

## 数字图像处理hm1

### 一、直方图均衡

算法描述

具体实现

实验效果

### 二、直方图规定化（匹配）

算法描述

具体实现

实验效果

# 数字图像处理hm1

## 一、直方图均衡

实现river.jpg 图像的直方图均衡，不能直接使用Matlab的histeq()函数。将有关均衡图像和调用histeq()函数的结果作比较。

### 算法描述

直方图均衡化属于图像增强的一种，通过在空间域上对图像的像素值进行非几何变换，可以对图像的对比度进行拉伸调整，增强局部的对比度而不影响整体的对比度，有效地扩展常用的亮度。对于背景和前景都太亮或者太暗的图像非常有用，这种方法可以将更好显示X光图像中的骨骼结构以及曝光过度或者曝光不足照片中的细节。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。

由于给的river.jpg为离散的灰度图像，让  $n_i$  表示灰度  $i$  出现的次数， $L$  是图像中所有的灰度数（通常为256）， $n$  是图像中所有的像素数，这样图像中灰度为  $i$  的像素的概率密度函数是

$$p_x(i) = p(x = i) = \frac{n_i}{n}, \quad 0 \leq i < L$$

对应于  $p_x$  的累积分布函数为：

$$cdf_x(i) = \sum_{j=0}^i p_x(j),$$

然后根据一个  $s = T(r)$  的变换，将原图像的灰度值映射，得到的图像的直方图就近似满足均衡分布，之所以是近似，因为灰度值的类型在matlab中为uint8，必须将线性变换后的灰度值四舍五入到整数才可以保存。

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

此变换函数满足条件：

- (a)  $T(r)$  在区间  $0 \leq r \leq 1$  中为单值且单调递增
- (b) 当  $0 \leq r \leq 1$  时,  $0 \leq T(r) \leq 1$

## 具体实现

```
% pro1/src/pro1.m

clear all;
% 读取图片到矩阵，矩阵元素为灰度值，下标为图片的几何坐标，注意从下标1开始，而坐标从0开始
f = imread(' ../river.JPG');
[height,width] = size(f)
% 查看图片信息
% whos f
% imfinfo ../river.JPG

% 直方图均衡化算法
g = my_histeq(f,256);

% 用histeq接口得到的图像作为测试
h = histeq(f,256);

%显示原图像、均衡化后，以及histeq返回的图像
%显示原图像、均衡化后，以及histeq返回的直方图
figure;
subplot(2,3,1),imshow(f);title('origin image');
subplot(2,3,2),imshow(g);title('hist equal image');
subplot(2,3,3),imshow(h);title('histeq test image');
subplot(2,3,4),imhist(f);title('origin hist');
subplot(2,3,5),imhist(g);title('hist equal hist');
```

```
subplot(2,3,6),imhist(h);title('histeq test image');
```

```
% pro1/src/my_histeq.m

function f_equal = my_histeq(f,L)
    [height,width] = size(f);
    % (1, L) 灰度级为L, 返回每个灰度值 (0~L-1) 的个数
    hr = imhist(f,L);
    % 归一化
    pr = hr / numel(f);
    % (L-1) *cdf, 线性变换函数
    s = round((L-1)*cumsum(pr));
    % 映射
    f_equal = f;
    for i=1:height
        for j=1:width
            f_equal(i,j) = s(f(i,j));
        end
    end
    % 写回文件
    imwrite(f_equal,'../result/myhisteq_img.jpg')
end
```

