

# 增强图像对比度

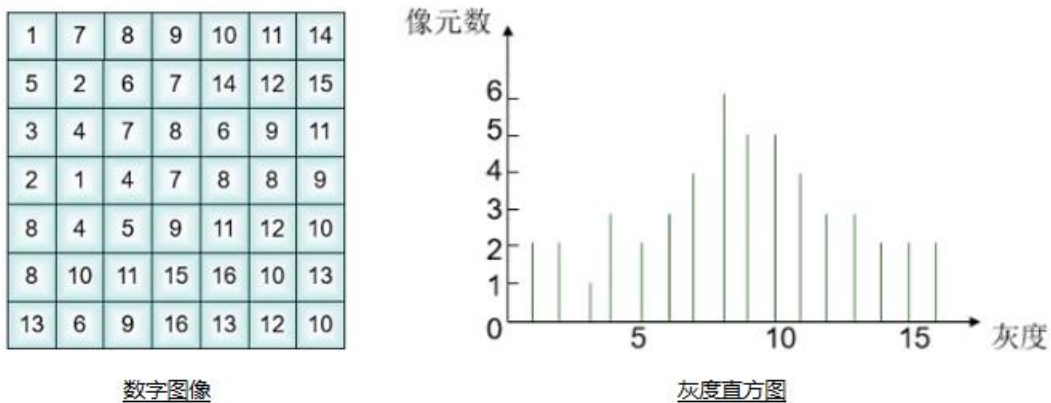
## 结构说明（目录）：

1. 灰度直方图解析
  - 1.1 灰度直方图性质
  - 1.2 对灰度直方图性质 3 的说明
2. 全局直方图均衡化
  - 2.1 简介
  - 2.2 理论分析
  - 2.3 MatLab 编程实现过程
  - 2.4 效果展示
  - 2.5 总结（优缺点）
3. 对比度受限制全局直方图均衡化
  - 3.1 简介
  - 3.2 MatLab 编程实现过程
  - 3.3 效果展示
  - 3.4 总结（优缺点）
4. 自适应直方图均衡化
  - 4.1 简介
  - 4.2 理论分析
  - 4.3 MatLab 编程实现过程
  - 4.4 效果展示
  - 4.5 总结（优缺点）
5. 受限制自适应直方图均衡化
  - 5.1 简介
  - 5.2 算法说明
  - 5.3 MatLab 编程实现过程
  - 5.4 效果展示
  - 5.5 总结（优缺点）

## 1 灰度直方图解析：

直方图是一种统计报告图，由一系列高度不等的纵向条纹或线段表示数据分布的情况。一般用横轴表示数据类型，纵轴表示分布情况。

在灰度直方图中横轴表示灰度级，纵轴表示每一灰度级具有的像元数或者像元数占总像元数的比例值。



直方图统计																
灰度级	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
像元数	2	2	1	3	2	3	4	6	5	5	4	3	3	2	2	2
比例值	0.04	0.04	0.02	0.06	0.04	0.06	0.08	0.12	0.1	0.1	0.08	0.06	0.06	0.04	0.04	0.04

图 1

### 1.1 灰度直方图的性质：

1. 直方图表示一幅图像中不同像素值的统计信息，它只能表示每个灰度值在图像中出现的总数或者是概率，而与灰度值在图中的位置信息无关。
2. 任何一幅图片具有与之对应的唯一的灰度直方图，但是不同的图可能具有一样的直方图。
3. 可以根据灰度值在灰度直方图中的分布情况判断一幅图片质量的好坏。

