**2 Android平台下C/S编程框架**

**2.1 Client/Server架构简介及其在野外工作中的优势**

计算机技术以及传感器网络的发展，深刻改变了传统的数据采集方法。尤其是生态学中的野外地质观测研究正逐渐朝着自动化、网络化、智能化、常态化的方向发展，是现代地质学科研信息化的重要组成部分。但是长期以来，不同种类野外采集系统间的异构性给数据的统一处理、存储造成了困难，不便于互联互通、共享展示。适应不同系统的统一数据采集平台成为当前野外数据采集系统的重要需求之一。针对野外地勘工作对于数据采集现场性和准确性的要求，以及提高数据处理和团队信息交流效率的迫切需求，而跨平台的Client/Server架构远距离数据交互能力尤为强大，因而在互联网+地质背景下，本文选择适合远距离交互的Client/Server架构来完成开发。

Client/Server架构是在网络连接模式中,除对等网外的另一种形式的网络,在本文开发的地勘系统客户机/服务器网络中，服务器是网络的核心，而野外移动端客户机是网络的基础，客户机依靠服务器获得所需要的网络资源，而服务器为客户机提供必须的网络资源和地质数据处理能力。

　　这里野外移动端客户和室内服务器都是指通信中所涉及的两个应用进程（软件）。使用计算机的地质工作者是计算机的“用户”（User）而不是“客户”（Client）。

它是软件系统体系结构，通过它可以充分利用两端硬件环境的优势，将地质勘探记录和处理任务合理分配到 Client端和Server端来实现，降低了野外工作时间开销，提高了地质工作效率。目前大多数应用软件系统都是Client/Server形式的两层结构，由于现在的软件应用系统正在向分布式的Web应用发展，Web和Client/Server 应用都可以进行同样的业务处理，应用不同的模块共享逻辑组件；因此，内部的和外部的用户都可以访问新的和现有的应用系统，通过现有应用系统中的逻辑可以扩展出新的应用系统。这也就是目前应用系统的发展方向。

C/S结构在地质工作中的优点：

　　1.C/S结构的优点之一是能充分发挥移动客户端的便携性，相较于人工笔记，应用移动端软件记录无论是存储介质的持久性还是记录的灵活性都有极大的提高，更不用说数据的上传、处理与交流效率。例如在本次开发的地勘系统中，每当地质工作人员打开测点信息记录或修改界面，后台接口会自动回调并定位，记录下当前测点坐标经纬度，而无需手动GPS定位和记录，大大提高了记录的效率和准确性。很多工作可以在客户端处理后再提交给服务器。对应的优点就是客户端响应速度快。

2. 不同于以往地质勘探数据记录在纸质介质或设备中，只有带回营地后方可进行室内处理，但当牵扯路程遥远或发生意外难免延缓工程进度。在本次开发的地勘系统中，面对小规模数据处理或无网络连接，移动设备可以进行数据离线存储和处理，而当户外存在网络连接，依靠C/S架构下移动客户端的联网能力，野外地质勘探数据可以实时上传至室内服务器，凭借巨型机强大处理能力进行海量数据运算并实时反馈处理结果，方便地质工作者及时查看成果图并进行解释工作，极大的提高了户外工作效率。

3. 传统地质勘探工作中，由于空间限制，各个地质工作队伍不能及时交流数据，从而决定下一步工作方向，大大延缓了工程效率。在本次开发的地勘系统中，通过团队之间共享账号，服务器端对应账号数据库中测区数据会同步下载至每一台移动设备客户端中，从而使工作人员可以总揽全局，以更大的地质尺度、更高的准确率进行解释工作。

4. 服务器端依靠数据库进行测区信息记录，这样现代化的地质数据记录方式有着传统记录方式不可比拟的优势，相较于纸质记录介质存储、老化和查找冗余复杂，数据库中信息可以进行海量数据存储和高效率跨平台转移、维护、处理。

