

作业一：直方图均衡化 实验报告

姓名：欧阳鸿荣

学号：161220096

邮箱：895254752@qq.com

手机：13055644369

1. 实现细节

1.1 综述

本次直方图均衡化实验我将其分为2个部分：

- 灰度图的直方图均衡化
- 彩色图的直方图均衡化

其中对于灰度图的直方图均衡化我实现了一种方法，彩色图的直方图均衡化我使用了3种方法。但是其核心终究都基于对于单通道的直方图均衡化（或者说对于任意矩阵的直方图均衡化）。

1.2 单通道（同灰度图）直方图均衡化算法

我使用的是CDF(*Cumulative Distribution Function*)累积分布函数法，来自于维基百科的[直方图均衡化](#)。

出于于计算方便和精度的保留，这里用累积分布直方图来等价代替累积分布函数（区别在于是否除以像素总数目归一化）。设 $cdf(v)$ 为原单通道图的累积分布直方图， cdf_{min} 为最小的数量非零的灰度值， M, N 分别为原图的长和宽， L 为灰度级别（一般为256），则对于灰度 v ，直方图均衡化后得到的结果 $h(v)$ 如下：

$$h(v) = round(\frac{cdf(v) - cdf_{min}}{(M \times N) - cdf_{min}} \times (L - 1))$$

通过这个方法，则可以把单通道图像做直方图均衡化。该方法在Matlab代码中主要通过如下步骤实现：

1. 将传入的单通道图矩阵精度扩充为 `double` 型，减少精度损失
2. 用 `imhist` 函数得到该矩阵的直方图，统计每个灰度值有几个像素
3. 借助 `find` 函数得到最小的数量非零的灰度值 cdf_{min}
4. 遍历所有的灰度等级，通过直方图得到累积分布直方图的值，并由上述公式得到变换后的灰度值
5. 通过用极小量 `eps` 代替零值，减少因为除零而产生的噪点。

对应代码如下：

```
function [output2] = hist_equal(input_channel)
% 直方图均衡化算法(input_chaneel是单通道的)
% 这里采用wiki上Histogram equalization的算法
```

```

% 0.观察到color.jpg 读入的值是uint8的矩阵, 为了保证处理的精度, 先将输入通道的精度提高到double
[input_channel] = im2double(input_channel);
% 1.对图像基础数据计算
[Height,Width] = size(input_channel); %图像的长宽
[Counts,Value] = imhist(input_channel); %Value为0-255的灰度值, Counts为灰度值对应的像素个数
% 2.找到CDF_Min
CDF_Map = find(Counts~=0); %指示非0值的下标
CDF_Min = Counts(min(CDF_Map)); %分布函数中最小的非零值
% 3 计算CDF从而直方图均衡化
[n,tmp] = size(CDF_Map);
[L,tmp] = size(Value);
display('L='+L);
CDF = 0;
new_Value = 0;
for i=1:n
    CDF = CDF + Counts(CDF_Map(i)); %累积分布直方图的值
    den = (Height*width-CDF_Min);
    den(den==0) = eps; %防止由于除零而产生的噪点
    new_Value = ((CDF-CDF_Min)*(L-1))/den; %原灰度值通过累积分布直方图均衡化后得到的新值
    %将原图中与原灰度值一样的像素点均衡化赋值
    input_channel(input_channel==Value(CDF_Map(i))) = new_Value;
end
output2 = input_channel;
end

```

1.3 彩色图像的直方图均衡化算法

相较于灰度图这样的单通道图像, 彩色图像有RGB三个通道。基于对哪些通道处理的问题, 我采用了三种方法:

(1) RGB法

这也是框架代码中给的方法, 对RGB通道分别进行直方图均衡化再合成。代码如下:

```

%this is a RGB image
%获取RGB三个通道的分量
r=input_image(:,:,1);
v=input_image(:,:,2);
b=input_image(:,:,3);
%对RGB通道分别进行直方图均衡化
r1 = hist_equal(r);
v1 = hist_equal(v);
b1 = hist_equal(b);
%合成
rgb = cat(3,r1,v1,b1);
output = uint8(rgb);

```

(2) HSV法

HSV模型是根据颜色的直观特性设计的一种颜色空间，分别表示色调(H)，饱和度(S)，明度(V)。而三个参数中与对比度相关最大的为明度，基于此，HSV法将RGB图像转为HSV图像，只对V通道直方图均衡化后再转为RGB图像，从而提高图像对比度。代码如下：

```
%this is a RGB image
%将RGB图像转为HSV图像
hsv = rgb2hsv(rgb);
h = hsv(:,:,1);
s = hsv(:,:,2);
v = hsv(:,:,3);
%此时的v取值在(0,1)，适应hist_equal的参数需求
v = uint8(round(v*255+0.5));
%对v通道单独进行直方图均衡化
v2 = hist_equal(v);
hsv = cat(3,h,s,v2);
%将HSV图像转为RGB图像
rgb = hsv2rgb(hsv);
output = uint8(rgb);
```

(3) HSI法

HSI是指一个数字图像的模型，它反映了人的视觉系统感知彩色的方式，以色调(H)、饱和度(S)和亮度(I)三种基本特征量来感知颜色。而三个参数中与对比度相关最大的为亮度，基于此，HSI法将RGB图像转为HSI图像，对I通道直方图均衡化后再转为RGB图像，从而提高图像对比度。代码如下：

```
%this is a RGB image
%将RGB图像转为HSI图像
hsi = rgb2hsi(rgb);
h = hsi(:,:,1);
s = hsi(:,:,2);
i = hsi(:,:,3);
%此时的i取值在(0,1)，适应hist_equal的参数需求
i = uint8(round(i*255+0.5));
%对i通道单独进行直方图均衡化
i2 = hist_equal(i);
i2 = i2/255;
i2 = double(i2);
hsi = cat(3,h,s,i2);
%将HSI图像转为RGB图像
rgb = hsi2rgb(hsi);
output = uint8(rgb);
```

matlab中有 `hsv2rgb`, `rgb2hsv` 这两个HSV和RGB图像互相转换的函数，但是没有HSI和RGB互相转换的函数，因此自己实现 `hsi2rgb.m`, `rgb2hsi.m` 这两个函数文件，也放在 `code` 目录下。

2. 结果

2.1 实验设置

2.2 实验结果