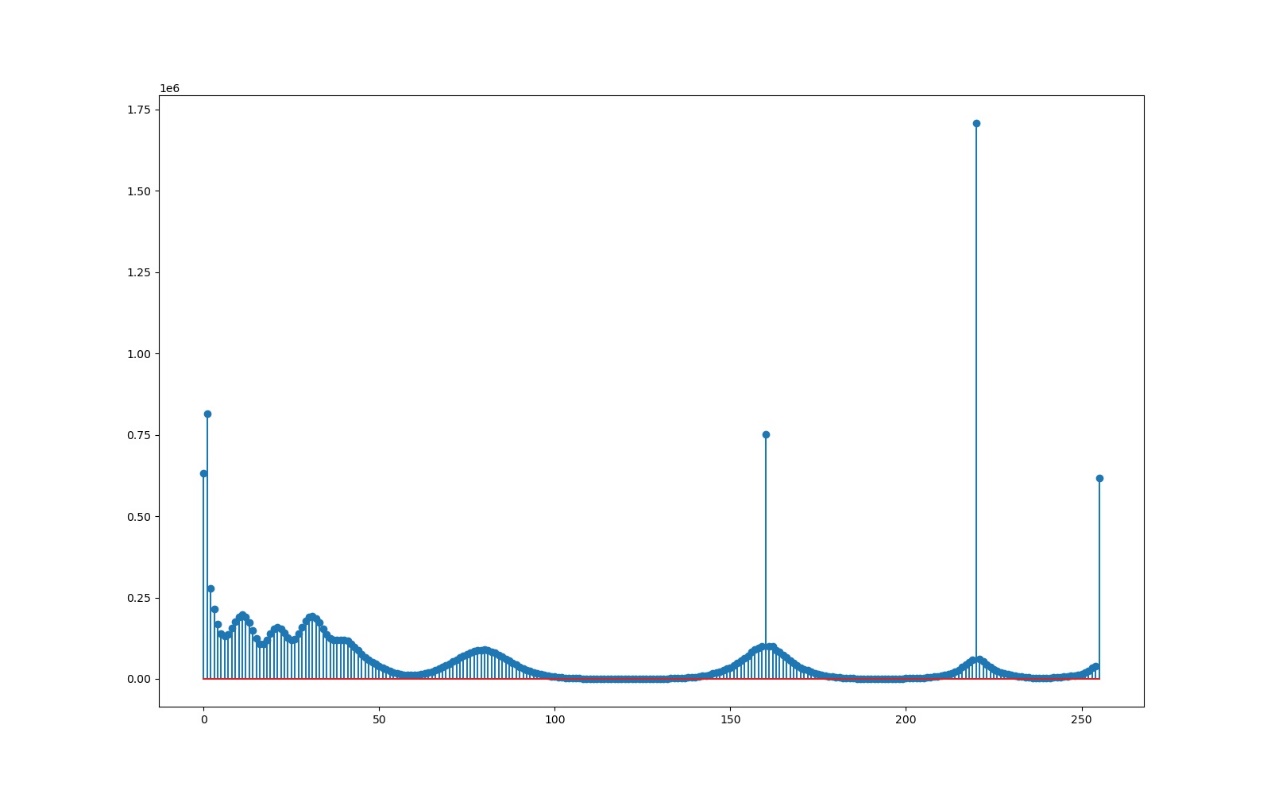
**1,10,20,40,80,160,255----0.1，0.2，0.3，0.4，0.5，0.6，1**



1,11,21,31,41,80,160,255----**0.1，0.2，0.3，0.4，0.5，0.6，0.8，1**

1,11,21,31,41,80,160,220,255----**0.1，0.2，0.3，0.4，0.5，0.6，0.7，0.9，1**

可以看到：选用不同的映射关系，不同的规定累计概率直方图，得出了不同效果的增强图像。在本次实验中以映射关系为SML，规定累计直方图为1,11,21,31,41,80,160,220,255----0.1，0.2，0.3，0.4，0.5，0.6，0.7，0.9，1的规定化效果最好。既增强了暗处的细节，又没用出现过多的白噪声。而此图对应的频率图如下，像素分布相对平均，又可以体现出以黑色为主。