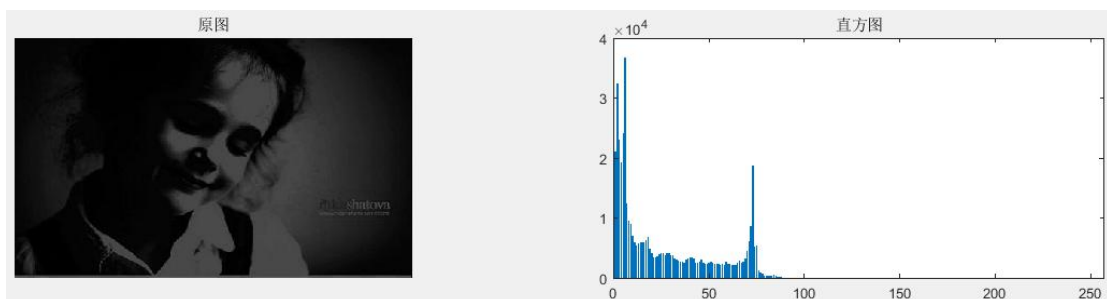


图 2

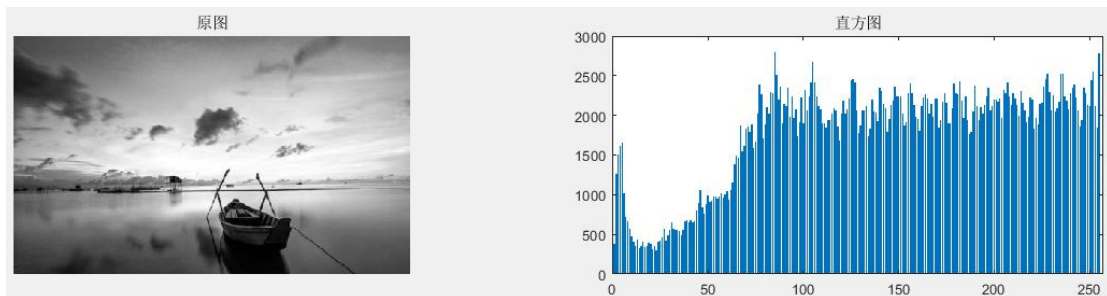


图 3

### 1.2 对灰度直方图性质 3 的说明：

以上是两幅灰度图的原图和与之对应的灰度直方图，可以看出第一幅图内容比较模糊并

且整体效果看上去非常的暗，第二幅图比较清晰并且小船周围的景色显示的非常清楚。那么我们可以对比一下这两幅图对应的直方图，第一幅图的灰度值集中在灰度级较低的一端，较高的一段基本上没有灰度值分布。第二幅图的灰度值分布比第一幅图的均匀的多，几乎所有的灰度级上都有相应的灰度值分布。

通过上一段落的分析，我们可以得出一幅图像的灰度直方图几乎覆盖了整个灰度的取值范围，并且除了个别灰度值的个数较为突出，整个灰度值分布近似于均匀分布，那么这幅图像就具有较大的灰度动态范围和较高的对比度，同时图像的细节更为丰富。

## 2 全局直方图均衡化 (Histogram equalization)

### 一：简介

直方图均衡化是一种增强数字图像对比度的方法，它的基本思想是将输入图像的灰度直方图分布均衡化，因此图像灰度值在直方图中均匀分布。从而可达到图像对比度整体增强的效果。

### 二：理论分析

直方图的变换是一种灰度变换，灰度变换的变换函数决定了输入随机变量与输出随机变量间的关系。一幅图像是二维的离散数据，因此不利于数学推导。我们需要先通过连续变量进行推导，然后将其推广到离散的情况。

假设  $r$  和  $s$  分别代表原图像灰度级和直方图均衡化后的图像灰度级，为了更加客观，需要将  $r$  和  $s$  进行归一化，即： $0 \leq r, s \leq 1$ 。

对于一幅图像，归一化之后灰度级分布在  $0 \leq r \leq 1$  范围内，对  $[0,1]$  区间内的任意一个  $r$  值进行变换：

$$s = T(r)$$

该变换式应满足条件：

1. 对于  $0 \leq r \leq 1$ ，有  $0 \leq s \leq 1$
2.  $T(r)$  在  $0 \leq r \leq 1$  上严格单调递增

从  $s$  变换到  $r$  的反变换用下式表示

$$r = T^{-1}(s)$$

$r$  的概率密度为  $p_r(r)$ ， $s$  的概率密度可由  $p_r(r)$  求出

$$p_s(s) = \left( p_r(r) \frac{dr}{ds} \right) \Big|_{r=T^{-1}(s)}$$

假定变换函数为

$$s = T(r) = \int_0^r p_r(w) dw$$

式中  $w$  是积分变量，而  $\int_0^r p_r(w) dw$  就是  $r$  的累积分布函数。

对式中的  $r$  求导，则

$$\frac{ds}{dr} = \frac{dT(r)}{dr} = p_r(r)$$

再把结果带入前面公式，可得到

$$\begin{aligned} p_s(s) &= \left[ p_r(r) \cdot \frac{dr}{ds} \right]_{r=T^{-1}(s)} = \left[ p_r(r) \cdot \frac{1}{ds/dr} \right]_{r=T^{-1}(s)} \\ &= \left[ p_r(r) \cdot \frac{dr}{ds} \right] = 1 \end{aligned}$$

由以上过程可得，用  $r$  的累积分布函数作为变换函数可以使图像灰度值分布均匀，扩展了像素取值的动态范围。

推广到离散形式：

设图像的像元数为  $n$  且共有  $m$  个灰度级， $n_k$  代表灰度级为  $r_k$  的灰度值的数目，则第  $k$  个灰度级出现的概率可表示为：

$$p_r(r_k) = \frac{n_k}{n}, 0 \leq r_k \leq 1, k = 0, 1, \dots, m-1$$

变换函数  $T(r)$  可改写为：

$$s_k = T(r_k) = \sum_{j=0}^k p_r(r_j) = \sum_{j=0}^k \frac{n_j}{n}, 0 \leq r_k \leq 1, k = 0, 1, \dots, m-1$$

均衡化后各像素的灰度值可直接由原图像的直方图算出。

### 三：MatLab 编程实现过程

第一步：导入灰度图像

第二步：计算图像的灰度直方图

第三步：归一化直方图分布  $H_s(i) = h(i)/N_f$ ,  $N_f = m * n$  (像素总数)

