

实验三 直方图均衡与分段灰度变换

一、基本原理与主要用途

1. 分段灰度变换

如果 $r' = T(r)$ 是一个线性的或分段线性的单值函数，则由它所确定的灰度变换称为灰度的线性变换。

a) 线性变换

设图像的灰度值 $r = f(x, y)$ 的可能取值区间为 $[r_{min}, r_{max}]$ 。但是由于连续图像 $f(x, y)$ 的值得动态范围过小，或者黑白照片曝光不足，会使得取值范围大为缩小，成为 $[r_a, r_b]$ ，其中 $r_{min} \leq r_a < r_b \leq r_{max}$ ，远未占满灰度可能的取值范围 $r_{min} \sim r_{max}$ ，我们希望把 $[r_a, r_b]$ 拉大。

现在采用这样一种线性变换：

$$\begin{cases} r' = \frac{r'_b - r'_a}{r_b - r_a}(r - r_a) + r'_a, r \in [r_a, r_b] \\ r' = r'_a, r \in [r_{min}, r_a] \\ r' = r'_b, r \in [r_b, r_{max}] \end{cases}$$

则可以把变换后的灰度 r' 的取值范围扩大到 $[r'_a, r'_b]$ 。经过这种变换后，直方图展宽了，反差拉开了，达到了灰度增强的效果。这种灰度增强方法称灰度线性变换。

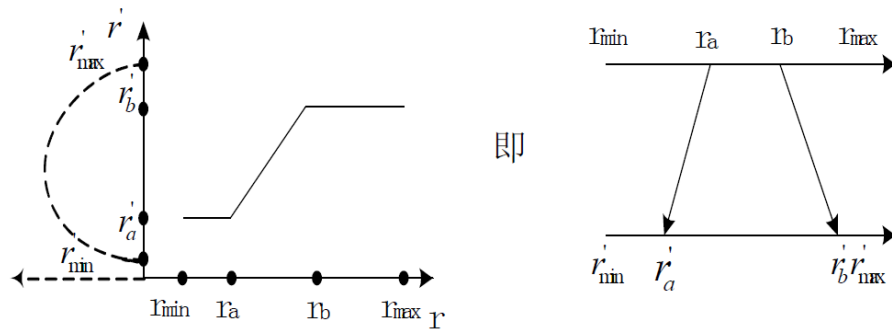


图1 灰度线性变换

b) 分段线性变换

该图像把 $[r_a, r_b]$ 分成两段 $[r_a, r_c]$ 、 $[r_c, r_b]$ 进行线性变换，就是分段线性变换，有时也将 $[r_a, r_b]$ 分成多段进行线性变换。采用分段线性变换可以压缩一部分灰度区间，扩展另一部分灰度区间。其变换表达式为：

$$\begin{cases} r' = \frac{r'_c - r'_a}{r_c - r_a}(r - r_a) + r'_a, r \in [r_a, r_c] \\ r' = \frac{r'_b - r'_c}{r_b - r_c}(r - r_c) + r'_c, r \in [r_c, r_b] \end{cases}$$

变换曲线如下图所示。图中扩展了第一区间 $[r_a, r_c]$ 而压缩第二区间 $[r_c, r_b]$ 。用多段折线构成一个单值函数，可以逼近一条曲线，完成较为精细的灰度变换。

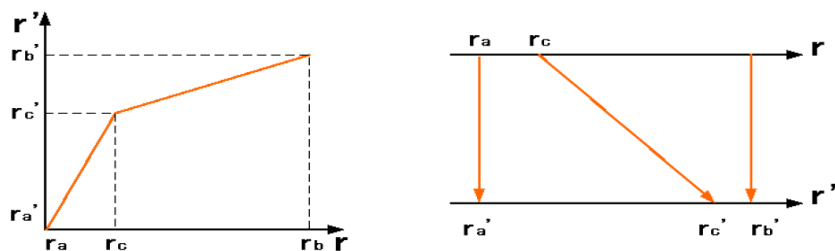


图2 分段线性变换

2. 直方图均衡

当要求灰度变换后的直方图 $p(r)$ 为一常数时，这种非线性变换称为直方图均衡。该方法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，直方图均衡通过有效地扩展常用的亮度来实现这种功能。

