数字图像处理 project

Project title:	Histogram Equalization	Project number:	Project 0302
姓　　名:	彭思达	学　　号:	3140103545
学　　院:	信电学院	专　　业:	信息工程
Date due:	2017.3.27	Date handed in:	March 27, 2017

Abstract

这篇报告讨论了直方图均衡连续和离散的情况。在连续可微的情况下，本文证明了通过直方图均衡的变换函数，输出图像的概率密度函数将是一个常数。随后还给出了离散情况下直方图均衡的变换函数。

为了感受直方图均衡变换公式的效果，本文用代码实现了变换函数，并以书上 fig 3.8(a) 为实验对象，得到它的直方图、相应输出图像及其直方图。在文章的最后附上了相应的实现代码。

March 27, 2017

1 Technical discussion

1.1 限制条件

首先我们考虑连续灰度值，使用变量 r 表示待处理图像的灰度。假设 r 的取值区间为 [0, L-1]。

对于相应的灰度变换公式，我们有三个限制条件：

1. $T(r)$ 在区间 $0 \leq T(r) \leq L-1$ 上为单调递增函数。
2. 当 $0 \leq r \leq L-1$ 时，$0 \leq T(r) \leq L-1$。
3. 当存在反函数时，$T(r)$ 在区间 $0 \leq T(r) \leq L-1$ 上为严格单调递增函数。

对于上述 3 个限制条件，原因分别如下：

1. 第一个限制是为了保证输出灰度值不少于相应的输入值，防止灰度反变换时产生人为缺陷。
2. 第二个限制是为了保证输出灰度的范围与输入灰度的范围相同。
3. 第三个限制是为了保证从 s 到 r 的反映射是一对一的，防止出现二义性。

1.2 连续情况下的直方图均衡

在连续情况下，直方图均衡的变换函数如下所示：

$$s = T(r) = (L-1)\int_0^r p_r(w)dw \tag{1}$$

其中，$p_r(r)$ 表示随机变量 r 的概率密度函数。

接下来，我们来证明通过上述变换公式，得到的输出图像的概率密度函数是一个常数。

首先理清 $p_s(s)$ 和 $p_r(r)$ 的关系，其中 $p_s(s)$ 和 $p_r(r)$ 分别是随机变量 s 和 r 的概率密度函数：

$$p_s(s) = \frac{F_s(s)}{|ds|} = \frac{F_r(r)}{|dr|}|\frac{dr}{ds}| = p_r(r)|\frac{dr}{ds}| \tag{2}$$

根据这个关系式以及变换函数可以推导出：

$$\begin{aligned} p_s(s) &= p_r(r)\left|\frac{dr}{ds}\right| = p_r(r)\frac{1}{\left|\frac{ds}{dr}\right|} \\ &= p_r(r)\frac{1}{\left|(L-1)p_r(r)\right|} \\ &= \frac{1}{L-1} \end{aligned} \tag{3}$$

1.3　离散情况下的直方图均衡

对于离散值，本文使用其概率与求和来代替处理概率密度函数与积分。

一幅数字图像中灰度级 r_k 出现的概率近似为：

$$p_r(r_k) = \frac{n_k}{MN}, k = 0, 1, 2, ..., L-1 \tag{4}$$

其中，MN 是图像中像素的总数，n_k 是灰度为 r_k 的像素个数，L 是图像中可能的灰度级的数量。

在离散情况下，直方图均衡的变换公式如下：

$$s_k = T(r_k) = (L-1)\sum_{j=0}^{k} p_r(r_j) = \frac{L-1}{MN}\sum_{j=0}^{k} n_j, k = 0, 1, 2, ..., L-1 \tag{5}$$

2　Discussion of results

对于连续情况下，直方图均衡的变换公式为：

$$s = T(r) = (L-1)\int_0^r p_r(w)dw \tag{6}$$

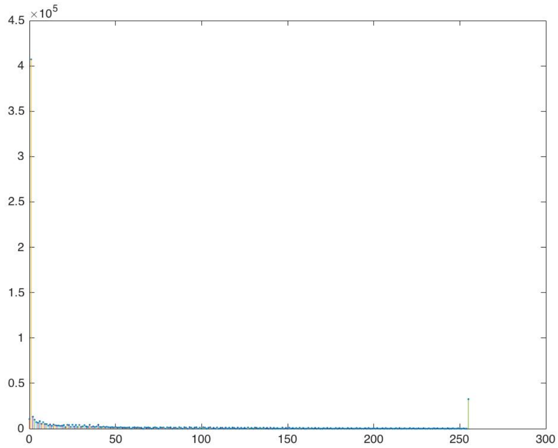
对于离散情况下，直方图均衡的变换公式为：

$$s_k = T(r_k) = (L-1)\sum_{j=0}^{k} p_r(r_j) = \frac{L-1}{MN}\sum_{j=0}^{k} n_j, k = 0, 1, 2, ..., L-1 \tag{7}$$

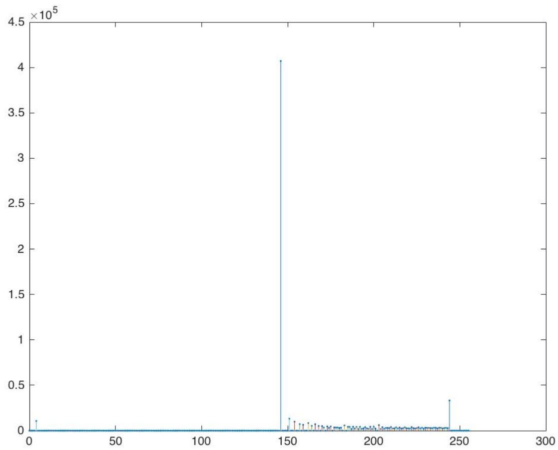
在连续情况下，这个变换公式可以实现直方图真正的均衡。在离散情况下，因为我们用概率与求和近似地取代处理概率密度函数与积分，所以无法实现直方图真正的均衡。

3 Results

首先，我们的待处理图像和它的直方图如下：



通过离散情况的直方图均衡变换公式后，得到的输出图像和它的直方图如下：



4 Appendix

直方图均衡的实现代码：

```
% 图像文件名称
filename='Fig0308_a.tif';
% 图像灰度级
L = 256;
% 读取图像
img = imread(filename);
% 显示原图
% imshow(img)
[M, N] = size(img);
n = zeros(1, L);
s = zeros(M, N);
% 获得原图中各像素值的个数
for index = 0:L-1
    temp = find(img == index);
    n(index+1) = length(temp);
end
% 显示原图的直方图
figure
plot([0:L-1], n, '.');
hold on;
for index = 1:L
    plot([index-1,index-1],[0,n(index)]);
end
% 根据公式进行直方图均衡，得到目标图像
for index_row = 1:M
    for index_col = 1:N
        value = img(index_row, index_col);
        s(index_row,index_col) = sum(n(1:value+1))*(L-1)/(M*N);
    end
end
s = uint8(s);
% 获得输出图像中各像素值的个数
for index = 0:L-1
    temp = find(s == index);
    n(index+1) = length(temp);
end
% 显示输出图像的直方图
figure
plot([0:L-1], n, '.');
hold on;
for index = 1:L
    plot([index-1,index-1],[0,n(index)]);
end
figure
% 显示输出图像
% imshow(s);
% 保存输出图像
imwrite(s,'result.tif');
```