第四步：计算直方图的概率累计分布

第五步：灰度映射

### 四：效果展示

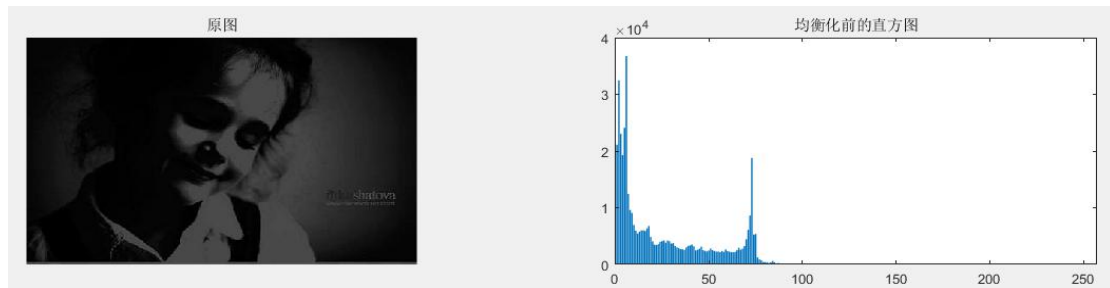


图 4

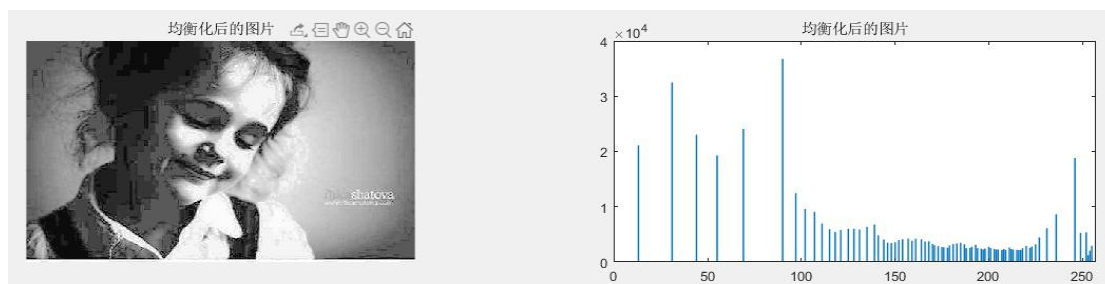


图 5

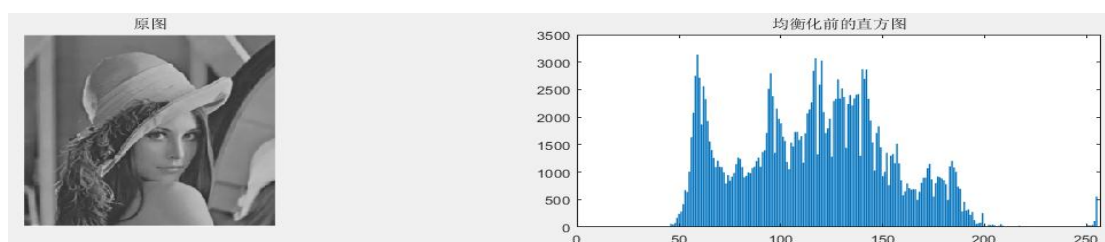


图 6

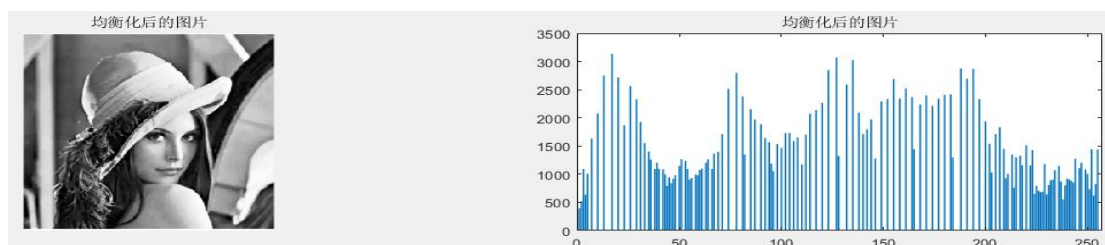


图 7

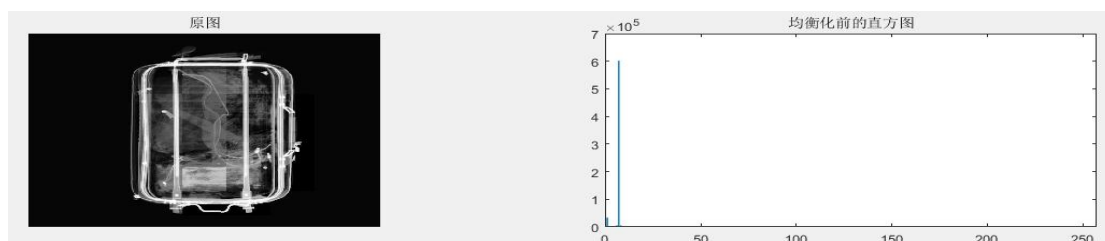


图 8

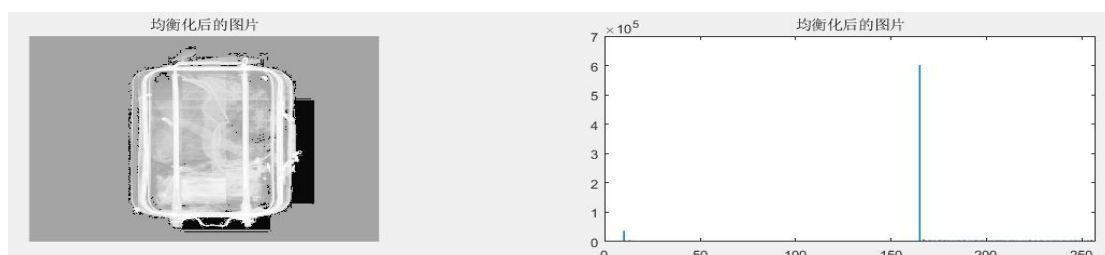


图 9

### 总结（优缺点）：

1. 从图 4 和图 5 以及图 6 和图 7 看出，当图像整体偏暗或者偏亮的时候，全局直方图均衡化的方法的效果很好。
2. 直方图均衡化是一种全局处理方式，它对处理的数据不加选择，可能会增强背景干扰信息的对比度和降低有用信息的对比度，从图 8 和图 9 可以看出，通过均衡化以后，图像所有区域的对比度增强了，效果不是很好。

