

数字图像处理作业一 实验报告

银琦 141220132 141220132@smail.nju.edu.cn 18260066573

一、实验环境

MATLAB R2015a

二、实现细节

我采用了五种方法对图像进行处理，其中两种为调用系统函数，三种为自己实现。

1. 直方图均衡化

```
function [output2] = hist_equal(input_channel)
    %you should complete this sub-function
    input_channel = im2double(input_channel);
    output2 = input_channel;
    [M,N]=size(input_channel);
    [counts,x]=imhist(input_channel);%H是读取的图像，imhist是对图像直方图进行统计，其中count，是每个灰度值得个数，x代表灰度值
    location=find(counts~=0);%找到所有像素个数不为0的灰度级
    MinCDF=min(counts(location));%找到包含个数最少的灰度级
    for j=1:length(location)
        CDF=sum(counts(location(1:j)));%计算各个灰度级像素个数累计分布
        P= input_channel==x(location(j));%找到图像中等于某个灰度级所有像素点所在位置
        output2(P)=(CDF-MinCDF)/(M*N-MinCDF);%%利用灰度换算公式，修改所有位置上的像素值
    end
end
```

读取图像，并调用 `imhist` 函数对图像直方图进行统计，利用灰度换算公式对像素值进行修改，得出处理后的图片，最后输出图片。

2. 直方图均衡化

```
function [output3] = hist_equal1(input_channel)
    %you should complete this sub-function
    [R, C, K] = size(input_channel); % 新增的K表示颜色通道数
    % output3 = input_channel;
    % 统计每个像素值出现次数
    cnt = zeros(K, 256);
    for i = 1 : R...
        f = zeros(3, 256);
        f = double(f); cnt = double(cnt);
        % 统计每个像素值出现的概率，得到概率直方图
        for k = 1 : K...
            % 求累计概率，得到累计直方图
            for k = 1 : K...
                % 用f数组实现像素值[0, 255]的映射。
                for k = 1 : K...
                    % 完成每个像素点的映射
                    output3 = input_channel;
                    % output3 = double(output3);
                for i = 1 : R...
```

end

读入图像后对每个通道分别统计像素值[0,255]出现的次数，然后对每个通道分别求像素值[0,255]出现的概率，得到概率直方图，对每个通道分别求像素值[0,255]概率的前缀和，得到累计直方图，对每个通道根据累计直方图分别求像素映射函数，对每个通道完成每个像素点的映射，最后输出直方图均衡化的图像。具体代码见 Histogram_equalization.m 文件。

3. 空域锐化

```
function [output4] = hist_equal2(input_channel)
    input_channel=im2double(input_channel);%转换数据类型，将uint8图像转为double类型，范围为0-1
    [height, width, R]=size(input_channel);%返回矩阵I的行列
    % output = input_channel;
    for i=2:height-1
        for j=2:width-1
            R(i,j)=abs(input_channel(i+1,j+1)-input_channel(i,j))+abs(input_channel(i+1,j)-input_channel(i,j+1));
        end
    end
    T=R;
    [m,n]=size(T);
    output4(1:m,1:n)=input_channel(1:m,1:n)+T(1:m,1:n);
end
```

利用一阶微分锐化增强，实现 Roberts 算子的锐化处理。Roberts 算子是突出图像的细节或者是增强被模糊了的细节。因此要对图像实现锐化处理，可以用空间微分来完成，但是，这样图像的微分增强了边缘和其他的突变(如噪声)并削弱了灰度变化缓慢区域。具体代码见 Histogram_equalization.m 文件。

4. 调用 `adapthisteq` 函数

该函数将基于全局的直方图均衡化操作变成了局部操作。原理是将图片先分成若干区域块，然后在每个区域中根据一定的输出直方图分布单独进行均衡化的处理，最后相邻的区域通过线性差值进行拼接，有效降低边界伪影。这样一来，该方法有效的抑制了噪声，也使得局部对比度得到了有效的增强。