**图1 传统户外地质数据人工记录**

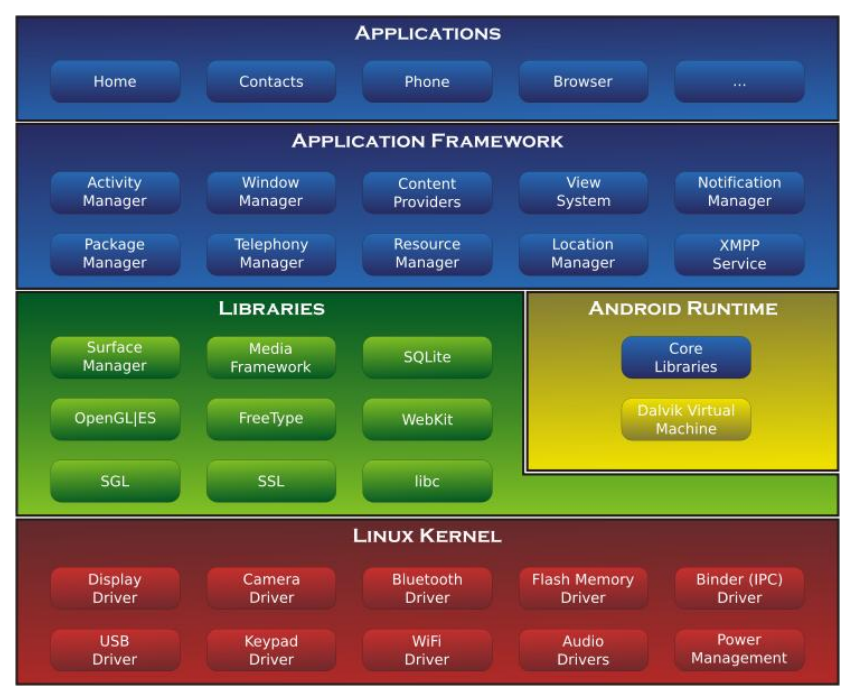
**图2 利用C/S架构地勘系统进行记录**

**2.2 Android开发简介**

**2.2.1 Android OS简介**

Android是一种基于Linux的自由及开放源代码的操作系统，主要使用于移动设备，如智能手机和平板电脑，由Google公司和开放手机联盟领导及开发。尚未有统一中文名称，中国大陆地区较多人使用“安卓”或“安致”。Android操作系统最初由Andy Rubin开发，主要支持手机。经过数年的发展与完善， Android系统已被广泛应用智能手机、智能电视及可穿戴设备，已经成为大众极为熟悉的日常操作系统。

Android系统构架是安卓系统的体系结构，android的系统架构和其操作系统一样，采用了分层的架构，共分为四层，从高到低分别是Android应用层，Android应用框架层，Android系统运行层和Linux内核层。Android系统构架主要应用于ARM平台，但不仅限于ARM，通过编译控制，在X86、MAC等体系结构的机器上同样可以运行。



**图3 层次化的Android系统结构**

**·Linux内核层：**Android系统是基于Linux 2.6内核的，这一层为Android设备的各种硬件提供了底层的驱动，如显示驱动、音频驱动、照相机驱动、蓝牙驱动、WiFi驱动、电源管理等。

**·系统运行层：**这一层通过一些C/C++库来为Android系统提供了主要的特性支持，比如SQLite库提供了数据库的支持，OpenGL ES库提供了3D绘图的支持，Webkit库提供浏览器内核的支持等。同时，在这一层还有Android运行时库，它提供了一些核心库，能允许开发者使用Java来编写Android应用。

·**应用框架层：**这一层主要提供了构建应用时可能用到的API，Android自带的一些核心应用程序就是使用这些API完成的，开发者可以通过使用这些API构建自己的应用程序。比如有活动管理器、View系统。内容提供器、通知管理器等。

**·应用层：**所有安装在手机上的应用程序都是属于这一层的，比如系统自带的联系人、短信等程序，或者我们从Google Play上下载的程序，当然也包括本文中开发的基于C/S架构的地质勘探软件。

**2.2.2 Android基础组件**

**1.Acivity/活动：**在Android系统中，Activity是所有程序的根本，所有程序的流程都运行在Activity 之中，Activity可以算是开发者遇到的最频繁，也是Android 当中最基本的模块之一。在Android的程序当中，Activity 一般代表手机屏幕的一屏。Activity 概念和网页的概念相当类似，每个Activity中含有多个控件。一般一个Android 应用是由多个Activity 组成的。当打开一个新的Activity时，之前一个Activity会被置为暂停状态，并且压入历史堆栈中。用户可以通过回退操作返回到以前打开过的Activity。可以选择性的移除一些没有必要保留的Activity，因为Android会把每个应用的开始到当前的每个屏幕保存在堆栈中。

**2.Service/服务：**是android 系统中的一种组件，它跟Activity 的级别差不多，区别是其不能单独运行，只能在后台运行，并且可以和其他组件进行交互。Service 是没有界面的长生命周期的代码。Service是一种守护程序，它可以长时间运行，但是它始终没有可视化用户界面。

