一、工程包含的三个文件

dmcp.cmd：链接命令文件。

main.c：包含主函数（main()），通过调用系统函数和自定义的函数实现算法的主流程，即：打开文件——读取图像数据——图像处理——存储图像文件。

imagefun1.c：包含自定义的函数，程序的主要功能。

二、主要的变量

1、main.c中开头的几个变量用了关键字“extern”，表示外部引用，即这些变量是在其他文件（imagefun1.c）中定义的。

2、i\_img：输入图像点阵的起始地址。定义为“char”（8位）类型的指针，使用方法与一维数组类，似可通过i\_img[k]的形式存取每一个字节，其中k为偏移量。之所以未使用数组，是因为图像大小未知，无法预先确定数组大小，用此方法通过alloc\_mem(buffer\_size)函数动态分配存储空间。

3、o\_img：输出图像点阵的起始地址。与i\_img相似。

4、img\_row,img\_col,line\_size：图像高度(行数)，图像宽度(列数)，图像每行字节数。

注意img\_col与line\_size不一定相等，对于8位灰度图，只有img\_col为4的整数倍时才相等，否则应这样计算：line\_size = ((img\_col+3)/4)\*4。

因为实验中图像是120\*120的，这样line\_size = img\_col，为方便，后面都统一称为图像宽度，不再说明。

5、BMP\_Head：是一个结构型变量，其每个字段与《位图文件格式》中图像头部各个字节对应，将图像头部按字节顺序读入此结构，即可方便地以各个字段表示图像的对应属性，如：bmp\_head.ImageHeight即为图像的高度值。

三、自定义的几个主要函数

1、ReadBMPHeadInfo(filename)：读取图像头部信息。其中文件操作需要“stdio.h”库。

2、alloc\_mem(buffer\_size)：动态分配存储空间，其中用到的malloc(buffer\_size)系统函数需要“stdlib.h”库。

3、load\_data(filename,in\_img)：从图像文件中按字节顺序将图像点阵数据读出，存储到in\_img开始的连续空间。

4、save\_data(filename,buffer)：将buffer开始的连续空间的图像点阵数据写回图像文件。

5、FanSe(ip,jp,lx,ly)：对图像进行反色处理。

四、程序执行流程（main()函数）

蓝色字符为系统函数，品红色为自定义函数。

|  |  |
| --- | --- |
| char filename[40]; |  |
| printf("Please input BMPimage filename[\*.bmp]:\n"); | 提示输入文件名。 |
| scanf("%s",filename); | 读取文件名。 |
| ReadBMPHeadInfo(filename); | 读取图像头部信息。 |
| printf("opened\n"); | 输出提示信息。 |
| i\_img = (unsigned char \*)alloc\_mem(img\_row\*line\_size); | 动态分配存储空间，起始地址赋值给指针i\_img，空间大小为图像高度\*宽度。 |
| load\_data(filename,i\_img); | 读取图像点阵数据，存储到i\_img开始的连续空间。 |
| o\_img = (unsigned char \*)alloc\_mem(img\_row\*line\_size); | 分配存储空间给输出图像指针。 |
| FanSe(i\_img,o\_img,line\_size,img\_row); | 图像反色运算，结果存入o\_img开始的连续空间。 |
| save\_data(filename,o\_img); | 将处理后的输出图像数据存回图像文件。 |
| free(i\_img); | 释放内存。 |
| free(o\_img); |
| printf("zz\n"); | 输出提示信息。 |

五、反色处理函数（FanSe(ip,jp,lx,ly)）

1、各条语句讲解

|  |  |  |
| --- | --- | --- |
| void FanSe(ip,jp,lx,ly)  unsigned char \*ip,\*jp;  unsigned long lx,ly; | 调用时的语句为：FanSe(i\_img,o\_img,line\_size,img\_row)，参数对应：ip输入图像数据指针i\_img，jp输出图像数据指针o\_img，lx图像宽度line\_size，ly图像高度img\_row。 | |
| { |  | |
| unsigned long i,j; | 行号i，列号j。 | |
| for(i=0;i<ly;i++) | 每一行循环 | 此两行相当于按行列顺序依次访问每一个像素点（遍历图像）。 |
| for(j=0;j<lx;j++) | 每一列循环 |
| { |  | |
| gg(i,j)=0; | 输出图像灰度值预设为0。 | |
| } |  | |
| for(i=0;i<ly;i++) | 遍历图像。 | |
| for(j=0;j<lx;j++) |
| { |  | |
| gg(i,j)=256-1-ff(i,j); | 图像反色运算。 | |
| } |  | |
| } |  | |

2、ff(i,j)、gg(i,j)的含义：

文件开始处两条语句：

#define ff(i,j) ip[(long)(i)\*lx+j]

#define gg(i,j) jp[(long)(i)\*lx+j]

此为宏定义，即编译前用后面的字符串代替前面的字符串。ff(i,j)替换为ip[(long)(i)\*lx+j]，即输入图像指针ip偏移量为i\*lx+j的字节，也就是第i行j列的像素点的灰度值。