3. 全局直方图均衡化后图像的灰度级将会减少，某些细节将会消失。

4. 图 8 中的原图其实可以用全局直方图均衡化来很好的提高对比度，原因是有些图像的特定灰度级上的灰度值数量太多或比例太高（直方图有突出高峰），因此经过均衡化后对比度将会不自然的过分增强，为了解决此问题我们可以用对比度受限制的方法。

### 3 对比度受限制全局直方图均衡化（Contrast Limited Histogram equalization）

#### 一：简介

该方法与全局直方图均衡化的原理一样，但是它可以处理全局直方图的一个特殊情况。有些图中会出现属于某种灰度级的灰度值的数目远大于其他灰度值的情况，该情况影响图片的质量（对比度过分增强），因此我们计算 CDF 前用预先定义的阈值来裁剪直方图，以此达到限制增强对比度的目的。

#### 二：MatLab 编程实现过程

第一步：导入灰度图像

第二步：计算图像的灰度直方图

第三步：定义的阈值来裁剪直方图（裁剪的部分平分给所有的灰度级）

第三步：归一化直方图分布  $H_s(i) = h(i) / N_f$ ,  $N_f = m * n$  (像素总数)

第四步：计算直方图的概率累计分布

第五步：灰度映射

#### 三：效果展示：

阈值 = (输入图像高度 \* 输入图像宽度) / 256

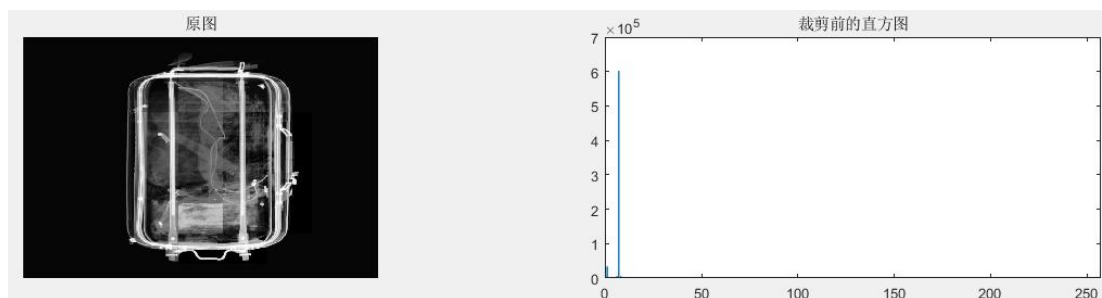


图 10

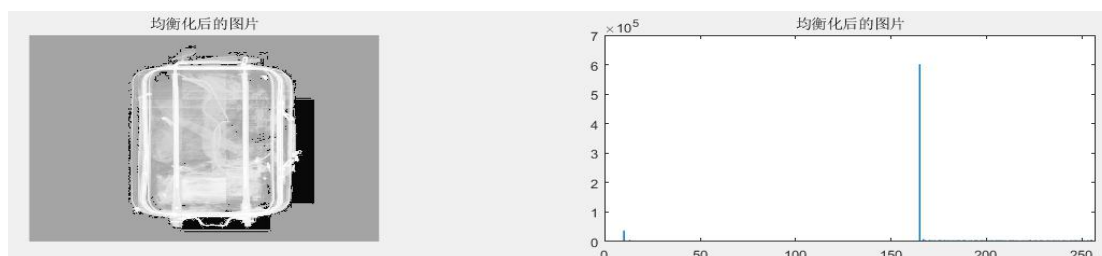


图 9（副）

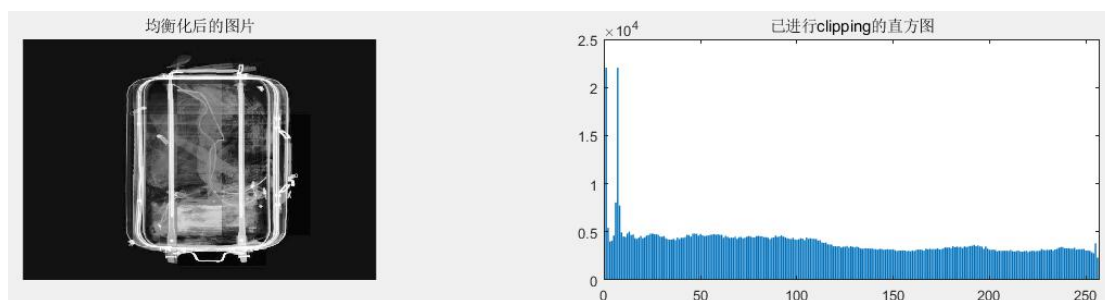


图 11

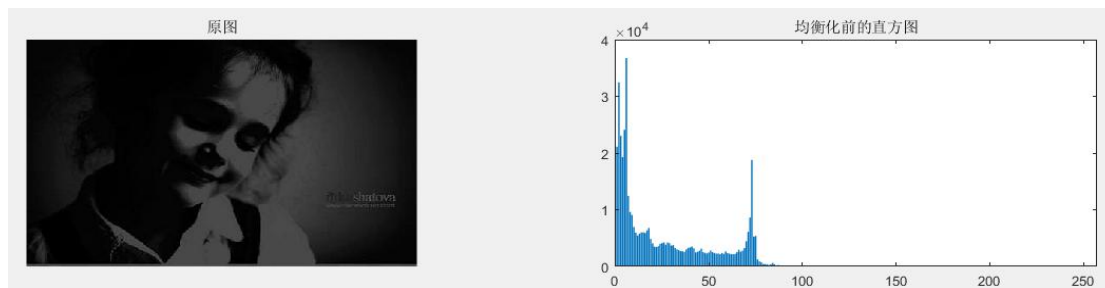


图 4 (副)

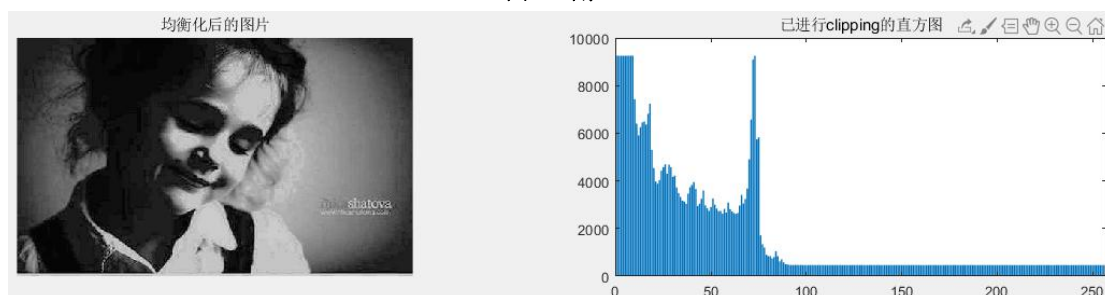