## histeq

Enhance contrast using histogram equalization

collap

### Syntax

```
J = histeq(I,hgram)
J = histeq(I,n)
[J,T] = histeq(I)

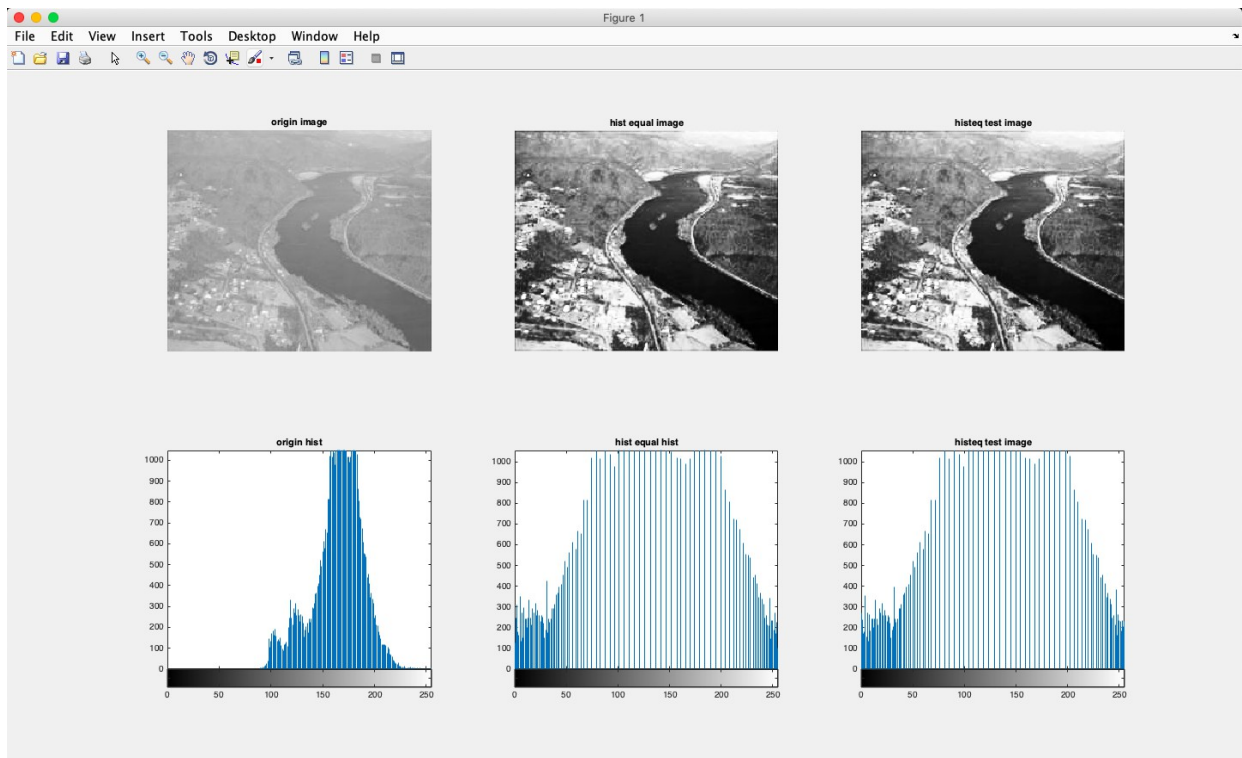
newmap = histeq(X,map)
newmap = histeq(X,map,hgram)
[newmap,T] = histeq(X, __ )
```

### Description

**J** = **histeq(I,hgram)** transforms the grayscale image **I** so that the histogram of the output grayscale image **J** with **length(hgram)** bins approximately matches the target histogram **hgram**.

You optionally can perform histogram equalization of grayscale images using a GPU (requires Parallel Computing Toolbox™). For more information, see [Image Processing on a GPU](#).

## 实验效果



第一列为原始图像及直方图，第二列为实现均衡化后的图像及直方图，最后一列为使用histeq后测试得到的图像和直方图。

## 二、直方图规定化（匹配）

将图像 EightAM.png 的直方图匹配为图像 LENA.png 的直方图，显示 EightAM.png 在直方图匹配前后的图像，并绘制 LENA.png 的直方图、直方图匹配前后 EightAM.png 的直方图，检查直方图匹配的效果。

### 算法描述

直方图匹配又称为直方图规定化，是指将一幅图像的直方图变成规定形状的直方图而进行的图像增强方法。即将某幅影像或某一区域的直方图匹配到另一幅影像上，使两幅影像的色调保持一致。

由于两张给定图片还是为离散的灰度图像，假设 $r$ 为输入图像EightAM.png的灰度值， $z$ 为输出得到的灰度值， $s$ 为输入图像经过直方图均衡化后的灰度值。一般情况的灰度变换，只需要一次映射： $r \rightarrow s$ ，例如直方图均衡化。而直方图匹配，需要两次映射， $r \rightarrow s \rightarrow z$ 。

首先将输入图像进行直方图均衡化得到对应的 $s$ 值。

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

然后将匹配直方图的图像同样均衡化得到对应的 $s$ 值。

$$G(z) = \int_0^z p_z(t) dt = s$$

因为离散的 $s$ 变量的概率密度函数近似为均匀分布，也就是说 $s$ 与 $r$ 的概率密度函数无关。同样对于 $z$ 经过直方图均衡化后的 $s$ 变量与 $z$ 的概率密度函数也无关。所以 $s$ 可以作为中间变量连接输入和输出灰度值。

又由于要输出的 $z$ 值不确定，但已经规定好了 $z$ 变量服从的概率密度函数，即由lena图片的直方图决定，所以可以将 $s$ 反变换得到想要的 $z$ 值。

