

浙江大学实验报告

戴毅阳 (3200104915)

课程名称: 图像信息处理 指导老师: 宋明黎

实验名称: 图像的对数增强和直方图均衡化

一、实验目的和要求

1. 目的:

- (1) 学习图像增强的处理方式, 以对数处理的方式增强图像
- (2) 学习直方图均衡化

2. 要求:

- (1) 对数增强图像
- (2) 直方图均衡化

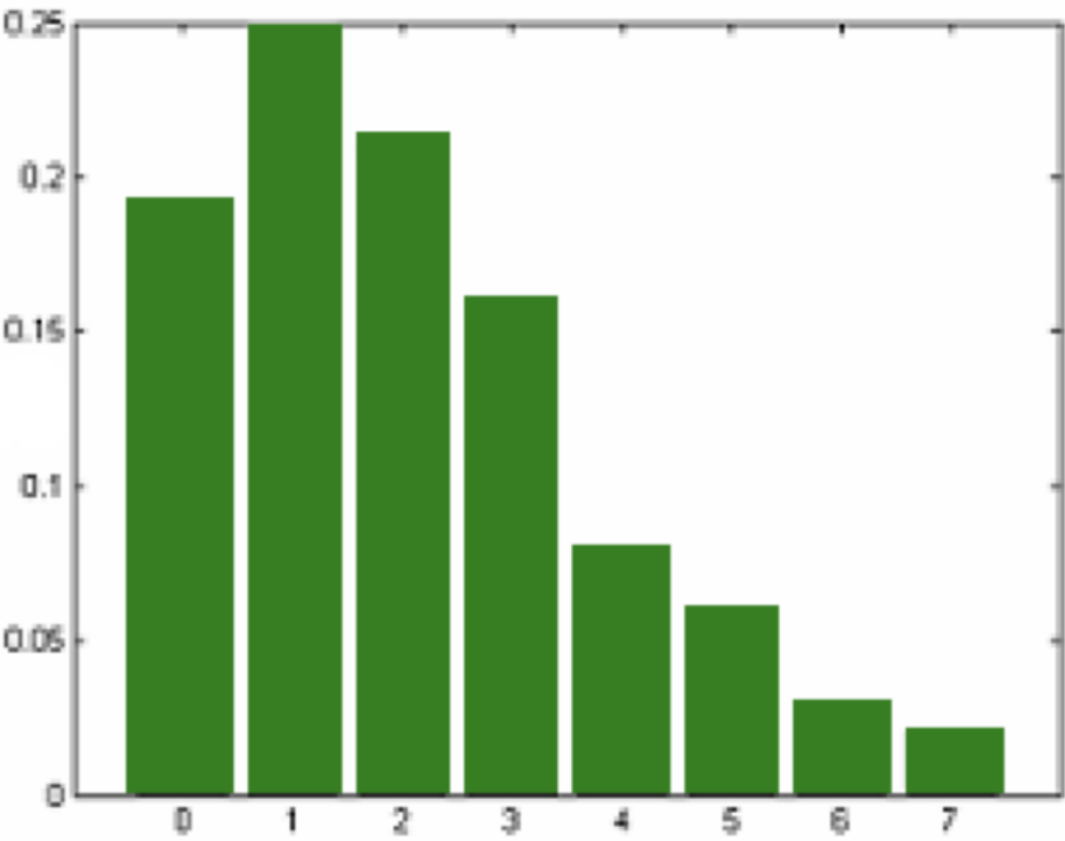
二、实验内容和原理

1. 图像增强(主要为对数增强) 为了实现图像可视性的增强, 采取以下公式:

$$L_d = \frac{\log(L_w + 1)}{\log(L_{\max} + 1)}$$
 其中 L_d 为我们所要求的增强后的亮度, 而 L_{\max} 为原图的最大亮度值。在对于图片所有像素点的亮度更新后, 可知其范围处于 $[0, 1]$ 之间, 下一步则是归一化, 将其分为放大至 $[0, 255]$ 的范围。