5. 调用 `imadjust` 函数

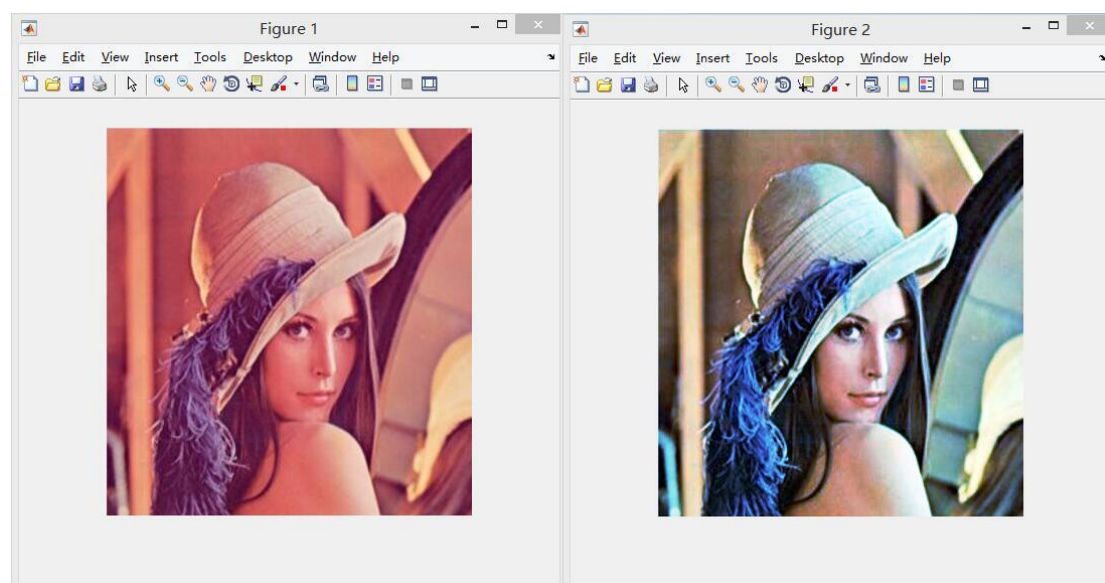
将灰度图像中的亮度值映射为新值，使得图像中 1% 的数据饱和至最低和最高亮度，这可以增加输出图像的对比度值。