a) 灰度连续变化情况的直方图均衡

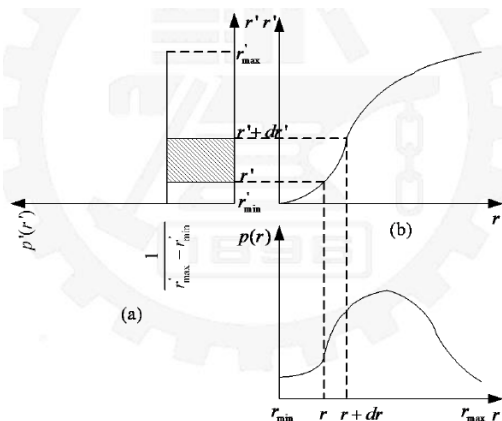
直方图均衡就是找出一种灰度变换来使变换后的图像直方图是“均匀的”，即 $p(r)$ 或 $p(k)$ 为一常数。做直方图变换的理论依据是最大熵原理。

在灰度连续变换情况下，定义熵

$$H_c = - \int_{r_{\min}}^{r_{\max}} p(r) \log p(r) dr$$

可以证明，当 $p(r)$ 为常数时，即直方图均衡时，熵最大，即可使视觉系统获取最大信息量。

这里使用概率论的常用方法：由密度函数 $p(r)$ 求分布函数 $F(r)$ 。定义分布函数为灰度 $\leq r$ 时的概率，即 $F(r) = \int_{r_{\min}}^r p(w) dw$ 。可以证明，当 $r' = T(r) = F(r)$ 时，即当灰度变换规律等于分布函数 $F(r)$ 时，就实现了直方图均衡。



b) 灰度离散化情况的直方图均衡和直方图规定化

对于离散的灰度级，一般来说，处理后的图像直方图不再均匀，这源于变量的离散属性。

令 $p_r(r_j), j = 1, 2, 3, \dots, L$ 表示给定图像灰度相关的直方图，且直方图的值大致是图像取各灰度级的概率。对于离散的灰度级，我们使用求和的方式，均衡化变换变为

$$s_k = T(r_k) = \sum_{j=1}^k p_r(r_j) = \sum_{j=1}^k \frac{n_j}{n}$$

其中 s_k 为输出灰度级。

二、实现方案

1. 软件平台：MATLAB R2016a

2. 主要算法及主流程图

a) 分段灰度变换

根据之前介绍的实验原理，分段灰度变换实现算法如下：首先读入灰度图文件，然后选择图二中 $[r_c, r_c]$ 点，构造灰度变换曲线。遍历图像的每一个像素的灰度值，根据之前确定的灰度变换曲线计算出该像素点的新灰度值，最后输出结果并保存图像。

主流程图如图 3 所示。

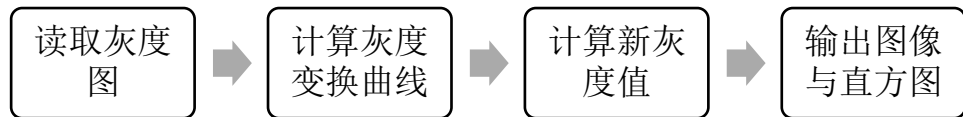


图 3 分段灰度变换主流程图

b) 直方图均衡

根据之前介绍的实验原理，直方图均衡实现算法如下：首先读入灰度图文件，然后遍历每一个像素点计算灰度值出现的频率。根据该频率计算各灰度密度函数，并由此求出灰度分布函数。根据分布函数确定原灰度与新灰度值得映射，计算新灰度值，最后输出结果并保存图像。