**3.Broadcast/广播接收器：**在Android 中，Broadcast是一种广泛运用的在应用程序之间传输信息的机制。而BroadcastReceiver 是对发送出来的Broadcast进行过滤接受并响应的一类组件。可以使用BroadcastReceiver 来让应用对一个外部的事件做出响应。BroadcastReceiver不能生成UI，也就是说对于用户来说不是透明的，用户是看不到的。BroadcastReceiver通过NotificationManager 来通知用户这些事情发生了。各种应用还可以通过使用Android提供的接口将它们自己的Intent Broadcasts广播给其他应用程序。

**4. Content Provider/内容提供者：**Content Provider 是Android提供的第三方应用数据的访问方案。在Android中，对数据的保护非常严格，除了存放在SD卡中的数据，一个应用所持有的数据库、文件等内容，都是不允许其他直接访问的。如果应用想对外提供的数据，可以通过派生Content Provider类， 封装成一枚Content Provider，每个Content Provider都用一个uri作为独立的标识，实现应用间数据交互。

**2．2．3 Android移动端开发流程**

**设计实施阶段**

Android开发环境搭建

安装JAVA SE

JDK

模拟器AVD搭建

测试

创建JAVA工程

将动态链接库添加到工程，运行JAVA程序

编译测试

单元测试

Android工程调试

新软件调优/新机型适配

无法达成预期

验收合格/打包发布

√

集成/功能/系统测试

模拟机/真机调试运行

AVD创建及配置

IDE中配置ADT+SDK

安装Android sdk +IDE/ecplise

配置JAVA环境变量

产品验收

**项目测试阶段**

**项目验收阶段**

**2.3 地勘系统服务端进程介绍**

在本次尝试开发的地勘系统中，为了持续接受并处理移动客户端请求，因而在服务端设立守护进程(daemon)。守护进程是运行在后台的一种特殊进程。它独立于控制终端并且周期性地执行某种任务或等待处理某些发生的事件。它不需要用户输入就能运行而且提供某种服务，不是对整个系统就是对某个用户程序提供服务。（注意：这里所说的无交互并不是说真的不能和这类进程打交道，不能控制其运行，而是说不能通过传统的终端用类似shell的交互方式控制其运行）。  
 守护进程一般在系统启动时开始运行，除非强行终止，否则直到系统关机都保持运行。守护进程经常以超级用户（root）权限运行，因为它们要使用特殊的端口（1-1024）或访问某些特殊的资源。  
 特别对于本次开发的地质勘探系统，针对项目提出的用户验证，测区信息存储，重力测值处理等需求，由移动客户端创建封装TCP/IP协议的套接字Socket发起访问请求，在服务端设计相应的接口，在特定端口调用ServerSocket进行监听，对于不同类型的请求实行相应的解决方法，充分发挥移动客户端的便携灵活性和服务端强大的数据存储、处理能力，大大提高地质工作者户外工作效率。

图4 地勘系统服务进程逻辑流程

Log\_in(NewUser nuser){

···}

updateDisInf(DisInformation n){

···}

Sign\_in(User user){

···}

processGri(DisInformation n){

···}

注册请求

登录请求

测区信息存储请求

重力测值处理请求

···

地勘系统服务器

地勘系统数据库

otherMethods(argus ){

···}

**2.4 OpenGL ES简介**

OpenGL ES (OpenGL for Embedded Systems) 是[OpenGL](https://baike.baidu.com/item/OpenGL" \t "_blank)三维图形 API 的子集，针对手机、PDA和游戏主机等[嵌入式设备](https://baike.baidu.com/item/%E5%B5%8C%E5%85%A5%E5%BC%8F%E8%AE%BE%E5%A4%87" \t "_blank)而设计。该API由Khronos集团定义推广，Khronos是一个图形软硬件行业协会，该协会主要关注图形和多媒体方面的开放标准。

OpenGL- ES 是免授权费的，跨平台的，功能完善的2D和3D图形应用程序接口API，主要针对多种嵌入式系统专门设计 - 包括控制台、移动电话、手持设备、家电设备和汽车。它由精心定义的桌面OpenGL子集组成，创造了软件与图形加速间灵活强大的底层交互接口。 OpenGL ES 包含浮点运算和定点运算系统描述以及 EGL针对便携设备的本地视窗系统规范。OpenGL ES 1.X 面向功能固定的硬件所设计并提供加速支持、图形质量及性能标准。OpenGL ES 2.X 则提供包括遮盖器技术在内的全可编程3D图形算法。OpenGL ES-SC 专为有高安全性需求的特殊市场精心打造。

