

实验报告

16340282 袁之浩

直方图均衡

图像直方图用以表示数字图像中亮度分布的直方图,标绘了图像中每个亮度值的像素数。

直方图均衡就是将直方图均衡的分布在整个像素空间中,可以提高图像的对比度和视觉效果。

首先统计图像中每个像素的频数和频率

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

```
%进行像素灰度统计;
grayPixel = zeros(256,1);    %统计各灰度数目,共256个灰度级
for i = 1:height
    for j = 1: width
        grayPixel(I(i,j) + 1) = grayPixel(I(i,j) + 1) + 1;    %对应灰度值像素点数量增加一
    end
end

%计算灰度分布密度
ProbPixel = zeros(256,1);
for i = 1:256
    ProbPixel(i) = grayPixel(i) / (height * width);
end
```

然后计算累积分布函数

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

```

%计算累计直方图分布
cdfPixel = zeros(256,1);
for i = 1:256
    if i == 1
        cdfPixel(i) = ProbPixel(i);
    else
        cdfPixel(i) = cdfPixel(i - 1) + ProbPixel(i);
    end
end

```

这里可以使用 matlab 的函数 `imhist` 统计频率，`cumsum` 函数计算累积函数值。

```

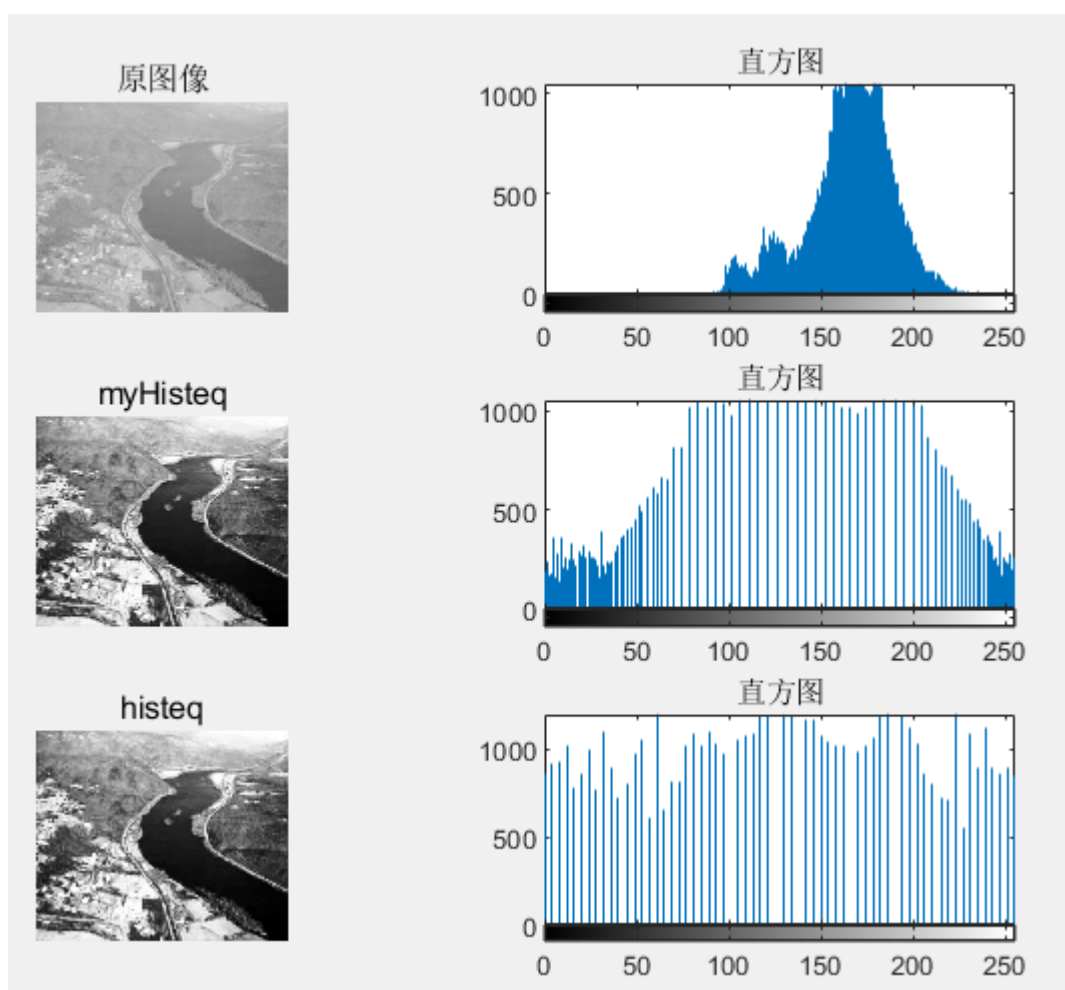
%简洁写法
%hist=imhist(I);
%cdfPixel=cumsum(hist) / numel(I);

```

由于像素都是整数值，使用 `round` 函数将小数四舍五入，并拓展到 0-255，最后根据原图像亮度值赋值新的亮度。

我把上述步骤写成了一个函数 `myHisteq`，然后在 `main` 中调用。

对比效果，可以看到图片效果差不多，但是自己写的函数在两端的均衡还是不如标准函数。



直方图匹配

直方图匹配是给定标准图像，将原图像的直方图匹配成标准图像的直方图，弥补直方图均衡化不具备交互作用的特性。

有了上一步的经验，直接使用简洁写法获得两幅图片的 cdf，P1 和 P2

```
im=imread('EightAM.png');
hist=imhist(im);
cdf=cumsum(hist)/numel(im);

im1=imread('LENA.png');
hist1=imhist(im1);
cdf1=cumsum(hist1)/numel(im1);
```

然后对 P1 中的每一项，寻找 P2 中与它函数值最接近的项做映射，最后把原图中每个

像素的亮度值赋值为他映射的 P2 对应的亮度值。

```
%匹配
M = zeros(256,1);
for i=1:256
    [~,index] = min(abs(cdf(i)-cdf1));
    M(i)=index;
end

result = uint8(M(im)-1);
```

这里需要注意的是 matlab 的数组下标是从 1 开始，所以转换到灰度值时需要-1。

最后结果，可以看到与 histeq 函数相比，基本相同，只有对 255 灰度值的处理不一样。

