**遥感程序设计实验记录**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **时间** | **模块（函数）名称** | **内容简要描述** | **完成情况说明** | **工作量** | **主要存在问题** | **处理结果** | **备注** |
| 2019.10.12 | 平台数据结构设计 | 设计三个结构体，分别是点、线和遥感图像，并且设计对应的类，在类中添加对结构体操作的函数申明 | 成功设置了存放点、线和遥感数据信息的三个结构体，但是将对每一种数据的操作封装成类需要之后慢慢实现 | 2h | 点和线的结构体设计较为简单没有什么问题，但是遥感数据的结构体中需要考虑较多的变量用来存放遥感数据的信息，参照老师的结构体虽然设置好了，但是对于其中一些变量的用途还尚不清除 | 暂时无法解决，之后添加各种处理功能后会慢慢弄懂 |  |
| 2019.10.15 | 平台窗体、菜单设计 | 设计简单的图形界面，有菜单栏，菜单栏上对应相应要实现的功能，其中有些点击后要弹出新的对话框，一般设计为非模态，平台为VS2012 | 创建了MFCRS工程，有一个主对话框和7个子对话框，另外完成了菜单栏和对应的功能，点击对应的功能可以弹出子对话框，且为非模态，既可以与主对话框或者其他子对话框切换。向每一个对话框添加了一个类，从而产生了这个类的头文件和cpp文件。对该对话框的操作，只需在该类中添加成员函数或成员变量。 | 10h | 第一次接触VS和MFC，故了解其基本用法需要一定的时间。  另外初步了解如何添加事件处理程序，以及如何向自己新添加的对话框添加类。  如何适当地include头文件以及头文件的标准写法，由于长时间未接触这些东西后遗忘了，导致编写过程中出现了很多error | 经过上网查阅资料，回顾c++的一些用法，以及经过多次的调试，最终解决了这些问题 |  |
| 2019.10.20 | 界面窗口点输入 | 鼠标左击按下，判断要进行什么操作，设置MouseDraw的值，表明是插入点的模式，同时还要判断是否处于编辑状态。在点击处显示点的符号。鼠标松开后，弹出对话框，显示点的XY坐标，相对于对话框的逻辑坐标，并提供一个输入点注记的编辑框。输入注记后确定，关闭该弹出框，并在点的右边显示注记。 | 完成了点输入的所有要求。包括进入插入模式，鼠标左击画点，松开后弹出对话框添加注记。除此之外，可以将画的多个点的信息保存到文件中，类型为mshp。而且重新打开后可以将记录的所有点重新画出来，包括注记。 | 6h | 一开始不知道如何实现在对话框的客户区画图这一功能。  不知道如何在不同对话框类中间传递消息。  重新打开mshp文件后没有复原记录的点。 | 在网上百度时找到了微软官方对于MFC中每个类的详细解释，包括每个类的成员函数和成员变量的具体含义和使用方法。知道了应该使用CDC这个类来作画，并了解了用这个类的哪些成员函数来画图。  不 同对话框之间传递信息可以直接访问对话框的类对象的成员变量进行赋值。  发现文件保存时格式不正确，保存与读取的方式不统一 |  |
| 2019.10.27 | 界面窗口线输入 | 首先要设置处于编辑状态，再设置处于插入线的状态中。点击第一个点时（down和up），画出第一个点，随后移动鼠标时，会从上一个点到当前捕捉到的鼠标点画一条线，此处用到了橡皮条技术，即XOR模式，在同一个地方画两处则使其变为白色底色。当点击右键时，会弹出右键菜单，可选择继续画线或结束，若选择结束则弹出新的对话框，可以设置保留的画的线的点个数。 | 成功实现了画线要求的实现功能。  将点和线的信息保存在同一个文件中，读取文件时，可以将记录的所有点和所有线全部加载并画出显示它们 | 12h | 画线需要用到橡皮条技术，一开始不知道使用这个技术，故自己琢磨如何在鼠标移动过程中保持只有一条线跟着动。