三、使用说明

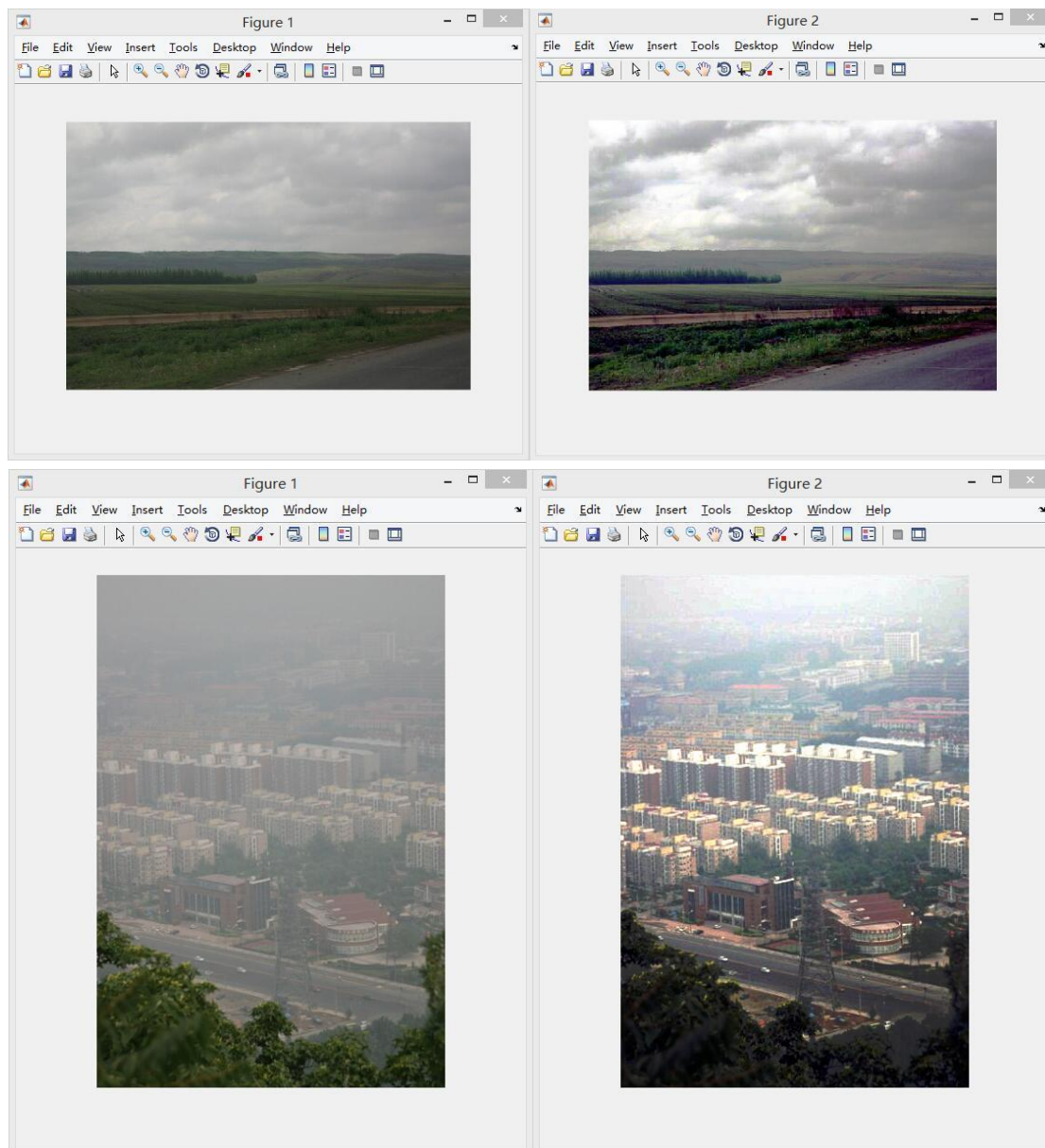
按照实验要求将代码都写进了 `Histogram_equalization.m` 中，共实现了三个函数：`hist_equal`，`hist_equal1`，`hist_equal2`。在函数 `Histogram_equalization` 中调用了这三个函数以及两个系统函数，只有一个函数未被注释，其余四个函数均被注释，在使用给出的测试文件进行测试时取消注释即可。