图 12

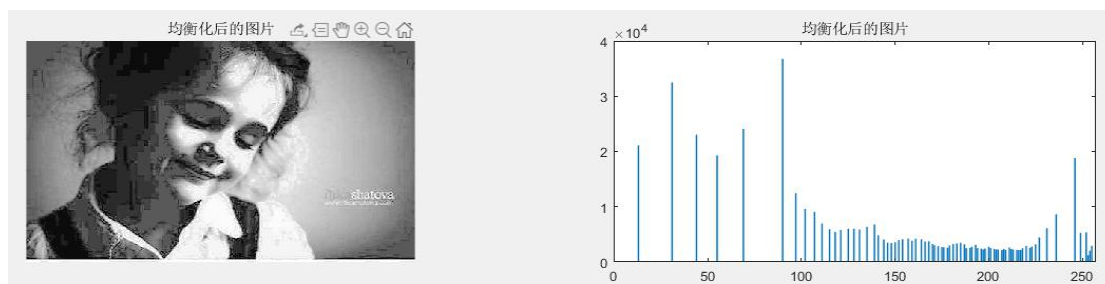


图 5 (副)

#### 四：总结（优缺点）

1. 从图 10、图 11 和图 9（副）看出，限制对比度直方图均衡化方法很好的解决了增强图 10 中原图的对比度的问题。因此该方法可以称得上是特殊的直方图均衡化，用来解决具有突出高峰的直方图的图像对比度增强问题。

2. 从图 4（副）、图 12 和图 5（副）看出，对该图像来说对比度限制均衡化方法的对比度增强效果比没有对比度限制的均衡化方法的对比度增强效果差。这说明，当图像整体偏暗或者偏亮的时候，全局直方图均衡化方法的效果比对比度受限制全局直方图均衡化方法的好。

## 4 自适应直方图均衡化 (Adaptive histogram equalization)

### 一：简介

自适应直方图均衡化方法和普通的直方图均衡算法不同，该方法通过计算图像的局部直方图，然后重新分布亮度来改变图像对比度。因此，该算法更适合于改进图像的局部对比度以及获取更多的图像细节。

### 二：理论分析

每个像素通过其周边一个矩形范围内的像素的直方图进行均衡化，均衡的方式跟普通的均衡化算法是一样的，变换函数和像素周边的累积直方图函数 (CDF) 成比例。图像边缘的像素需要特殊处理，因为边缘像素的领域 (被分的一个小块) 不完全在图像内部。这个通过镜像图像边缘的行像素或列像素来解决。直接复制边缘的像素进行扩充是不合适的。因为这会导致带有剑锋的领域直方图。

如果按照关于自适应直方图均衡化方法的早些时候的观点去编写程序，那么程序的复杂度会非常的高。因为早些时候提出论文的思想是需要对图像的每个像素点进行遍历，用像素点周围  $W * W$  的窗口进行计算直方图变换的 CDF，然后对该像素点进行映射。这是一个复杂度非常高的算法： $O(m*n*((w*w)*256))$ 。