主流程图如图 4 所示。

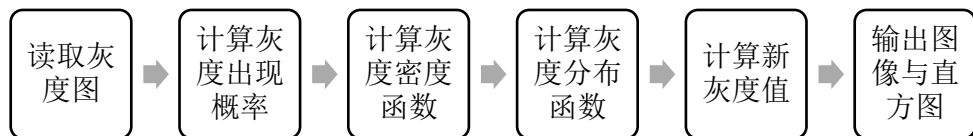


图 4 直方图均衡主流程图

三、结果分析

我们首先展示原图及其灰度直方图。显然该图像整体偏亮，灰度值高的部分占比高而灰度值低的部分较少。由此我们进行接下来的分析。

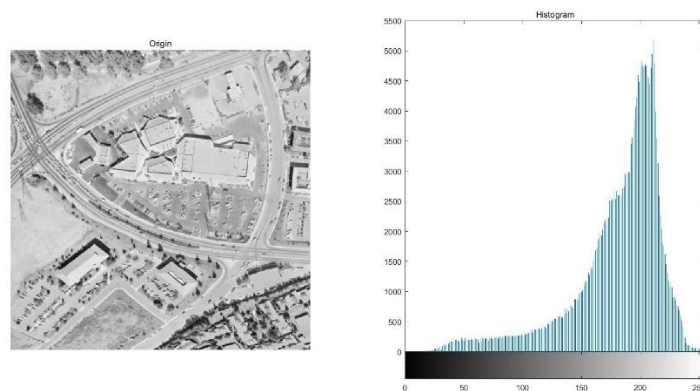


图 5 原始图像与其灰度直方图

1. 分段灰度变换

由于原图灰度值集中在较高的灰度值区间内，我们选取(0,150)作为第一区间，(150,255)作为第二区间，压缩第一区间而扩展第二区间，以达到增加较暗灰度级的目的。所得的分段线性变换曲线如下：

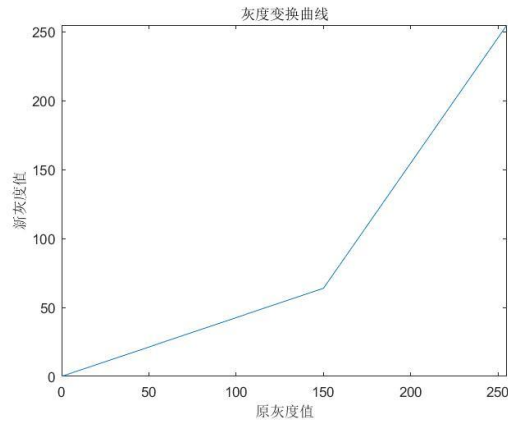


图 6 灰度变换曲线

将该映射作用于原图，我们得到结果图像及灰度直方图如下：

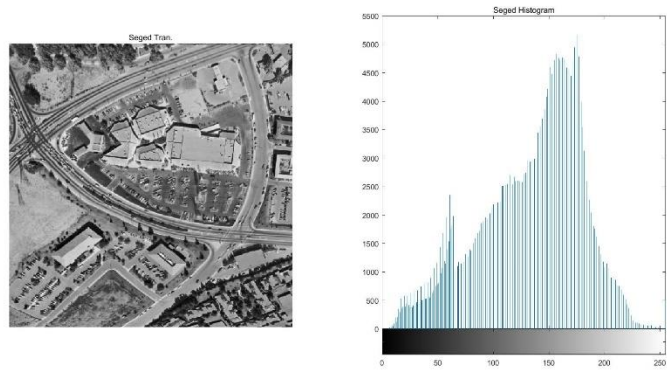


图 7 分段灰度变换结果及其直方图

原图亮区灰度饱和而难以看清。此时对暗区压缩灰度范围，亮区拉大反差，图片在较暗部分的细节得到显示，灰度直方图不再大量集中在高灰度值的部分，灰度分布相对更为平均。但是结果中出现了明显的灰度值合并情况，导致了一定的信息损失。