四、部分实验结果

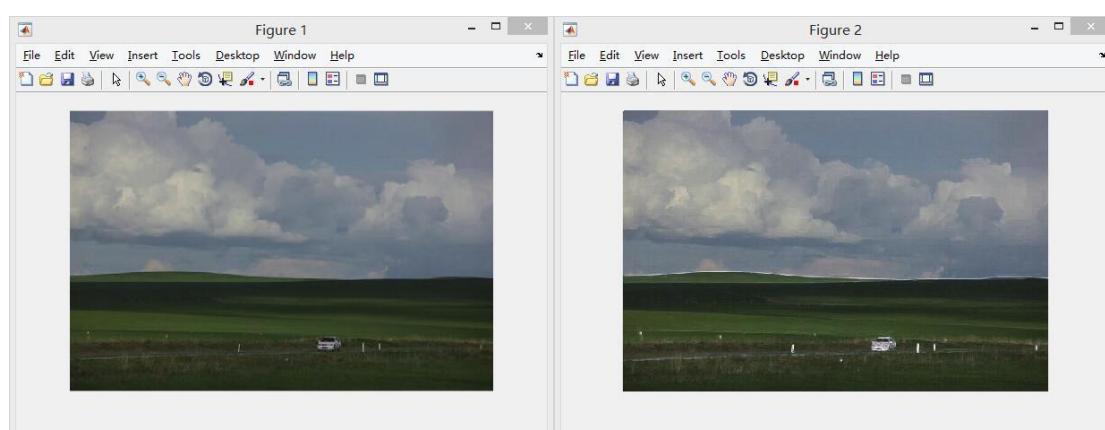
调用 `hist_equal` 函数处理 RGB 图像



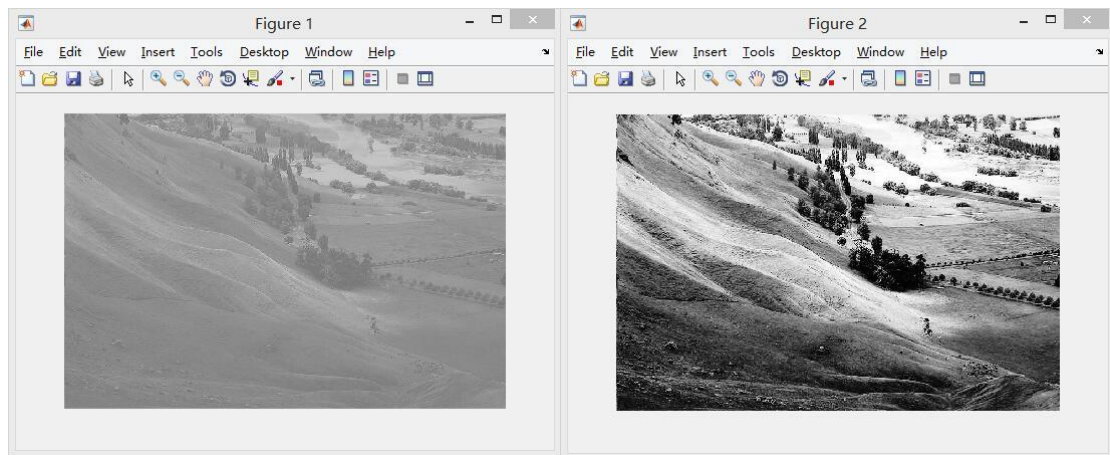
调用 hist_equal1 处理 RGB 图像



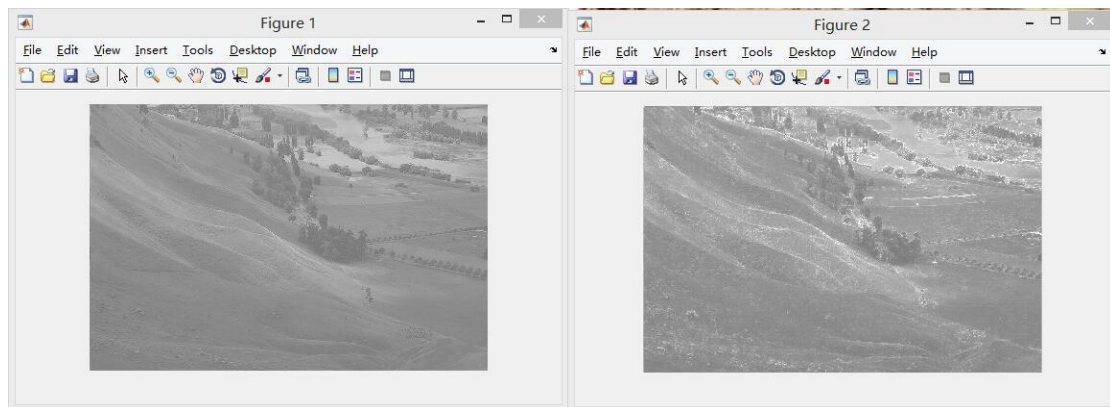
调用 hist_equal2 处理 RGB 图像



调用 hist_equal1 处理灰度图像



调用 hist_equal2 处理灰度图像



五、参考资料

- [1] <https://www.zhihu.com/question/56491446>
- [2] http://blog.csdn.net/cy_543/article/details/41548399
- [3] <http://3y.uu456.com/bp-84226q18fad61qsf312ba6dd-1.html>