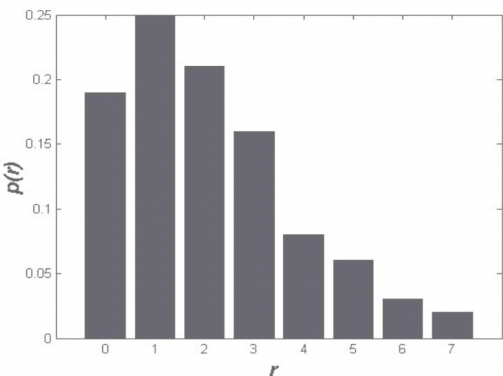
2. 直方图均衡化(灰度图):

灰度直方图是一种统计图，它表示给定图像的总像素数中不同灰度级的像素数所占的比例。

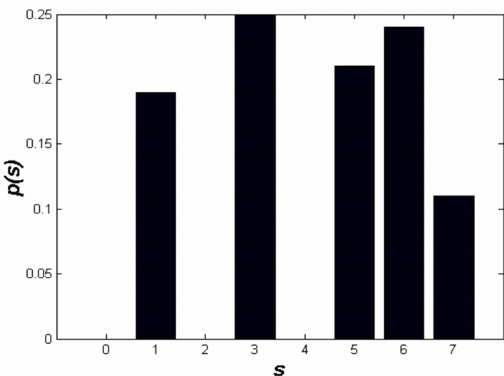


上图即为一副图像的直方图

对于一副给定的24位bmp图像，先将其转化为灰度图，然后求出0-255每一个灰度级别出现的次数、频率、累计频率 n_k, r_k, s_k $r_k = \frac{n_i}{n}$ $s_k = \frac{1}{n} \sum_{i=0}^n n_i = s_{k-1} + r_k$



(a)Before histogram equalization



(b) After histogram equalization

三、实验步骤及分析

1.对数增强

下方即为使用对数增强图像可视性的代码：

```
fpwrite = fopen((bmpname+"_enhancement.bmp").c_str(), "wb");  
// 找到最大值和最小值 (其中+1防止log(0)出现)  
double maxvalue = log(1+*max_element(graydata, graydata+size));  
double minvalue = log(1+*min_element(graydata, graydata+size));  
cout << maxvalue << endl << minvalue << endl;  
// 此处将对数处理和归一化 (线性) 结合起来一起完成了  
for(int i=0; i<size; i++) graydata[i] = (int)(255*(log(1+graydata[i])-  
minvalue)/(maxvalue-minvalue));  
write(graydata, &fpwrite);  
fclose(fpwrite);
```

2.直方图均衡化

```
void equalizeHist1(BYTE graydata[])  
{  
    int gray_num[256] = {0};  
    double gray_p[256];  
    double gray_cumulative_p[256];  
    int gray_equal[256], size = bmpInfo.biSizeImage;  
    for(int i=0; i<size; i++) gray_num[graydata[i]]++;  
    for(int i=0; i<256; i++) gray_p[i] = 1.0*gray_num[i]/size;  
    gray_cumulative_p[0] = gray_p[0];  
    for(int i=1; i<256; i++) gray_cumulative_p[i] = gray_cumulative_p[i-  
1]+gray_p[i];  
    for(int i=0; i<256; i++) gray_equal[i] = floor(255.0 *  
gray_cumulative_p[i] + 0.5);  
    for(int i=0; i<size; i++) graydata[i] = gray_equal[graydata[i]];  
}
```

上方为灰度图均衡化的主要函数代码，下面给出调用该函数时的情景：

```
fread(graydata, sizeof(BYTE)*size, 1, fpread);  
// for(int i=0; i<10; i++) cout << (int)graydata[i] << " ";  
// cout << endl;  
fclose(fpread);  
equalizeHist1(graydata);  
fpwrite = fopen((bmpname+"_equal1.bmp").c_str(), "wb");  
write(graydata, &fpwrite);  
fclose(fpwrite);
```