我们也尝试使用了二次曲线对原始图像做非线性灰度变换，灰度变换曲线及结果如下。

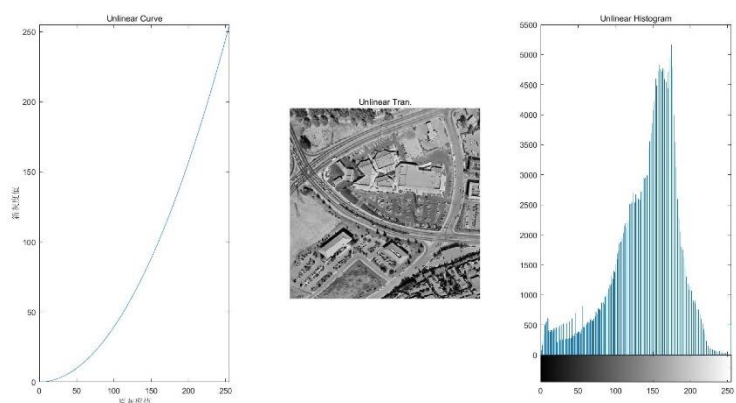


图 8 非线性灰度变换结果

其同样将高灰度值向低灰度值转换，但是相对线性灰度变换其结果更为平滑，更自然。

2. 直方图均衡

所得直方图分布函数如下：

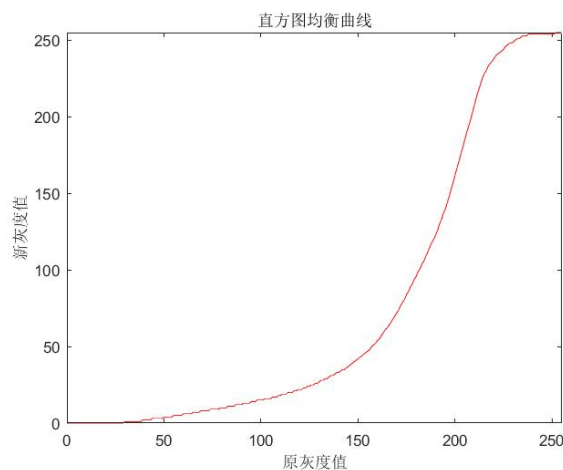


图 9 直方图均衡曲线

将该曲线作用于原图，得到的结果图像及其直方图如图 10 所示：

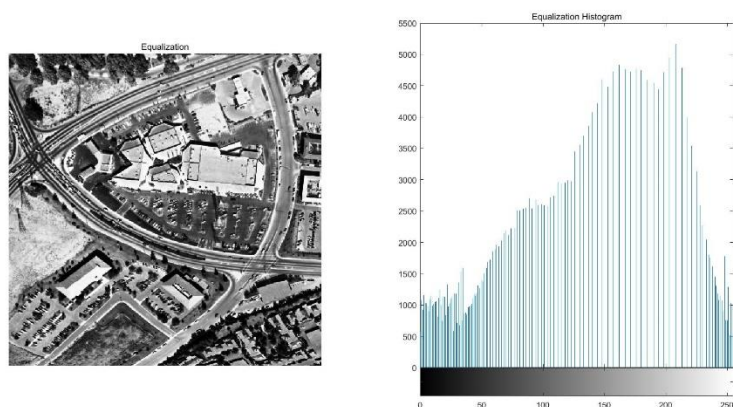


图 10 直方图均衡结果及其直方图

显然结果图像对比度明显升高，原图中模糊的一些线条变得清晰可辨。但是图像整体给人一种较为生硬的感受。灰度直方图相比原图变得较为均衡。

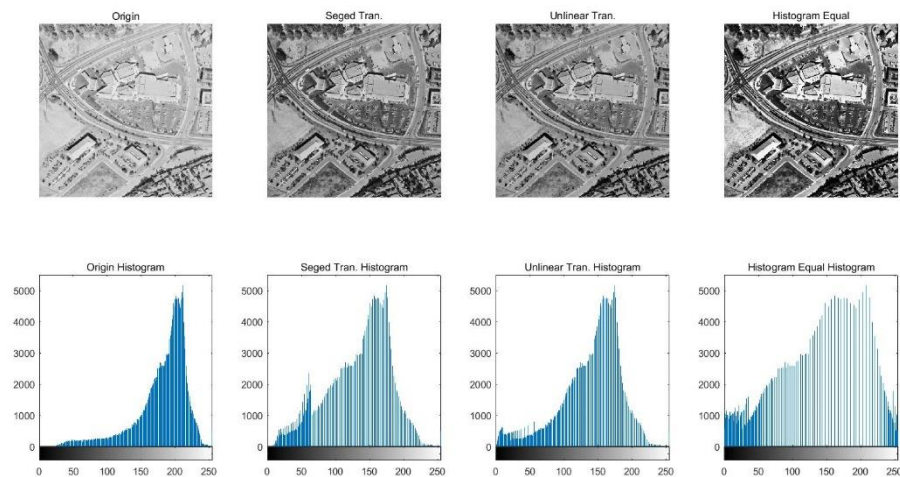


图 11 原图及灰度变换后结果

四、学习体会

本次作业相比前两次较为顺利，根据课本及课件的内容可以方便的完成灰度变换曲线的构造。图像增强相比图像压缩编码显得更为有趣，压缩总是在损失信息，而图像增强尽管有部分信息损失，一些主观评价反而有升高，处理得当会很有成就感。希望在将来的课程中能够学到更多有意思的东西。

五、源码

```
clear;
I = imread('aerial.bmp');
gray = I;
equal = I;
[height, width] = size(gray);
histogram = zeros(1,256);
for i = 1:height
    for j = 1:width
        histogram(gray(i,j)+1) = histogram(gray(i,j)+1)+1;
    end
end

subplot(2,4,1);
imshow(gray);
title('Origin');

subplot(2,4,5);
imhist(gray);
axis([0,255,0,5500]);
title('Origin Histogram');
```

```

x1 = 0;
y1 = 0;
x2 = 150;
y2 = 64;
x3 = 255;
y3 = 255;

a = polyfit([x1,x2],[y1,y2],1);
b = polyfit([x2,x3],[y2,y3],1);
result1 = double(gray);
for i=1:height
    for j = 1:width