一开始的想法是在原来的旧线上画一条白色的线，但是这样子会把其他擦出其他有色的区域。  研究如何把点和线的信息同时保存在一个文件中，而且读取的时候也能全部画出来，一开始想不通，而且尝试自己的想法时有很多error出现，对文件的存放和读取不够熟悉。 | 使用异或笔画线，在微软官方MFC库中找到了设置异或笔的函数，接下来只需要通过合适得算法就能实现。  经过很多次得调试，花了一整个晚上，最终完善了mshp文件存储的格式。文件一开始放long类型大小的变量代表该文件是只存储了点还是只存储了线还是两者都存有。若是两者都有，先存储点，再存储线的信息。存储的点和线数据开头前，有一个int类型的变量存放之后所有点信息数据的大小，线类似。 |  |
| 2019.11.2 | 遥感数据输入 | 首先打开000文件，读取000文件中关于遥感数据的信息，并以此初始化遥感数据图像结构体对象mrsimg的部分信息，并开辟内存空间，再读取相应的img文件到已开辟的空间。 | 参照老师的代码，完成了打开000文件并读取遥感数据的参数的功能，根据这些参数开辟内存空间，并将img文件中的遥感数据加载到内存空间中 | 8h | 开辟内存空间相关的函数比较好理解，但是读取000文件中的内容不是很懂，特别是函数SsToStr60，还有如何读取玩000文件中的所有信息并存储到遥感数据结构的对象中。 因为对结构体中一些参数仍然不够了解，故理解读取000文件这个函数比较困难，而且运行过程中遇到了很多程序中断或者没反应的现象。 | 经过大量时间，理解了SsToStr60的函数是将一整行的字符串打断成多个有效字符串。通过自己修改和调试，最终使该函数能实现想要的功能。过程中出现很多中断，其实是访问了未申请内存空间或者是数组越界，发现是一些指针未申请内存就调用，或者循环次数有误而发生数组越界的情况，经过漫长的调试最终解决了这些bug |  |
| 2019.11.9 | 遥感数据显示 | 根据用户设置的显示条件，比如长和宽、起始未知、彩色还是灰色等，初始化结构体对象中显示相关的参数，并开辟显示用的内存空间，并将原始遥感数据处理后读入到显示内存中，利用window自带的位图显示函数进行显示  为了消除闪屏想象运用了双缓冲技术 | 完成了根据输入的高宽、初始的左上角位置、彩色还是灰度设定显示图像的功能。  创建了兼容的hdc，完成了双缓冲显示图像的功能（后期完善的功能） | 10h | 一开始不知道重采样这一概念，既显示内存和遥感数据内存之间的关系，在理解显示图像相关的函数上有很大困难。  一开始不知道用哪个函数显示位图，一直在网上查，结果很多都没有用，因为提前做这一块内容，不知道用strechDIBits函数实现。但是对于这个函数里的参数要求不明白 | 经过长时间的研究，弄懂了显示相关的函数作用。因为显示屏幕或者区域的大小限制，庞大的遥感数据图像很多时候不需要所有都拿来显示，这需要重采样，开辟显示用的内存，将重采样的结果放到显示内存中。另外显示的函数需要一些与位图相关变量的初始化。 |  |
| 2019.11.13 | 遥感图像统计分析 | 对 读入的遥感数据进行直方图统计，即对每个波段的每个值得频数进行统计，并将结果以直方图得形式在新的非模态对话框中进行显示，可以在picture控件中，从而可以通过鼠标点击或者编辑框，选择显示的数值的最小值和最大值，并使对应的显示图像也随之变化 | 完成了要求的所有功能，而且对于手工调整直方图，设置了两种模式。 | 12h | 在直方图统计的子对话框上，选择使用picture控件显示，但是这个控件的消息函数只有鼠标点击事件，没有区分右键还是左键。为了特别上网查阅获取鼠标事件的函数，但可能MFC比较古老用的人也不多，没有查阅到正确的函数。另外，不知道怎么在picture控件上画图显示。 | 经过资料查阅，发现可以获取picture控件的hdc，从而可以像在主对话框中一样画图。可以通过添加控件变量，从而获取picture的hdc  因为无法判断在picture控件中点击的左键还是右键，所以只能通过一个checkbox判断当前点击是改变最小值还是最大值。同时该控件的鼠标点击函数并没有点击的坐标信息，故需要自己获取再将其转为相对控件的相对坐标 |  |