在本次尝试开发的地勘系统中，重力异常的移动端可视化也是依赖于百度地图组件开放的OpenGl ES接口实现的，应用底层绘图库，使得高效绘制海量测点信息成为可能。截至目前，OpenGL ES已经更新版本至3.0，但多集中于渲染管线多重增强、纹理功能增强和增加渲染缓冲格式，本次地勘系统重力异常开发仍然使用2.0版本进行实现。

|  |  |
| --- | --- |
| 原始作者 | Khronos Group |
| 开发者 | Khronos Group |
| 稳定版本 | 2.0/ 2007年3月5日 |
| 编译语言 | C语言(在Android中通过NDK编译JAVA实现) |
| 操作系统 | Cross-platform  跨平台 |
| 类型 | API  接口 |
| 支持平台/机型 | - 支持iPad, iPhone3GS 和后续版本，以及iPodTouch3代和后续版本。  - 支持Android平台从Android 2.2版本开始。  - 支持Android NDK从Android 2.0版本开始。  - 支持BlackBerryPlayBook黑莓。  - 支持Pandora潘多拉控制台的3D库。  - 被WebGL支持：浏览器支持OpenGL  - 支持少数新款Nokia诺基亚手机，比如N900上的Maemo和N8上的Symbian3塞班3系统。  - 支持多款三星手机，包括Galaxy S和Wave。  - 使用开发插件可以支持Palm webOS。  - 支持Archos 爱可视上网本：70 IT, 101 IT。 |

表1 OpenGL ES 2.0 概况

**3重力勘探离散测点二维插值加密**

**3.1 插值地质勘探中的应用**

插值是对原勘探数据的重新分布，从而来改变测点数量的一种方法。在数据矩阵尺寸放大过程中，数据量也相应地增加，增加的过程就是“插值”发生作用的过程，“插值”程序自动选择信息较好的测值作为增加、弥补空白测点的空间，而并非只使用临近的测点信息，所以在放大矩阵时，图像看上去会比较平滑、干净。

考虑地勘系统所采集的重力勘探测值的数据结构，在经过自由空间改正、中间层改正、地形改正和均衡改正后，每个被经纬度确定的坐标点对应一个重力测值，初步考虑应该使用二维插值，二维插值方法包括最临近插值发、分片线性画质、双线性插值、样条插值等一系列类似的方法，常用于图像处理、温度场计算、气压等压线和地图等高线的绘制、计算机模拟和数字信号处理等多种方面。但以上常用于数据呈规则网格节点，在野外实际地质工作中，实际测得测区内测点排布常常不满足条件，呈离散测分布，因而尝试使用反距离加权插值算法实现。

反距离加权插值(Inverse Distance Weighted)，也可以称为距离倒数乘方法。是指距离倒数乘方格网化方法是一个加权平均插值法，可以进行确切的或者圆滑的方式插值。方次参数控制着权系数如何随着离开一个格网结点距离的增加而下降。对于一个较大的方次，较近的数据点被给定一个较高的权重份额，对于一个较小的方次，权重比较均匀地分配给各数据点。反距离权重法主要依赖于反距离的幂值，幂参数可基于距输出点的距离来控制已知点对内插值的影响。幂参数是一个正实数，默认值为2。（一般0.5到3的值可获得最合理的结果）。

**3.2 测区反距离加权插值算法实现**

在进行IDW插值之前，为了后续等值线构造曲线更加平滑，需要对目标插值矩阵进行尺度初始化，因而我们设计函数接口传入希望得到的测区插值尺寸参数。

**①初始化插值矩阵：**

VAR initInterp[Matrix](javascript:;)(List longitude,List latitude,List value,int row,int col,int n)

参数列表：

|  |  |  |
| --- | --- | --- |
| 变量名 | 类型 | 释义 |
| Longitude | List | 测区内测点经度值 |
| Latitude | List | 测区内测点纬度值 |
| Value | List | 测区内测点重力测值 |
| Row | int | 目标插值矩阵行数 |
| Col | int | 目标插值矩阵列数 |