        if(gray(i,j) < x2)
            result1(i,j)=a(1)*result1(i,j)+a(2);
        else
            result1(i,j)=b(1)*result1(i,j)+b(2);
        end
    end
end
result1 = uint8(result1);

subplot(2,4,2);
imshow(result1);
title('Seged Tran. ');
imwrite(result1, 'Seged.bmp');

subplot(2,4,6);
imhist(result1);
axis([0,255,0,5500]);
title('Seged Tran. Histogram');

gray = double(gray);
result = gray.*gray./255;
result = uint8(result);
subplot(2,4,3);
imshow(result);
title('Unlinear Tran. ');
imwrite(result, 'Unlinear.bmp');

subplot(2,4,7);
imhist(result);
axis([0,255,0,5500]);
title('Unlinear Tran. Histogram');

```

```

    histogram = histogram./(height*width);

    cdf = zeros(1,256);
    cdf(1) = histogram(1);
    for i=2:256
        cdf(i) = cdf(i-1) + histogram(i);
    end

    cdf = uint8(255.*cdf);
    for i = 1:height
        for j = 1:width
            equal(i, j) = cdf(equal(i, j));
        end
    end

    subplot(2,4,4);
    imshow(equal);
    title('Histogram Equal');
    imwrite(equal, 'Equal.bmp');

    subplot(2,4,8);
    imhist(equal);
    axis([0,255,0,5500]);
    title('Histogram Equal Histogram');

```