| 2019.11.20 | 遥感数据二维散点图 | 横 纵坐标分别是选择的波段的0-255数值。代表的意义是图上一个点对应的两个选择波段的数值，若是黑色则无法表示密度，若是彩色，以不同颜色代表密度。 | 完成了显示二维散点图的所有功能，包括不填色或者填色的散点图、读取填色用的颜色库 | 5h | 下拉框第一次使用不知道该如何调用  二维散点图的函数一开始看不懂一些过程的意义是什么，特别是填色后的显示 | 通过上网查阅资料，看别人的代码和注释，最终学会了如何使用下拉框  参照老师的代码以及自己的多次调试，最终理解了每一句代码的含义和作用 |  |
| 2019.11.27 | 遥感数据添加噪声 | 椒盐噪声。利用C语言自带的随机数生成函数，选择随机的图像点变为0或者255，即黑色或白色。  高斯噪音。即每个图像点的数据加上高斯噪音值。高斯噪音的值由自己写的函数进行生成 | 完成了对遥感数据添加椒盐噪声和高斯噪声，不管是彩色的RGB图还是一个波段的全色图。 | 5h | 椒盐噪声比较容易，但是为彩色图像添加椒盐噪声的时候，只响一个波段添加了噪声，导致显示的噪声为红色  高斯噪声是自己了解到并想添加的一种噪声而类型，但是对于高斯噪声产生的原理不清楚 | 对于RGB彩色图，同时向三个波段添加同一位置的椒盐噪声即可。  对于高斯噪声，对于生成复合高斯分布的噪声值的原理仍然不懂，这其中涉及到了数学原理，可以不去理解。但是如何获取了这个噪声值后如何对每一个遥感数据添加的原理和方法都已掌握 |  |
| 2019.12.1 | 图像平滑、卷积、中值处理 | 选取3\*3的数据，其中平滑和卷积有8种不同的权重系数矩阵，中间的像元值由3\*3的数据矩阵和3\*3的系数权重矩阵计算后得到。而中值滤波则是选区3\*3区域中像元值的中值当作其像元值。 | 完成了老师要求的平滑滤波和卷积滤波，同时还完成了老师额外的要求中值滤波 | 3h | 该模块代码实现很简单，经过长时间的接触，已经了解了VS和MFC的一些基本使用方法，该模块没有什么编程上的困难。  对于卷积滤波的原理仍然不是很理解，因为没有上过相关的数学课 | 查阅资料发现卷积函数的原理还是很复杂的，需要数学知识的支撑，本次实验不需要掌握太深层次的卷积函数的相关知识 |  |
| 2019.12.8 | 小波变化 | 包括小波正变换（分解）和逆变化（合成）。小波分解就是将数据压缩，分为低频的部分和高频的部分。而小波合成则是将低频部分和高频部分合成还原为原来分辨率的图像。 | 完成了小波正逆变换的功能，并且用户可以选择小波变换的级数 | 5h | 本次实验采用了最简单的小波： Haar小波。对于用该小波进行的小波变化，已明白弄懂。但是对于一些复杂的小波，还无法理解。主要是没有学过傅里叶变换相关的知识，虽然学过傅里叶级数，但是两者之间差异还是很大的，数学知识掌握不够好。  在小波变化过程中，会有部分的图像信息的损失，有些是可以避免的，有些是无法避免的，处理这一点的某些代码比较难以理解，需要花费思考其含义。 | 关于傅里叶变换的数学知识，暂且不去管太多，但理解透彻了本次实验的小波变换原理。  其次高频部分因为会产生负数，需要存储负数的信息，在这一点特别做了处理，否则逆变换会无法复原。 |  |
| 2019.12.9 | 图像增强处理——缨帽分析 | 按要求对遥感数据完成缨帽变换，并输出四个结果文件 | 完成缨帽变换的功能，可以输出四个文件 | 2h | 原理和实现代码都比较简单，没有遇到什么问题，而且这是一种经验变换 |  |  |