为了解决复杂度的问题，我们可以用线性插值法。具体过程：先对图像进行分块，然后计算各个块的直方图以及 CDF。然后对于图像的每一个像素点，找到其邻近的四个块 (图像边缘需要特殊处理,) 的中心像素点, 并获取其 CDF 映射值  $f_a(D), f_b(D), f_c(D), f_d(D)$ , 然后进行双线性插值得到最终该像素点的映射值。

双曲线插值公式:

$$f(D) = (1 - \Delta y)((1 - \Delta x)f_a(D) + \Delta x f_c(D)) + \Delta y((1 - \Delta x)f_b(D) + \Delta x f_d(D))$$

特殊区域计算说明：红色区域只使用其最近的块的 CDF 进行映射，绿色区域用邻近的两个窗口的 CDF 映射值进行线性插值。

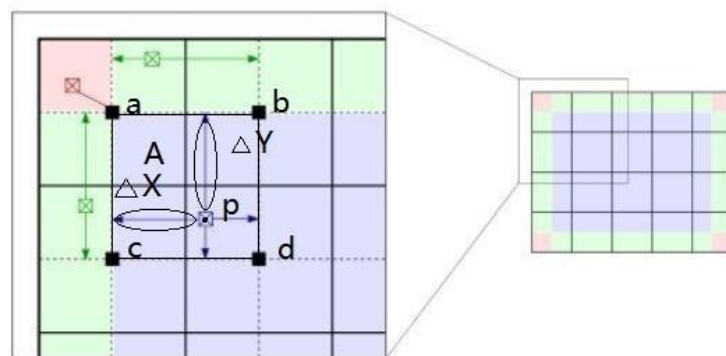


图 13

### 三：MatLab 编程实现过程

第一步：导入 (计算) 灰度图像

第二步：对图像进行分块

第三步：对每一个块进行直方图和 CDF 计算

第四步：利用双曲线插值公式，得到图像的所有像素的映射值

第五步：显示均衡化后的图像



四：效果展示

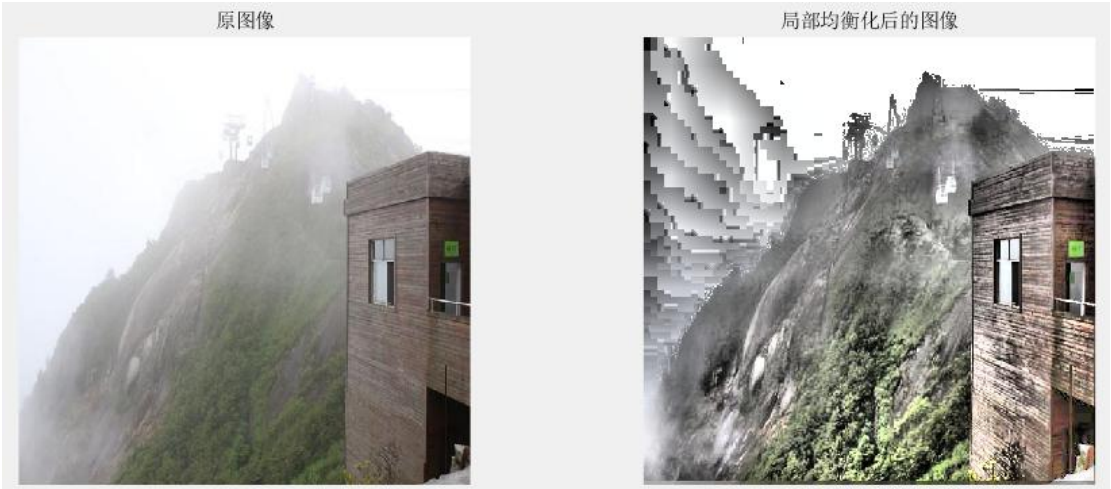


图 14

块数目=3\*3



图 15

块数目=5\*5

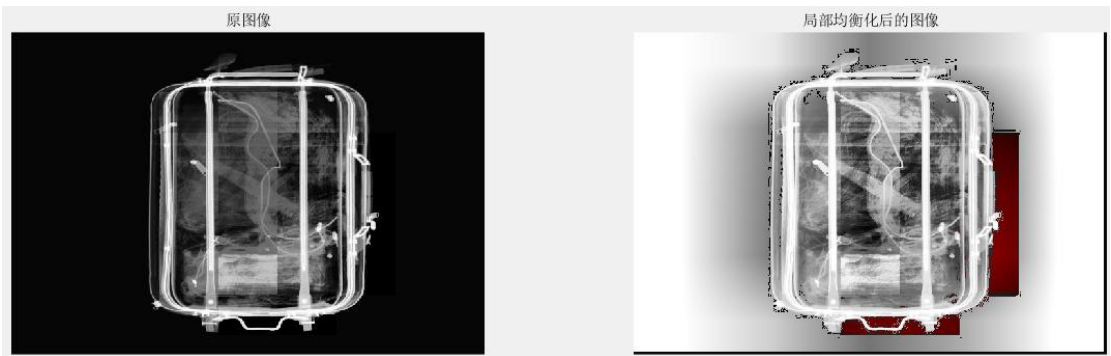


图 16

块数目=8\*8

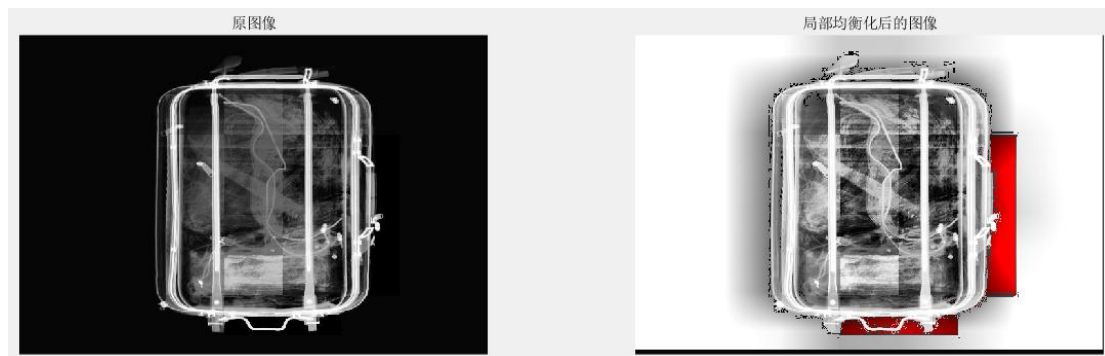


图 17

总结（优缺点）：

1. 自适应直方图均衡化算法解决了全局直方图均衡化中的“全局直方图均衡化后图像的灰度级将会减少，某些细节将会消失。”的问题，因为它的原理是将图像分块来增强图像对比度。

2. 当图像整体偏暗或者偏亮的时候全局直方图均衡化算法的效果好，所以对“对处理的数据不加选择，可能会增强背景干扰信息的对比度和降低有用信息的对比度”的问题没有能力处理。自适应直方图均衡化将图片分块来进行均衡化，解决了该问题。

3. 从图 14 可以看到，图片左上方区域（平坦区，因为该区的灰度级相似）的雾通过局部均衡化后变得模糊不清，这说明当块里的某个区域的灰度级比较相似的时候，该区域的直方图是尖状的。变换函数会将此很窄范围内的像素映射到整个块区域，这导致平坦区域（灰度级相似的）的一些噪音放大，影响增强对比度效果。

