数据结构专题实验5说明文档

**（一） 实验目的：**了解图片文件格式，掌握图元识别处理图片文件的方法。

**（二） 基本要求：**自定义合适的抽象数据类型按照要求存储数据，并使用图元识别处理图片文件格式的数据。

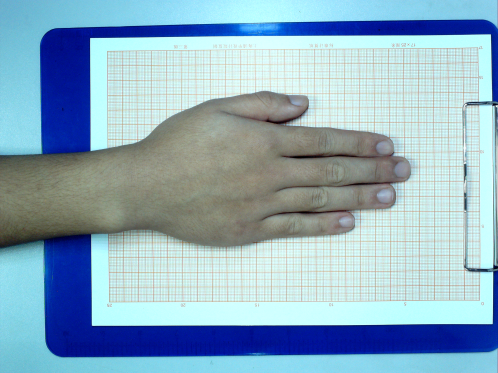
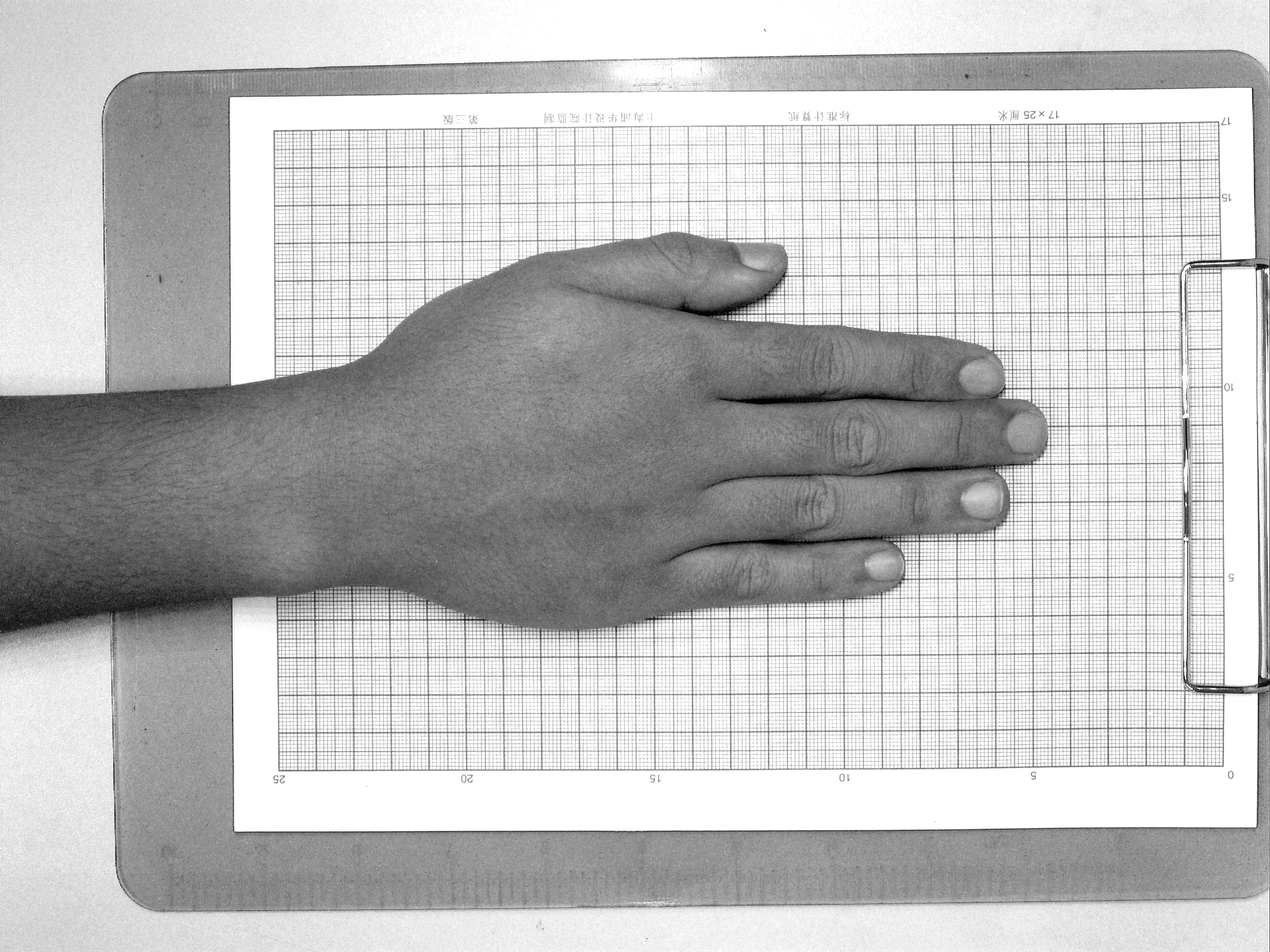
**（三） 实验内容：**