|  | 遥感数据图像编码——行程编码 | 是一种栅格数据压缩的编码方法。在各行数据的代码发生变化时依次记录该代码以及相同代码重复的个数。输出源数据的txt格式和以行程编码形式压缩的cod文件，最后添加压缩率信息 | 完成了压缩的功能，输出相应的txt和cod文件 | 5h | 行程编码的原理十分简单，代码实现也很简单，但是运行结果却出不来。 | 经过调试发现最后在文件中输出压缩相关信息时出错，经过排出，发现在文件中输出%会引发错误，原理不可得知，可能该函数不够完善，故不输出%，结果运行正常。 |  |
| 2019.12.15 | 遥感数据几何变化 | 旋转：左旋90°、右旋90°。注意的是对原始的遥感数据进行变换  镜像：X、Y、原点对称  平移 该操作结合鼠标点击和移动事件，改变遥感数据结构体中的Img\_X和Img\_Y从而实现。  放大或者缩小：一开始使用的方法时重新分配显示的内存和设置显示的大小 | 完成了所有要求的几何变化，包括用鼠标拖动图像任意移动和对图像的放缩 | 10h | 这一模块自己提前开做，在编左旋和右旋的时候，编写后运行是成功的，但是对图像进行放大或者缩小后，之前的几何变换效果就没了  实现鼠标拖动图片平移时，并没有那么简单，需要两对坐标来记录平移过程中的一些中间变量，一开始没有设置好，图片的移动路径和鼠标的移动路径完全不同。或者进行过图片缩放后，图片移动第一下就会出现问题。  实现图片的放大或缩小，认为就是显示图像内存的改变，既重采样分配。但是由于内存分配是否正确是看不见的，最终只能看显示的图像是否正确。  另外，虽然实现了鼠标拖动图片移动的功能，但是发现移动的时候刷屏现象很严重，即图像一闪一闪的，很不友好和美观。 | 参照老师给出的几何变换代码，发现对图像的几何变换，是对其真实的遥感数据的变换，而不是对其显示内存数据的改变。  为了实现鼠标拖动，需要设置图片停止时的坐标和鼠标点击处的坐标以及鼠标移动事件传回来的鼠标坐标。 | 虽然实现了图像的缩放，但是一开始的缩放发现是图像的左上角位置不变的缩放，这样子对用户不友好，经过一番调试和改良，实现了在鼠标指针停放处的图像缩放功能。即对感兴趣区域的缩放。  还有另一点是，但图片放大，其大小唱过显示的客户区时，我们仍然对整个图像进行重采样，但其实不需要，只需要对显示屏显示区域的图像数据进行重采样即可。这一功能其实与另外一个功能相联系；即用鼠标画出矩形框，客户区放大或缩小该区域的图像。但是这种功能因为时间紧迫，没有将其实现，会在寒假中摸索将其实现！ |
| 2019.12.22 | 遥感图像输出 | 拼接：将两张宽相同的bmp图像上下拼接保存得到一张bmp图像。  裁剪：对于一张bmp，设置裁剪的高度，以其为裁剪线，横向裁剪成两张bmp图像并保存。 | 完成了老师要求的图像拼接功能，还额外完成了图像裁剪的功能。 | 10h | 这个功能实现的主要难点在于读取各种有关文件，特别时对文件名的处理  还有对bmp文件一些变量的初始化或者设置  其原理较好理解，没有太大的问题，但还是需要一些时间去熟悉、掌握  完全不知道jpg格式的存放，因为此功能的局限是对jpg格式图片的操作，一开始去了jpg格式的官网下载并安装了静态链接库，但是发现自己编写十分复杂，根本无法获取jpg格式中的图像数据，甚至将其转换为bmp格式都做不到 | 经过查阅资料和请教老师，发现短时间内完全弄懂jpg格式图像是不现实的，而且静态链接库的调用存在莫名的bug，有些函数无法生效，故放弃该方法，上网查阅发现在MFC中使用CImage类可以轻松完成jpg格式的文件转换为bmp，或者将bmp格式的文件转换为jpg格式。   * 对于如何将数据放到bmp文件中，经过对bmp格式的深度理解和反复看代码后理解了，没有问题。 |  |
