程序设计说明文档

Bmp文件为



一．图像处理

BMP 文件的数据按照从文件头开始的先后顺序分为四个部分：

bmp 文件头(bmp file header)：提供文件的格式、大小等信息。

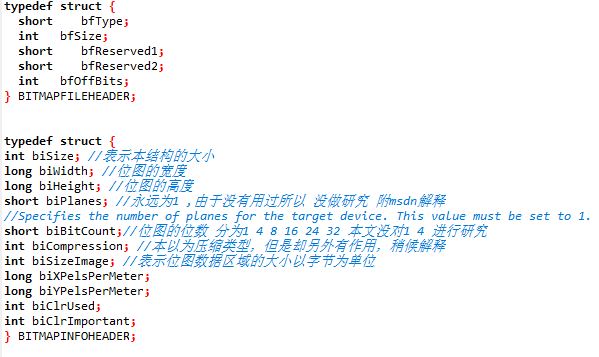
位图信息头(bitmap information)：提供图像数据的尺寸、位平面数、压缩

方式、颜色索引等信息。

调色板(color palette)：可选，如使用索引来表示图像，调色板就是索引

与其对应的颜色的映射表。

位图数据(bitmap data)：就是图像数据。

1. 学习bmp文件的数据形式并命名：下面的代码是定义了位图文件头和位图信息头的两个结构体。

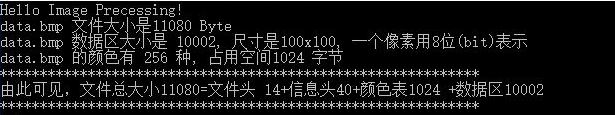
下一步

1. 二进制文件的读写

为了输出bmp文件的各项数据，以及对文件的各项操作，首先定义结构体名称，文件大小，数据区大小，颜色表大小，宽度，高度，像素位数，颜色字节数等参量；其次定义为文件指针，以实现文件的打开读写等工作。接下来以二进制方式打开bmp文件，使用指针和fopen函数以读的方式打开二进制文件data.bmp。

文件解析工作，第一步是读取文件头信息，使用fread函数从指针fp\_bmp中读取数据，其中文件头中的数据共有14个字节，&filehead作为接收数据的内存地址。则将获得的数据放入开始时定义的变量，既可以得出文件的大小。

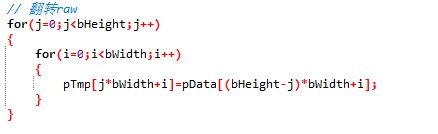
第二步读取信息头，同样地，使用fread函数从指针fp\_bmp中读取数据，信息头中数据共有40个字节，&infohead作为接收数据的内存地址。则将获得的数据放入开始定义的变量，即可以获得信息头内的各项数据。

（3）最后是输出信息结果

注意注意：像素字节必须是4的整数倍，否则需要补齐。

（4）将data.bmp转化为raw格式

若要生成raw文件，则首先要申请动态内存。定义指针后使用malloc函数申请内存空间。使用fseek函数使文件指针从当前位置移动nColor\*4个字节，即跳过颜色表，到达数据区。然后使用fread函数从文件中读取bmp图片的所有字节，将数据储存在p1所指的数据块中。使用fp\_raw指针以及fopen函数以写的形式打开二进制raw文件。然后使用fwrite函数从p1所指的数据块中获取数据，写入二进制文件data.raw中，然后进行翻转。然后关闭文件data.bmp和data.raw。其转化为raw格式的图片在photoshop中打开如下图所示：（由于bmp 图像文件的数据是从下到上，从左到右的。即从文件中最先读到的是图像最下面一行的左边第一个像素，然后是坐标第二个，接下来是倒数第二行的第一个像素。所以读取之后的raw图片本是与原bmp图片上下左右颠倒的，在输出raw文件之间输入翻转代码将其翻转后再输出，则保持上下一致）



二.图像处理

（1）图像的旋转和翻转

我进行的第一步是图像的90度旋转，使用fp\_raw指针以及fopen函数以写的形式打开二进制raw文件。然后使用fread函数从文件中读取raw图片的所有字节，将数据储存所指的数据块中。对数据块的宽度和高度进行坐标的变换，用双层for循环实现90度顺时针旋转，将旋转后的数据再次存储所指数据区中。使用fp\_process指针和fopen函数以写的形式创建process.raw文件，随后使用fwrite函数将从再次存储所指数据区获取的数据写入process.raw文件中，即实现了图像的90度旋转。process.raw图片在photoshop中打开如下图所示：



第二步是图像的180度旋转，与上面相同的地，打开raw文件后使用fread函数从文件中读取raw图片的所有字节，将数据储存所指的数据块中。对宽度和高度进行坐标变换，用双层for循环实现180度旋转，将旋转后的数据再次存储在所指数据区中。使用fp\_process指针和fopen函数以写的形式创建process1.raw文件，随后使用fwrite函数将从在此存储所指数据区获取的数据写入process1.raw文件中，即实现了图像的180度旋转。process1.raw图片在photoshop中打开如下图所示：



第三步是图像的270度旋转，由于前面做的是顺时针的旋转，所以旋转270度即将图片逆时针旋转90度，打开文件和读取文件方式同上，然后进行与顺时针旋转90度相反的坐标变换，就实现了图像的270度旋转。创建process2.raw文件和数据的写入同上。process2.raw图片在photoshop中打开如下图所示：



