

Digital Image Processing Hw1

Task 1: Histogram equalization

Request

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。

Description

直方图均衡化的直观感受是在视觉上增大了图像的对比度，它在不改变某个灰度级像素个数的前提下，将图像的像素分布通过一种方法映射到另外一种分布上去，在该映射过程中主要使用了累积分布函数。累积分布函数用于描述随机变量的概率分布，该函数是递增的，并且值域分布范围是0-1，在本例中，图像的像素值分布在0-255，下面为本例中实现直方图均衡的主要步骤：

1. 读取原图像，统计不同灰度级像素的数量，并将其存储在一个向量中：

```
tmp = zeros(1, 256);
for i = 1:row
    for j = 1:column
        tmp(img(i, j) + 1) = tmp(img(i, j) + 1) + 1;
    end
end
```

2. 计算该向量中不同灰度级出现的概率：

```
for i = 1:256
    tmp(i) = tmp(i) / (row * column * 1.0);
end
```

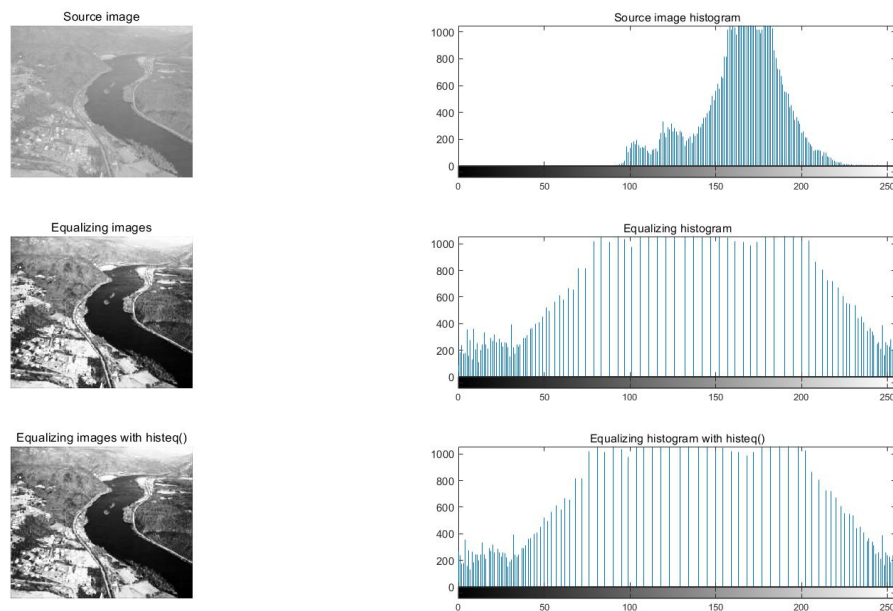
3. 计算不同灰度级直方图均衡变换函数，即累积分布函数的值，并将该值进行取整：

```
sum = zeros(1, 256);
sum(1) = tmp(1);
for i = 2:256
    sum(i) = sum(i - 1) + tmp(i);
end
%取整
sum = round(255 .* sum);
```

4. 对图像进行均衡化：

```
for i = 1:row
    for j = 1:column
        img(i, j) = sum(img(i, j) + 1);
    end
end
```

Result



由上图，我们观察到实现的直方图均衡化程序的执行结果与调用 Matlab 内置的 `histeq()` 函数的结果差异很小，程序执行良好。

Task 2: Histogram matching

Request

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。

Description

在某些应用中，采用直方图均衡化的图像增强方法并不是最好的方法，有的时候，我们更倾向于由一幅图像得到一幅其灰度级具有制定概率密度函数的图像，这种产生处理后有特殊直方图的方法就是直方图匹配。

一般而言，使用以下步骤可以得到具有直方图匹配的图像：

1. 根据直方图均衡化原理，对原始图的直方图进行灰度均衡化处理；
2. 按照目标图像的概率密度函数 $P_z(z)$ ，求解目标图像进行均衡化处理的变换函数 $G(z)$ ；
3. 用原始图像均衡化中得到的灰度级 s 代替 v ，求解逆变换 $z = G^{-1}(s)$ ；

在本例中，我们使用的算法步骤如下：

1. 读取标准图像 `LENA.png`，得到其累积直方图；
2. 读取原图 `EightAM.png`，得到其累积直方图；
3. 比较两个图象的每个灰度级，通过找到两个累积直方图距离最近的点来得到映射关系：

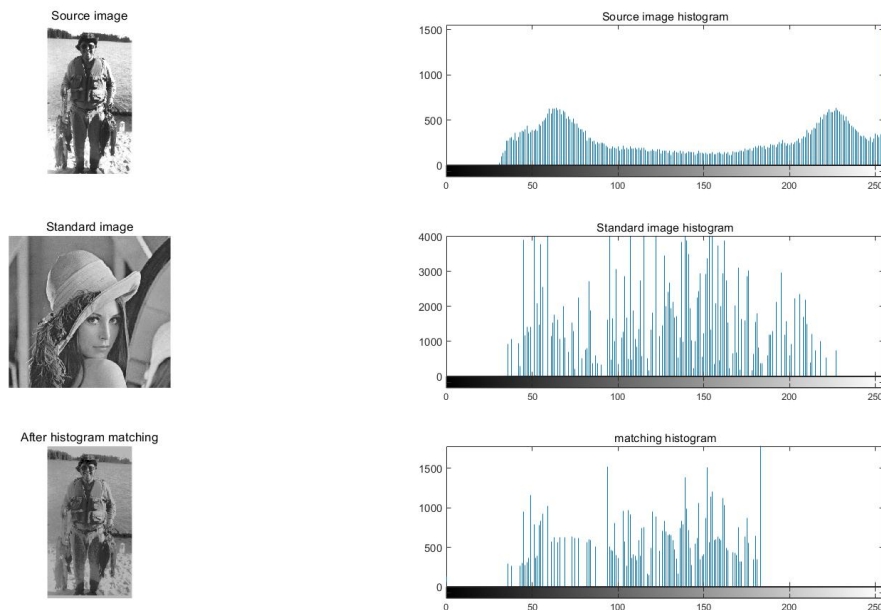
```

%计算概率和
for i = 1:256
    vecOrigin = [vecOrigin sum(histOrigin(1:i))];
    vecStd = [vecStd sum(histStd(1:i))];
end
%得到映射关系
for i = 1:256
    tmp{i} = vecStd - vecOrigin(i);
    tmp{i} = abs(tmp{i});
    [mid index(i)] = min(tmp{i});
end

```

4. 将原图灰度映射到新的灰度;

Result



由于原输入图像的部分灰度大于230左右，部分小于30的灰度接近于0，而标准图像的直方图中很少有像素灰度大于230左右，当把这种变换应用于原输入图像的灰度时，原输入图像的部分暗像素空间以及部分亮像素空间被映射到了输入图像灰度级的中部，对比度反而没有原图像效果好。