实验处理的是24位BMP文件，典型的BMP图像文件由四部分组成：

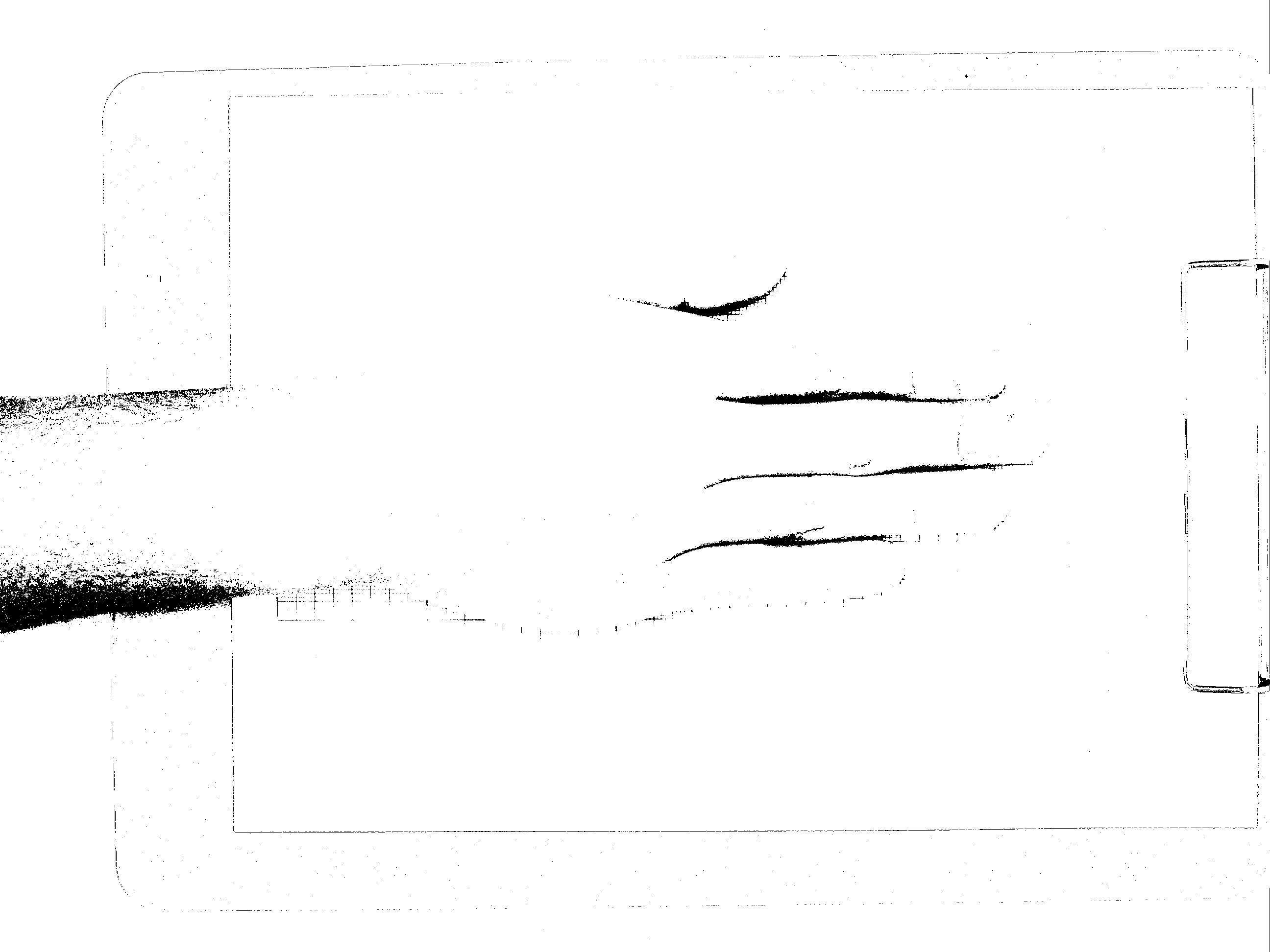
|  |  |
| --- | --- |
| 位图头文件数据结构 | 包含BMP图像文件的类型、显示内容等信息 |
| 位图信息数据结构 | 包含有BMP图像的宽、高、压缩方法，以及定义颜色等信息 |
| 调色板 | 这个部分是可选的，本次实验不必关注 |
| 位图数据 | 24位图中直接使用RGB数值，而其他的小于24位的使用调色板中颜色索引值 |