第四步是图像的上下翻转。打开文件和读取文件方式同上，同样使用坐标变换，上下变换的实质是列数不变，行数变为原来的总行数减原行数，利用双重for循环即可实现。创建process3.raw文件和数据的写入同上。process3.raw图片在photoshop中打开如下图所示：

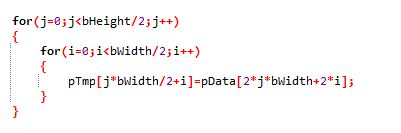


第五步是图像的左右翻转。打开文件和读取文件方式同上，同样使用坐标变换，上下变换的实质是行数不变，列数变为原来的总列数减原列数，利用双重for循环即可实现。创建process4.raw文件和数据的写入同上。process4.raw图片在photoshop中打开如下图所示：



(2)图像的缩放

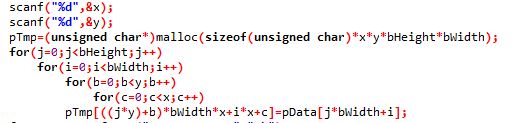
第一步是图像的缩小。打开文件和读取文件方式同上。坐标变换时要使用双线性差值的方法，先设定缩小倍数：宽度变为原来的1/2，高度变为原来的1/2。即已知图像像素为100\*100，现在变为50\*50，前后图像的边长比分别为2和2，原图像坐标为（i，j）是，缩小后的图像坐标为（i/2,j/2）,代码如下图：



创建process5.raw文件和数据的写入同上。process5.raw图片在photoshop中打开如下图所示：



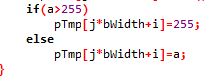
第二步是图像的放大。打开文件和读取文件方式同上。不过由于要把图片放大，所以p2所指的数据区也要相应增大，所以要额外申请p2的内存空间。设定放大倍数：宽度变为原来的x倍，高度变为原来的y倍。即已知图像像素为100\*100，现在变为(100\*x)\*(100\*y)，前后图像的边长比分别为1/x和1/y，原图像坐标为（i，j），放大后的图像坐标为（i\*x, j\*y）,代码如下图：(注意x,y是由根据自己输入)



创建process6.raw文件和数据的写入同上。process6.raw图片在photoshop中打开如下图所示：（如图为x=5,y=5时的图）

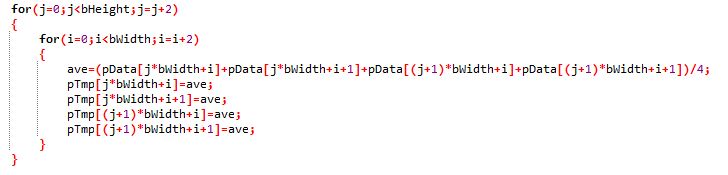


3）图像的加噪和去噪

 第一步是图像的加噪，在主函数中对高斯函数进行调用，在输入像素中加入高斯随机数，重新将像素值限制在（0，255）之间，代码如下：

创建process7.raw文件和数据的写入同上。process7.raw图片在photoshop中打开如下图所示：



第二步是图像的去噪（均值滤波）。打开文件和读取文件方式同上。线性滤波的基本原理是用均值代替原图像中的各个像素值，选用一个2\*2的像素组成模板，将该模板的均值代替该模板内所有像素，从而进行输出，具体代码如下：

创建process8.raw文件和数据的写入同上。process8.raw图片在photoshop中打开如下图所示：



（4）图像的亮度调整和反色

第一步是图片的反色。即对图像的像素进行取反，由于该灰度图共有256种颜色，所以新得出的像素数据应该是255减去原来的像素数据，就得要与原来相反的像素。创建process9.raw文件和数据的写入同上。process9.raw图片在photoshop中打开如下图所示：

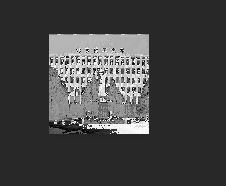


第二步是图像的亮度调整。当对图片数据加或乘一个正整数，图片将变亮；当对图像数据减或除一个正整数图片将变暗。同时重新将像素值限制在（0，255）之间。创建process10.raw文件和数据的写入同上。Process10.raw图片在photoshop中打开如下图所示：（当数据加5时（图片变亮）：）

三．图像编码

（1）图像的编码

打开文件和读取文件方式同上。将图像中读出的数据与100进行异或操作，即以100为密匙对原数据进行加密操作，放入新数据中。创建process11.raw文件和数据的写入同上。process11.raw图片在photoshop中打开如下图所示：



（2）图像的解码

打开文件和读取文件方式同上，不过打开的文件是前面加密形成的图像process11.raw。将图像中读出的数据与100进行异或操作，即以100为密匙对原数据进行解密操作，放入新数据中。创建process12.raw文件和数据的写入同上。process12.raw图片在photoshop中打开如下图所示：



总结

通过这次作业的完成和学习，我更加了解到c语言的一些应用和知识，知道了如何用c语言读取和输出文件，更会简单的用C语言进行一些图像处理，图像编辑。让我对for循环有了更充分的认识，了解到它功能的强大。了解到一个bmp文件所包含的信息，内容和大小。最最重要的是，我学会了对一个图片进行加密处理，可以保护自己的隐私。收获颇多，感谢老师。