4. 从图 15、图 16 和图 17 看出，当图像所分的块比较小的时候，对比度得到增强。当块的比较大的时候对比度将会降低。

## 5 限制对比度自适应直方图均衡（Contrast Limited Adaptive histogram equalization）

### 一：简介

限制对比度自适应直方图均衡算法与自适应直方图均衡算法的唯一区别就是对比度限幅，主要作用是解决自适应直方图均衡的过度放大噪音的问题。

### 二：算法说明

在我们计算每一个小块的 CDF 前用预先定义的阈值来裁剪该小块直方图，以此达到限制增强对比度的目的。

### 三：MatLab 编程实现过程

第一步：导入（计算）灰度图像

第二步：对图像进行分块

第三步：对每一个块进行直方图分布计算

第四步：通过预先设置的阈值来裁剪该小块直方图

第五步：计算每个块的 CDF 累积分布

第六步：利用双曲线插值公式，得到图像的所有像素的映射值

第五步：显示均衡化后的图像

四：效果展示

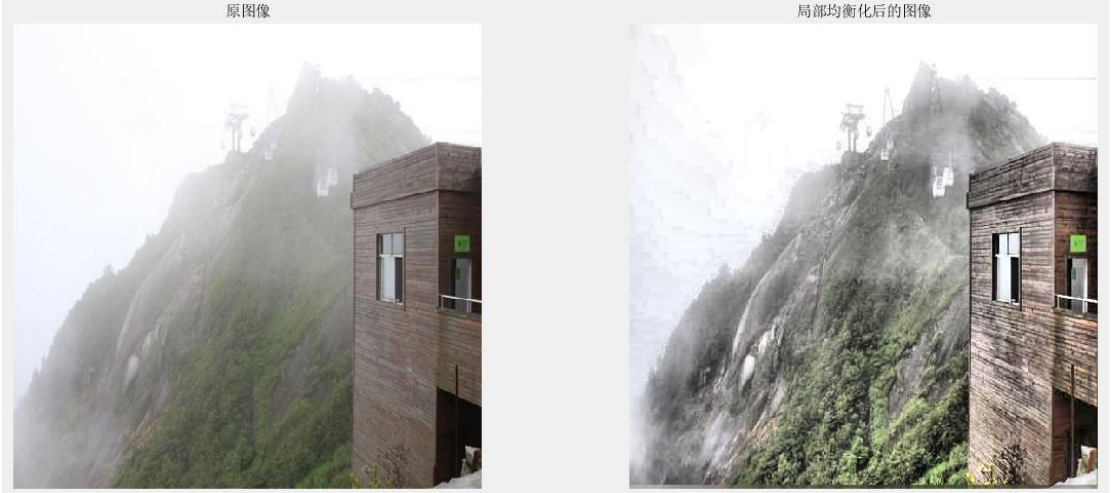


图 18

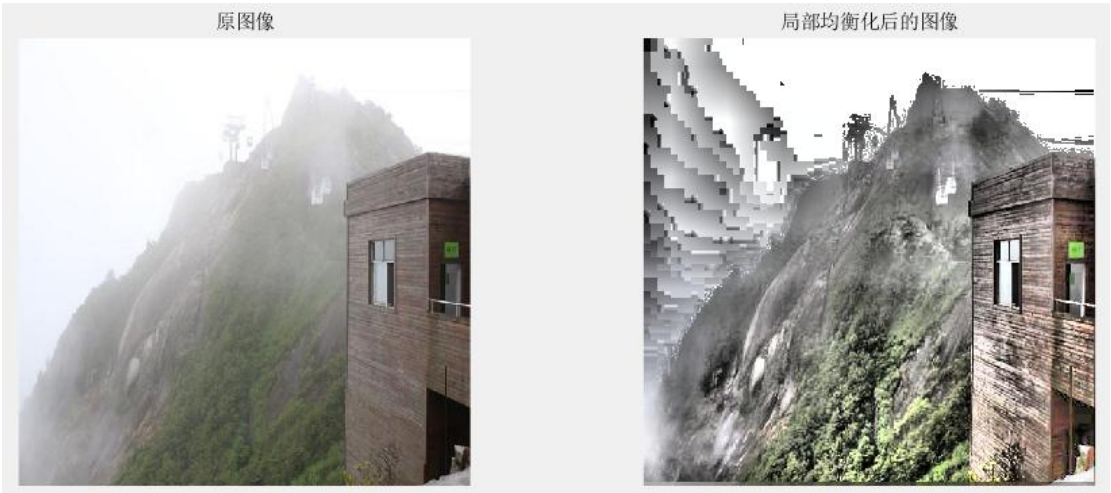


图 14（副）

块数目=3\*3



图 19

块数目=5\*5

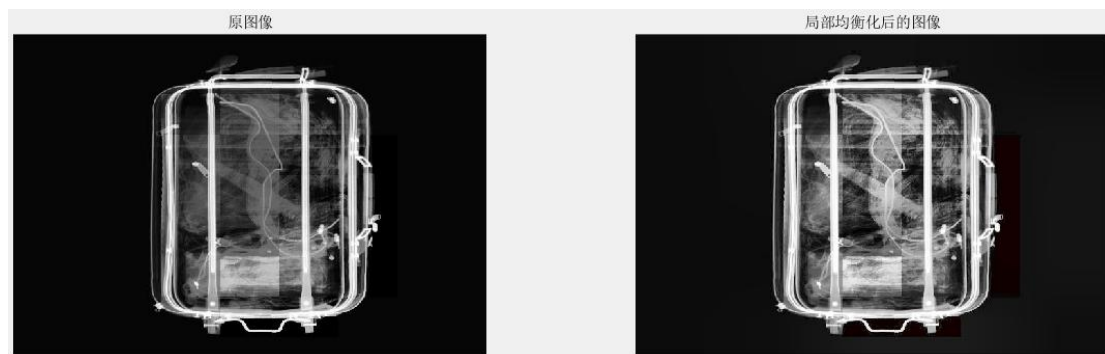


图 20

块数目=8\*8

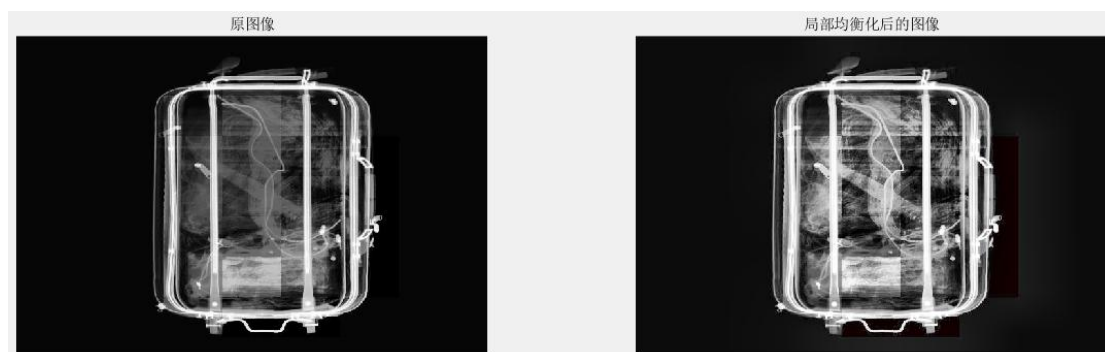


图 21

总结（优缺点）：

1. 通过对比图 18 和图 14（副）可以看出，图片左上方区域（平坦区，因为该区的灰度级相似）的雾通过限制对比度自适应局部均衡化后变得清晰了，这说明限制对比度自适应局部均衡化算法解决了自适应局部均衡化算法中的平坦区域的噪音被放大的问题。

2. 从图 19、图 20 和图 21 可以看出，自适应均衡化算法中的“当图像所分的块比较小的时候，对比度得到增强。当块的比较大的时候对比度将会降低”这问题解决了。增强图像对比度的效果更好了。

参考文献：数字图像处理-冈萨雷斯-第三版

<https://www.cnblogs.com/Ponys/archive/2013/05/11/3073325.html>

[https://blog.csdn.net/Aoman\\_Hao/article/details/102980987](https://blog.csdn.net/Aoman_Hao/article/details/102980987)

<https://cloud.tencent.com/developer/article/1503874>

<https://www.cnblogs.com/JiYF/p/9967796.html>

<https://www.cnblogs.com/luo-peng/p/4930013.html>

<https://www.cnblogs.com/wxl845235800/p/11122930.html>

[https://blog.csdn.net/schwein\\_van/article/details/84336633](https://blog.csdn.net/schwein_van/article/details/84336633)