位图数据是按照一定的格式在计算机中存储，包括了上述的四个组成部分，本次实验需要修改的只有第四部分——位图数据。假设有一张1944\*2592的24位彩色bmp图片，那么它的位图数据就是它所有像素对应的颜色值，图像左上角的像素点颜色对应矩阵M[0][0]的值，当以RGB三个分量来表示颜色，每个分量的值从0到255时，我们可以认为这张24位彩色bmp图片的位图数据是三个1944\*2592的矩阵，每个矩阵代表一个颜色通道，也就是说一个1944\*2592的矩阵代表R（红色）通道分量的值，一个1944\*2592的矩阵代表G（绿色）通道分量的值，一个1944\*2592的矩阵代表B（蓝色）通道分量的值。

为了方便数据处理，首先需要将RGB三个通道的彩色图片转换成一个通道的灰度图，从三个1944\*2592的矩阵变成一个。一般的转换公式是gray = (77 \* R + 151 \* G + 28 \* B) >> 8;提供的源码使用的方法是gray = R，这样做是为了更有效的减弱蓝色垫板的影响。

一般来说，指缝比手指其他地方的颜色深，为了获得颜色较深的部分而抛弃掉颜色较浅的部分，需要将灰度图转换成为二值图。具体做法是从0（黑色）到255（白色）中选择一个阈值T，遍历1944\*2592的矩阵，凡是小于等于阈值T的位置全部都赋值为0，其余部分赋值为255，通过调整阈值T的大小，可以将重要的黑色部分保留，而其余部分均为白色。下图T选择为70。

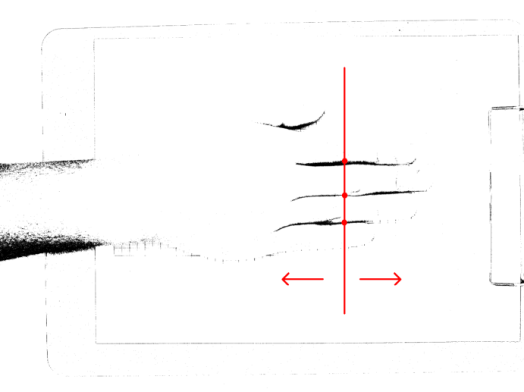
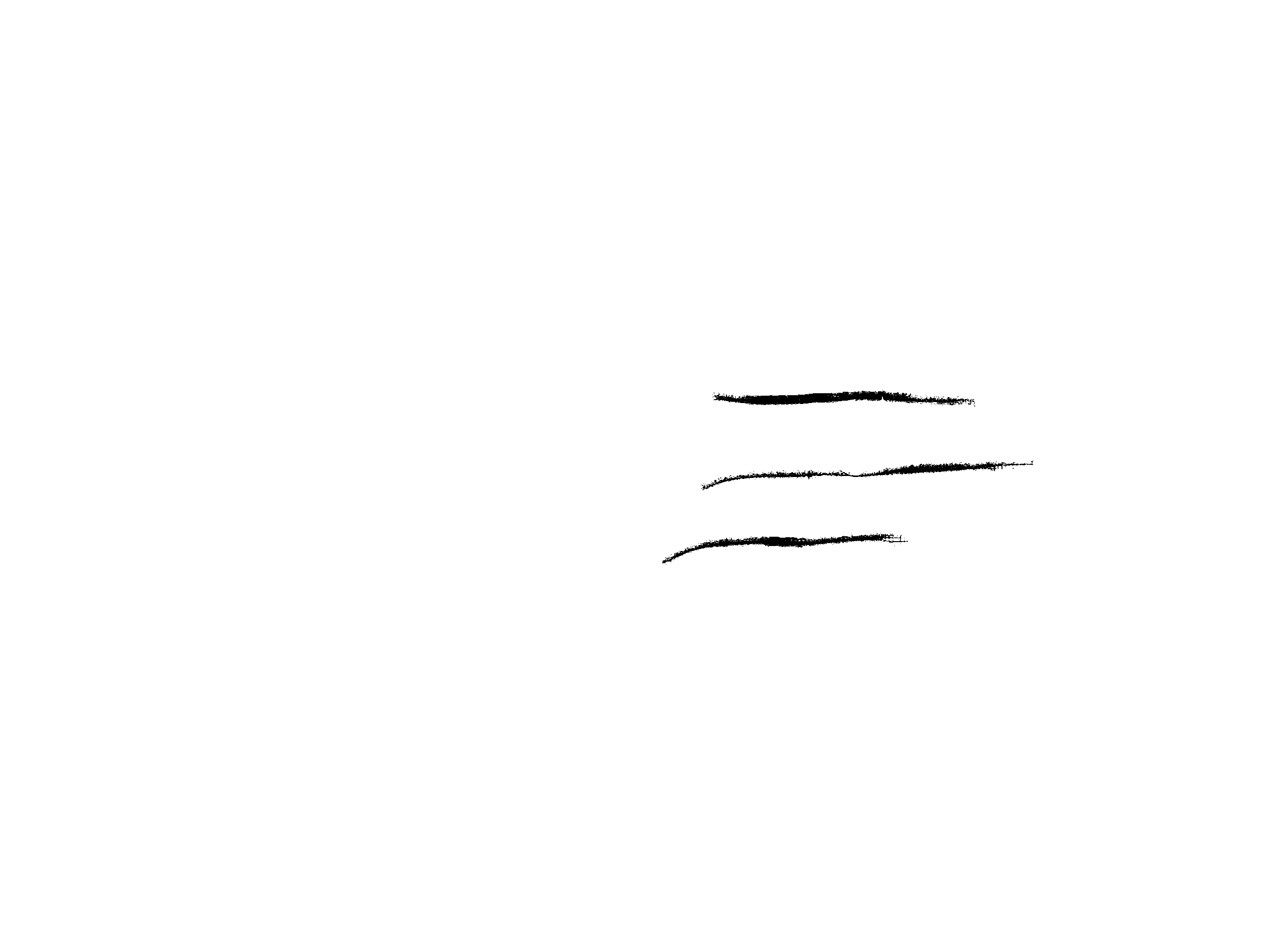