1. 计算给定图像的直方图  $p_r(r)$ ，并用它寻找式(3.3-13)的直方图均衡变换。把  $s_k$  四舍五入为范围  $[0, L-1]$  内的整数。
2. 用式(3.3-14)对  $q = 0, 1, 2, \dots, L-1$  计算变换函数  $G$  的所有值，其中  $p_z(z_i)$  是规定的直方图的值。把  $G$  的值四舍五入为范围  $[0, L-1]$  内的整数。将  $G$  的值存储在一个表中。
3. 对每一个值  $s_k, k = 0, 1, 2, \dots, L-1$ ，使用步骤 2 存储的  $G$  值寻找相应的  $z_q$  值，以使  $G(z_q)$  最接近  $s_k$ ，并存储这些从  $s$  到  $z$  的映射。当满足给定  $s_k$  的  $z_q$  值多于一个时(即映射不唯一时)，按惯例选择最小的值。
4. 首先对输入图像进行均衡，然后使用步骤 3 找到的映射把该图像中的每个均衡后的像素值  $s_k$  映射为直方图规定化后的图像中的相应  $z_q$  值，形成直方图规定化后的图像。正如连续情况那样，均衡输入图像的中间步骤是概念上的。它可以用合并两个变换函数  $T$  和  $G^{-1}$  跳过这一步，如例 3.8 所示。

## 具体实现

```
% pro2/src/pro2.m

clear all;

lena_img = imread('../LENA.png');
eight_img = imread('../EightAM.png');

% 直方图匹配算法
g = my_histmatch(eight_img, lena_img);

% 用imhistmatch接口得到的图片作为测试
t = imhistmatch(eight_img, lena_img);
% 或者使用histeq(eight_img, imhist(lena_img))结果相同

%从左到右显示原图像、匹配图像，匹配后的以及imhistmatch返回的图像
%从左到右显示原图像、匹配图像，匹配后的以及imhistmatch返回的直方图
figure;
subplot(2,4,1),imshow(eight_img);title('origin image');
subplot(2,4,2),imshow(lena_img);title('reference image');
subplot(2,4,3),imshow(g);title('my match image');
subplot(2,4,4),imshow(t);title('test match image');
subplot(2,4,5),imhist(eight_img);title('origin hist');
subplot(2,4,6),imhist(lena_img);title('reference hist');
subplot(2,4,7),imhist(g);title('my match image');
subplot(2,4,8),imhist(t);title('test match image');
```

```
% pro2/src/my_histmatch.m

function h = my_histmatch(f,g)
    [height_f,width_f] = size(f)
    [height_g,width_g] = size(g)

    hist_f = imhist(f);
    hist_g = imhist(g);

    cdf_f = cumsum(hist_f) / numel(f);
    cdf_g = cumsum(hist_g) / numel(g);

    % 建立r->z映射
    M = zeros(1,256);
    for i = 1 : 256
        % 对每一个灰度值，寻找与通过G(z)得到的s值差最小的那一项，其下标减少1即为对应的z灰度值。
        [tmp,ind] = min(abs(cdf_f(i) - cdf_g));
        % 将坐标减1变为灰度值
        M(i) = ind-1;
    end
    % 新图进行灰度变换
```

```

h = f;
for i=1:height_f
    for j=1:width_f
        h(i,j) = M(f(i,j)+1);
    end
end
imwrite(h,'../result/my_histmatch.jpg')
end

```

## imhistmatch

Adjust histogram of 2-D image to match histogram of reference image

R2018b

[collapse all in page](#)

### Syntax

```

J = imhistmatch(I,ref)
J = imhistmatch(I,ref,nbins)
J = imhistmatch(__,Name,Value)
[J,hgram] = imhistmatch(__)

```

### Description

**J** = **imhistmatch(I,ref)** transforms the 2-D grayscale or truecolor image **I** returning output image **J** whose histogram approximately matches the histogram of the reference image **ref**.

[example](#)

- If both **I** and **ref** are truecolor RGB images, then **imhistmatch** matches each color channel of **I** independently to the corresponding color channel of **ref**.
- If **I** is a truecolor RGB image and **ref** is a grayscale image, then **imhistmatch** matches each channel of **I** against the single histogram derived from **ref**.
- If **I** is a grayscale image, then **ref** must also be a grayscale image.

Images **I** and **ref** can be any of the permissible data types and need not be equal in size.

## 实验效果

根据算法得到的



使用imhistmatch或者histeq得到的



对比