| 2019.12.24 | 屏幕输出 | 用户输入屏幕输出的宽和高，随后输出以客户区的左上角开始的矩形内的显示内容，即完成“所见即所得” | 完成该功能，输出文件的格式为bmp | 8h | 一开始没有思路如何获取屏幕显示的内容，因为老师提供的参考代码是基于C++Builder的，一些函数在VS中没有或者因为名字不同一开始没有找到，故上网查阅资料，但发现有很多的方法，但是是不同的版本，因为VS有很多版本，而且很多都是老版本的方法或者函数。   * 没有充分理解bmp格式，不知道如何将内存中的数据保存到bmp格式的文件中。 | 在网上查了很多方法，因为不知道哪种方法正确或者适合VS2012中使用，故一一尝试看能否实现目的，经过长时间的代码修改和调试，最终完成了该功能。  再一次复习了bmp格式，深度理解了其数据结构。 |  |
| 2019.12.30 | 遥感数据的转换 | 实现bmp格式的遥感数据与img+000的遥感数据格式之间的相互转换。 | 实现了bmp与img格式之间的相互转换 | 5h | 在充分了解bmp格式的前提下，以及之前打开000文件时，对用000+img格式存储遥感数据的原理和方法都熟悉，故功能的实现没有什么问题，虽然代码写得比较繁琐，主要涉及到了文件名以及文件的打开，bmp格式参数的设置等等，但是总体坐下来还是很顺利的 | 没有问题 |  |
| 2019.1.1 | 遥感数据保存 | 将内存中的遥感数据保存为bmp、img或者jpg的格式，可以选择对原文件的保存，也可以选择另存为新的文件 | 实现了该模块所有要求的功能 | 10h | 将遥感数据保存为bmp或者img格式，这当中的操作其实和上一模块两者的相互转化很相似，故保存为bmp和img格式没有遇到困难。  保存为jpg格式也很简单，因为前面模块已经设计到bmp和jpg格式之间的相互转换，都是调用MFC现成的转换函数 | 没有困难 |  |
| 2019.1.5 | 界面优化 | 颠倒输入  直方图统计自动拉伸  读入bmp格式的遥感数据  两图相加 | 全部完成界面优化的功能 | 10h | 经过一学期的编程，对MFC编程有了充分的了解，故这一模块没有特别大的困难，而且是功能优化，对这些功能都已经有过了解 | 没有困难，回顾了一下之前写得代码 | 在对界面进行优化的时候，重新看了以前写过的代码，发现有很多都可以改进，有一些代码语句不需要，而且有些地方还存在bug。故对整个功能做了一次完善、优化，使得代码看起来更清晰、有条理逻辑性 |
| 2019.1.11 | 图像显示优化 | 之前虽然使用双缓冲技术实现了拖动图片时不闪屏，但是发现如果有点和线时，拖动图片后马上它们马上就没有了，这一模块是实现的功能是，移动图片或者刷新生成图片时不影像点和线 | 完成想要的功能，相当于分了两个图层，一层是图片，一层是点和线，其中点和线位于图片上方。在这里，充分了解了刷屏重画函数OnPaint()的功能和价值。 | 8h | 在编程过程中，一开始没有想到OnPaint函数的重要性，以前一直不理解老师说的每次画完线都刷屏重画这一含义，因为自己似乎也没有这么做也很成功，只不过是没有遇到出现bug的情况罢了  只考虑了使用技术绘制图片使其不闪屏，发现在移动图片的时候线和点也在疯狂移动 | 经过多次尝试和编写代码，最终发现，不论显示什么，都最好将绘制的函数放在OnPaint函数当中，因为真正的软件可能加载了很多图层，而刷屏其实是很常见的调用。例如在本软件中，在OnPaint()中放了显示图片、显示点和线的函数，然后再结合双缓冲技术，画任意东西的函数传进去的hdc都是当前设备的兼容hdc，当所有该画的都在兼容的hdc画幕中画完后，然后在OnPaint函数用Bitblt拷贝到当前的画幕中，从而图片和线都不会刷屏。 | 在这里我想补充一点，在很多软件中，如envi和arcgis中可以加载很多个图层，而且可以选择哪个图层在最上面，如将线矢量文件放在最上面，接下来一层是栅格图片。根据这一模块的实验，我猜想这种功能实现的原理就是改变图层画的顺序。不过这一功能并不太好完美实现，至少目前自己的能力还不行。 |