四、运行环境及方法

1. 编译器：clang++
2. 编辑器：vscode
3. 操作系统：macOS

4. 运行方法：已写makefile

```
make  
make run  
...(输入文件名即可)  
make clean
```

五、成果展示

此处给出两幅图的代码运行后的结果：

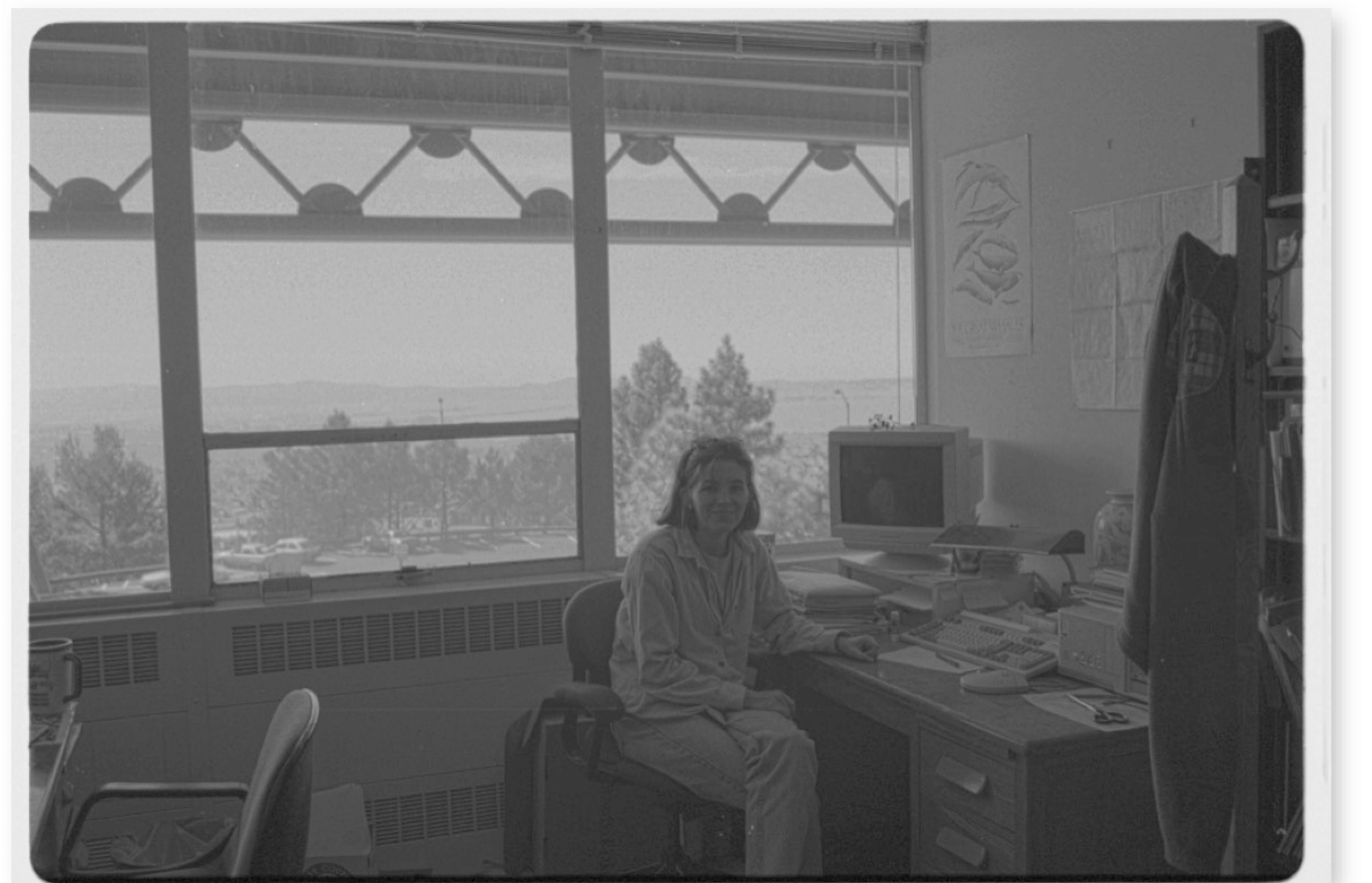