首先对传入的测区内测点经纬度参数进行起泡排序，获得经纬度最大/小值和对应的测点，如下图测点ABCD，利用经纬度最大/小值进一步可以构造一个矩形区域进行插值。很明显，这个矩形区域在数据结构上对应着一个二维数组，下一步的目标即为对此矩阵初始化，在确定插值矩阵之后，根据传入的插值矩阵(二维数组)尺度，实例化二维数组Interp\_value，一维数组Interp\_longitude和Interp\_latitude，其中二维数组用来保存即将得到的区域测点插值，一维数组用来保存插值矩阵每一行/列对应经纬度。

STEP1 实际户外工作得到的测区离散测点分布图

STEP2 利用测区最大/小经纬度确定插值矩形区域

STEP3 根据目标矩阵尺寸参数构造插值矩阵

**图5 initInterp**[**Matrix**](javascript:;)**(argus)方法处理流程**

在调用方法之后，返回数据结构Var，其中包含:

|  |  |  |
| --- | --- | --- |
| 变量名 | 类型 | 释义 |
| Interp\_longitude | Float [ ] | 插值后测区内测点经度列表 |
| Interp\_latitude | Float [ ] | 插值后测区内测点纬度列表 |
| Interp\_value | Float [ ][ ] | 矩形测区插值矩阵(未赋值) |

**②IDW插值：**  
 void IDW(Var var)

|  |  |  |
| --- | --- | --- |
| 变量名 | 类型 | 释义 |
| Var | 自定义类 | 经插值得到的测区矩阵信息 |

IDW插值方法假定每个输入点都有着局部影响，这种影响随着距离的增加而减弱。

步骤：(1)计算目标插值测点与实际测点距离：

以上公式计算了二维插值矩阵中第i行j列位置上的插值重力测点与第k个实际户外工作测点之间的距离。

以上公式计算了二维插值矩阵中第i行j列位置上的插值重力测点与全部K个实际户外工作测点之间距离倒数的总和。

（2）计算得到每个实际测点对目标插值测点的影响系数：

以上公式得到的系数将决定每个户外实际测点观测数据对期望测点插值结果的影响，或者说贡献程度，某测点系数越大说明实际测点离期望位置越近，插值结果更可能接近该实际测点的值。

(3)利用影响系数计算期望测点插值结果：

至此，在编程实现以上算法之后，二维矩阵得到赋值初始化，我们已经得到了经过IDW加密的测区插值矩阵，经过处理，可以使下一步等值线的构造更加平滑自然

**Unknown Pointer**

STEP1 插值矩阵未被赋值

STEP2 进行IDW插值加密

STEP3 插值矩阵已被赋值，等待构造等值线

**4 等值线构造与移动端可视化**

**4.1 重力等值线构造中三角剖分算法的应用**

等值线构造算法有很多，本文选用三角网格法实现，而三角网格构网以三角剖分为基础。三角剖分是代数拓扑学里最基本的研究方法。以曲面为例， 我们把曲面剖开成一块块碎片，要求满足下面条件：①每块碎片都是曲边三角形；②曲面上任何两个这样的曲边三角形，要么不相交，要么恰好相交于一条公共边（不能同时交两条或两条以上的边)。

在二维场问题中，涉及的区域是以Γ为边界的平面区域Ω，三角剖分是常用的形式，它适用于各种几何形状的区域和非均匀介质的情况。而在实际地质工作成果可视化过程中不仅会遇到平面点集的三角剖分，还会遇到多边形的剖分问题。这里主要介绍简单多边形的三角剖分问题。简单多边形具有以下几何特征：

(1)多边形的边界是由若干个结点顺序连接而成的闭合环，任意相邻两个结点对定义了一条有向边；

(2)任意两条有向边的交要么为多边形的边界上的一个结点，要么为空；

(3)经过多边形的边界上的任一个结点，有且仅有两条有向边。

图4(a)中的多边形不满足上述条件(1)，图4(b)不满足上述条件(2)，图4(c)中的多边形不满足上述条件(3)，图4(d)中多边形属简单多边形。

(a)

(b)

(c)

(d)

简单多边形的三角剖分问题是指将简单多边形划分成若干三角形的集合，即将简单多边形所围区域划分成二维单纯复形，而且，任意三角形的顶点均为简单多边形的边界结点。

由于简单多边形的三角剖分网格中，三角形的顶点均为简单多边形的边界结点，因此，所有三角形的边只能来自简单多边形的边与对角线。根据这个特点，我们可以采用对角线法进行简单多边形的三角剖分，算法过程如下：