（b）SML映射后 的频率直方图

# 优劣分析

直方图均衡化： 自动增强

效果不易控制

总得到全图增强的结果

直方图规定化： 有选择地增强

须给定需要的直方图

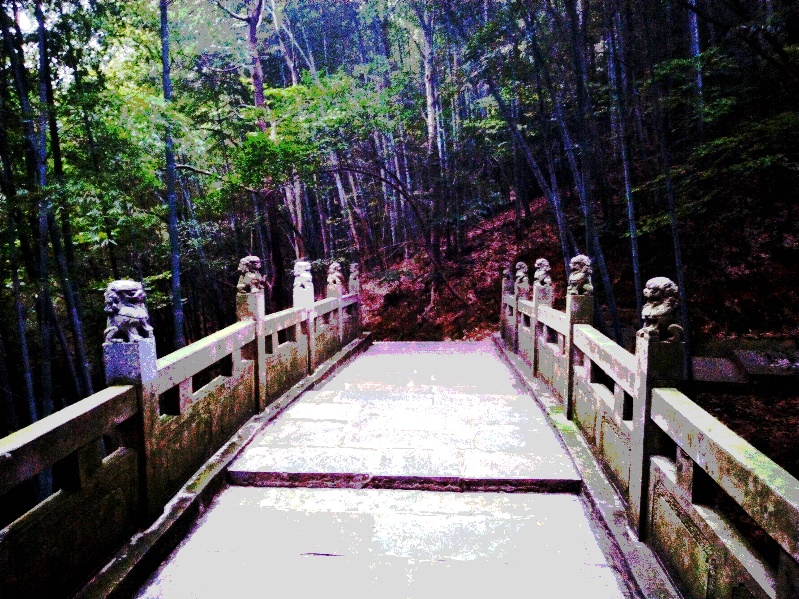
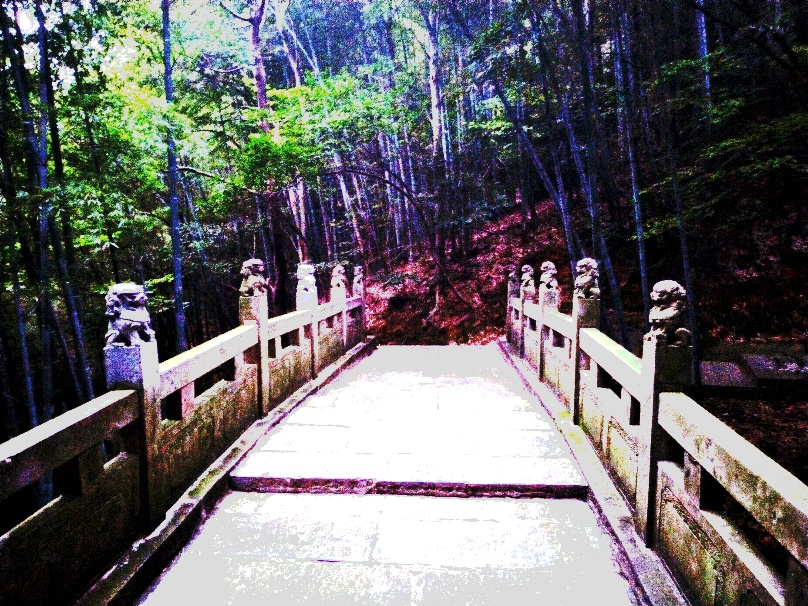
特定增强的结果

扩展：彩色图的直方图规定化与均衡化效果



原图

直方图均衡化

 SML

GML

主要代码讲解：

（1）灰度直方图.py文件

用于读取一幅图像并生成它的灰度直方图

（2）以caise开头的文件

对图像进行彩色的变换

（3）其他文件函数简介：

|  |  |
| --- | --- |
| def getfreqnum(arr): | 输入图像二维数字矩阵，得到图像灰度级频数分布（一个256长的数组） |
| def getfreq(freqnum,count): | 输入图像灰度级频数分布，与像素整数，得到图像灰度级频率分布 |
| def getsumfreq(freq): | 输入图像灰度级频率分布得到图像灰度级累积频率分布 |
| def getTransArr(sumfreq) | 输入图像灰度级累积频率分布，得到图像均衡化映射数组（一个256长的数组，存储着变换后对应的灰度级） |
| def getTransArrSML(sumfreq,setfreq): | 输入图像灰度级累积频率分布与直方图规定累计概率，得到图像均衡化映射数组（一个256长的数组，存储着变换后对应的灰度级） |
| def getTransArrGML(sumfreq,setfreq): | 输入图像灰度级累积频率分布与直方图规定累计概率，得到图像均衡化映射数组（一个256长的数组，存储着变换后对应的灰度级） |
|  |  |

# 代码附录