1. 图片1



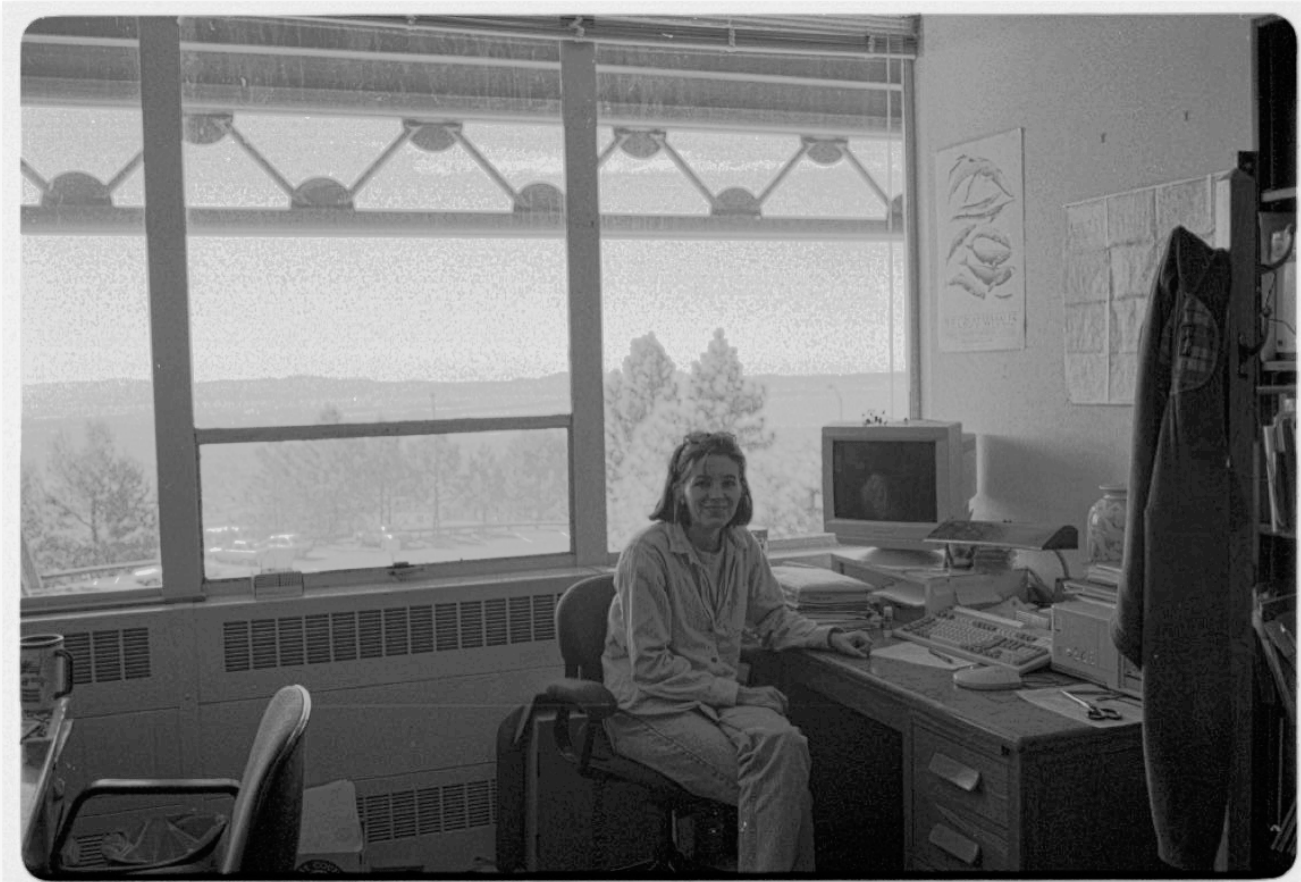
(0) 灰度图



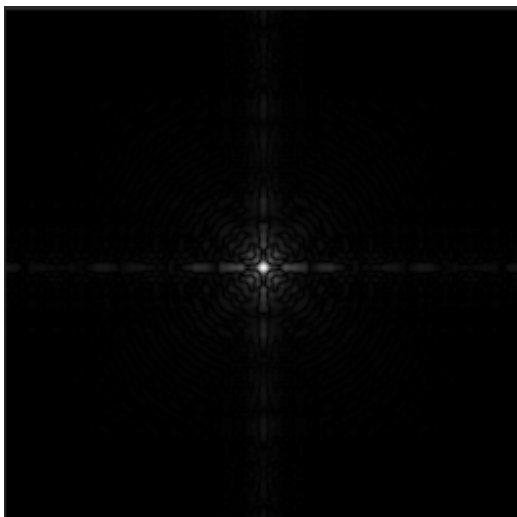
(1) 对数增强：



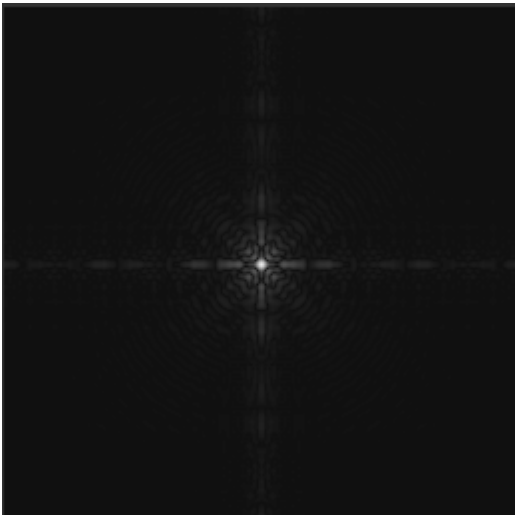
(2) 直方图均衡化



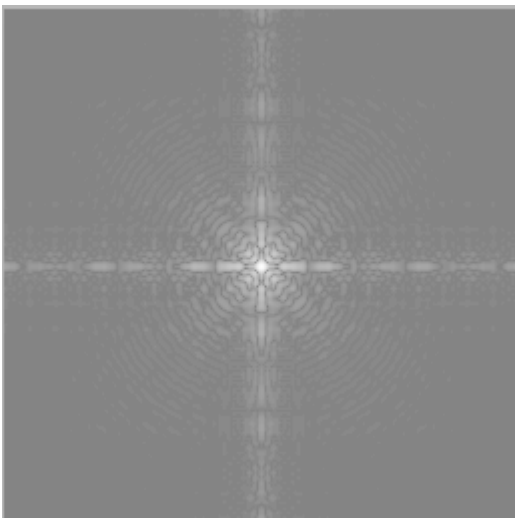
2. 图片2



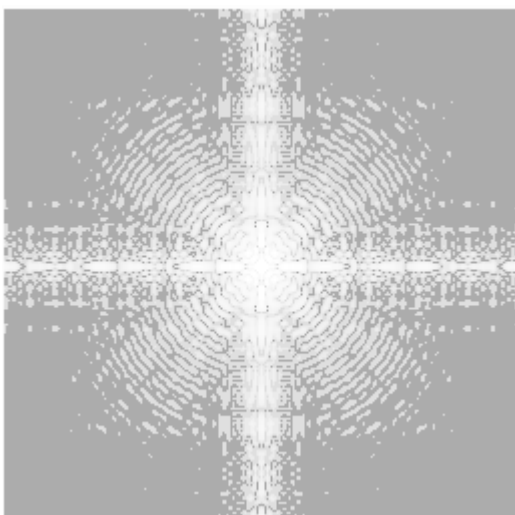
(0) 灰度图



(1) 对数增强



(2) 直方图均衡化



六、心得体会

通过这次的实验，我对于图像增强和直方图均衡化有了更加深刻的认识。但是，我依旧有一个疑惑无法消除：在对于灰度图像进行增强和均衡化的过程中，会出现图片变淡的情况，至今无法解决。