

浙江大学实验报告

专业：计算机科学与技术

姓名：颜晗

学号：3200105515

日期：2021/11/14

课程名称： 图像信息处理 指导老师： 宋明黎 成绩：

实验名称： 图像对数运算可视化增强与直方图均衡化

一、实验目的和要求

了解图像可视化增强的两种方法，并学会使用 C 语言实现这两种方法。

二、实验内容和原理

实验内容：1.在灰度图的基础上对图像灰度作对数处理，使图像增强；

2.获取灰度图的灰度直方图并根据直方图重新规划图像灰度分布，使图像灰度分布更均匀。

实验原理：1.在 0~1 的区间上，对数函数在较小区间的增长率大而较大区间的增长率很小，从而使图像的灰度整体向较大的数值偏，即亮度增加。

2.通过统计图像各灰度值的数量与比例，再将其重新映射到某个区间，可以将占比非常小的灰度值归入某相邻灰度中，从而使图像的灰度值分布相对均匀且对比度更加强烈。

三、实验步骤与分析

1.对数运算可视化增强

由于灰度图只需要一个字节显示一个像素，也不需要像彩色图一样进行变换，只需要一个公式计算即可完成。

$$L_d = 255 * \frac{\log_2(L_w + 1)}{\log_2(256)}$$

L_w 即像素的原灰度， L_d 为对数运算后的灰度，对数内的值加 1 是防止灰度 0 的出现使对数公式出错。

实现代码如下：

```
1. void log_visi(myBmpFile* bmp_log,myBmpFile* bmpImg)
2. {
3.     int i;
```

```

4.   unsigned long size=bmpImg->BIH.biSizeImage;
5.
6.   bmp_log->BFH=bmpImg->BFH;//复制文件头
7.   bmp_log->BIH=bmpImg->BIH;//复制信息头
8.   bmp_log->Palette=(myRgbQuad*)malloc(sizeof(myRgbQuad)*256);
9.   for(i=0;i<256;i++){
10.      bmp_log->Palette[i].rgbRed=i;
11.      bmp_log->Palette[i].rgbGreen=i;
12.      bmp_log->Palette[i].rgbBlue=i;
13.      bmp_log->Palette[i].rgbReserved=0;
14.  } //建立调色板
15.  bmp_log->imageData=(unsigned char*)malloc(sizeof(unsigned char)*size
    );
16.  memcpy(bmp_log->imageData,bmpImg->imageData,size);
17.  //将源bmp文件的信息克隆到目标bmp文件中
18.
19.  for(i=0;i<size;i++){
20.      bmp_log->imageData[i]=255*log(bmp_log->imageData[i]+1)/log(256);
21.  }
22.
23.  return ;
24.}

```

2.直方图均衡化

2.1 统计各灰度值在图像像素中的数目 2.2 计算各灰度占总灰度的比例 2.3 计算灰度的累计分布概率（可以与 2.2 合并） 2.4 将灰度重新映射至新区间。

直接对原图像进行直方图均衡化易导致背景噪点的出现，反而影响图像的观看效果，因此在具体处理中，我忽略了灰度值大于 200 的像素点，而将 200 以下的灰度值重新映射至 0~200，从而尽可能避免了噪点出现，（方便起见，概率分布计算依旧按照完整图像进行）。

代码如下：

```

1. void histeq(myBmpFile* bmp_hist,myBmpFile* bmpImg)
2. {
3.     int i;
4.     unsigned long size=bmpImg->BIH.biSizeImage;
5.
6.     bmp_hist->BFH=bmpImg->BFH;
7.     bmp_hist->BIH=bmpImg->BIH;
8.     bmp_hist->Palette=(myRgbQuad*)malloc(sizeof(myRgbQuad)*256);

```

```

9.     for(i=0;i<256;i++){
10.         bmp_hist->Palette[i].rgbRed=i;
11.         bmp_hist->Palette[i].rgbGreen=i;
12.         bmp_hist->Palette[i].rgbBlue=i;
13.         bmp_hist->Palette[i].rgbReserved=0;
14.     }
15.     bmp_hist->imageData=(unsigned char*)malloc(sizeof(unsigned char)*size);
16.     memcpy(bmp_hist->imageData,bmpImg->imageData,size);
17.     //将源 bmp 文件的信息克隆到目标 bmp 文件中
18.
19.     int gray_num[256];
20.     for(i=0;i<256;i++){
21.         gray_num[i]=0;
22.     }//初始化数组
23.     for(i=0;i<size;i++){
24.         gray_num[bmp_hist->imageData[i]]++;
25.     }//统计灰度直方图
26.
27.     double P[256];
28.     for(i=0;i<256;i++){
29.         P[i]=(double)gray_num[i]/size;
30.         if(i>0)P[i]=P[i]+P[i-1];
31.     }//计算累计分布概率
32.     for(i=0;i<size;i++){
33.         if(bmp_hist->imageData[i]<201)
34.             bmp_hist->imageData[i]=200*P[bmp_hist->imageData[i]];
35.         //bmp_hist->imageData[i]=255*P[bmp_hist->imageData[i]];//未改动
36.     }//对于较小灰度,重新映射至 0~200
37.     return ;
38. }

```

四、实验环境及运行方法

本人使用 dev-c++ 5.11 版 C 语言工程运行代码,如果电脑上安装有 dev-c++,单击文件“bmpwork.dev”即可打开工程文件,点击编译运行的图标或按 f11 即可。

使用 vscode 运行的方法:打开程序所处文件夹(包含 bmp.h 、bmp.c 以及 main.c 文件),在终端输入“gcc bmp.c main.c -o main”,敲击 enter 键后即可编译生成 main.exe,再直接在终端输入“.\main”或在文件夹中打开 main.exe 即可运行程序得到结果。

注意:在运行程序前,保证待操作的 bmp 文件在对应的路径(sample 文件夹)中。

五、实验结果展示

灰度原图	对数运算
	
直方图均衡化（改）	直方图均衡化（未改）
	

六、心得体会

本次作业主要是理解两种方法使图像可视化增强的原理，学习对应的像素灰度变换公式代入计算即可，实验过程基本未遇见什么问题。就是用 matlab 打开图像时底部会出现混乱像素点且图像移位，抛开移位，对应图像的显示不存在问题，暂时还未找出原因，说明对图像的了解依旧不够多，还需要平时再仔细研究，让图片在各平台都可以完整显示出来。