估计一个大概位置，使得位图数据矩阵的这一列同时包含三条指缝的黑色部分，如下图所示，具体做法是：选择矩阵的第j列(比如j=1700，如果不满足条件再换其他数值)，遍历第j列第300行到1600行之间数据，如果遇得到以M[i][j]为中心的3\*3的模板，即把当前位置和它周围8个像素的灰度值相加，灰度值之和小于255\*3，可以认为M[i][j]是某条指缝上一点。此时将i加上一个较大的值，比如100，跳过该指缝上其他满足条件的点，这样可以在每条指缝上只选取一个标识点。一直向下遍历，直到在3条指缝上各找到一个标识点。

3\*3模板相关位置如下表所示：

|  |  |  |
| --- | --- | --- |
| M[i-1][j-1] | M[i-1][j] | M[i-1][j+1] |
| M[i][j-1] | M[i][j] | M[i][j+1] |
| M[i+1][j-1] | M[i+1][j] | M[i+1][j+1] |

将每条指缝的标识点作为起点，假设为起点为si[i][j]，寻找si[i-1][j]上下一定范围内（si[i-1][j-m~j+m]）的灰度值为0的点，将其灰度赋值改为1，在这些点中选取合适一点，作为新的起点，重复上述步骤，直至将标识点左边指缝位置的灰度全部改为1；用类似的方法将标识点右边指缝位置的灰度也改为1；最后将除了灰度值为1的点外的所有点的灰度值赋值为255，输出图像即可。

**（三） 代码说明：**

bool readBmp(char \*fileName)

读入图像数据，并将相应信息存在对应变量中；

unsigned char\*\* convert()

将图像数据转换成二维数组,返回二维数组名。

BYTE\*\* Gray2Binary(BYTE\*\* gr,int threshold)

选取阈值将灰度图转换成二值图像，返回二值图像二维数组名。

bool writeBmp(char \*fileName,unsigned char \*\*GrayMatrix)

将二维数组数据转换成bmp要求格式，并输出新的bmp文件。

void freeMatrix(unsigned char \*\*Matrix)

释放动态分配的内存。

void work()

需要补充的功能，可以直接在work()中完成，不需要修改程序整个框架，只需要修改相应的矩阵数值，最后输出即可。

注：

1. 文件名在输入的时候，需要输入绝对路径，如：F:\123.bmp；如果文件本身在项目文件夹内，只需要输入123.bmp。
2. 在变量赋值之后，可以直接访问bmpInfoHeader.biHeight获得矩阵高度信息；

bmpInfoHeader.biWidth获得矩阵宽度信息。二维矩阵大小为：M[高度][宽度]，M[0][0]对应图片最左上角的点。本次实验所用图片矩阵均为是1944\*2592。unsigned char \*\* 和BYTE\*\*类型数据可以当做数组名用。

1. 文档中提到的参数具体数值，视实际图片而定。代码在vs2008环境下完成，不同环境可能有些地方需要稍加修改。
2. 如果有不明白的地方，请发邮件至ypsong@sei.xjtu.edu.cn。