(1)计算任意两非相邻结点之间的距离，即对角线长度，并存储到数组T中；

(2)根据对角线长短对T进行排序；

(3)按照对角线从短到长顺序从T中提取对角线t，并从T中删除t，如果t不与其他边相交且位于多边形内，则t定是三角剖分的一条边；

(4)如果t是三角剖分的一条边，则判断t是否构成某个三角形的边；

(5)重复执行步骤(3)直到T为空。

STEP3 插值矩阵已被赋值，等待构造等值线

STEP1 插值矩阵未被赋值

**4.2三角网格实现重力测值等值线追踪**

当经过简单多变性三角剖分算法后，整个矩形区域变成由多个三角网格构成的三角网络，接下来实现等值点的确定。一个网格是否有等值点,取决于该网格对应的空间三角形是否与某一数值的水平线相交。具体体现为该平面的数值是否介于三角形的某一边的两个顶点的数值之间。确定出全部的等值点后,有次序将它们连成等值线,这就是等值线的追踪问题。全过程分以下三步处理：

(1)确定等值线的走向。对于任意三角形单元,若棱边满足等值点存在条件,其等值线可能以3种方式通过三角形单元,如图()所示：

**2**

**j**

**1**

**k**

**3**

**i**

**2**

**j**

**1**

**k**

**3**

**i**

**2**

**j**

**1**

**k**

**3**

**i**

图4 三角网格中等值线的走向

(2)确定等值线进入网格后出去的走向。等值线进入网格后,从网格的另外两个边方向出去,可以按照这样一个次序追踪下一个等值点:即先考虑等值线原来的前进方向,后考虑当前等值点的远近。

(3)网格点为等值点的处理。有时网格点的数值与相交面数值相等,此时等值线通过网格点,而网格点同时又是3个相邻网格的公共点,同一等值点分别存放在3个不同的单元中,在追踪时一定会发生重复使用和追踪混乱的问题。故对此情况必须先处理,其方法是对网格加上一个值ε(ε→0) 给予修正。

在实现以上算法时，设计接口如下：

**void** **contour**(**float** [ ][ ] value,**int** ilb,**int** iub,**int** jlb, **int** jub, **float**[ ] x, **float**[ ] y, **int** nc, **float**[] gap\_z)

|  |  |  |
| --- | --- | --- |
| 变量名 | 类型 | 释义 |
| value | Float[][] | 已插值测区重力数据 |
| Lib | Int | 实际应用中很可能只需要绘制部分区域重力等值线，截取数据矩阵中从lib至jub行，jlb至jub列数据进行等值线构造 |
| Jub | Int |
| Jlb | Int |
| Jub | Int |
| X | Float[] | 测区经度坐标集合 |
| Y | Float[] | 测区维度做标集合 |
| Nc | Int | 等值线绘制级别(条数) |
| Gap\_z | Float[] | 每两条等值线数值间隔 |

此接口接受参数后，截取已插值重力数据矩阵目标区域进行处理，包括设置发生绘制的位置、绘制等值线的数量和每两条等值线间的数值间隔。在经过对每个三角网格进行等值点追踪后，在方法内调用绘制接口进行绘制。

在PC端调试Demo,调用JAVA graphics接口进行简单绘图演示如下：

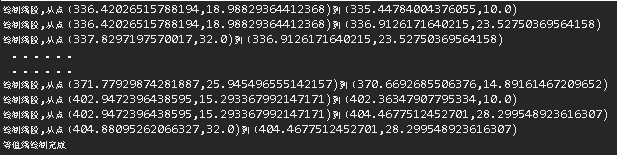


图5 控制台输出每一步绘制细节

STEP1 初步调试并观察算法效果，可以发现在精度要求不高时可以满足等值线绘制需求

SETP 2 改变函数参数，进一步增加等值线层次，提高绘制精度

在调试后修改代码，由于本文长实开发的地勘系统计划使用百度提供的地图组件实现重力异常可视化，因而修改代码使之可以适配移动端百度地图组件提供的OpenGL绘图接口，百度地图SDK为广大开发者开放了OpenGL绘制接口，帮助开发者在地图上实现更灵活的样式绘制，丰富地图使用效果体验。

自4.2.0起,地图SDK支持OpenGLES 2.0版本，每次绘制任务都会回调绘图接口onMapDrawFrame(MapStatus drawingMapStatus)，地图渲染每一帧画面过程中，以及每次需要重绘地图时（例如添加覆盖物）都会调用此接口。开发者可以在这个接口中进行opengl的绘制。自4.2.0起，旧版的onMapDrawFrame(GL10 gl, MapStatus drawingMapStatus)废弃。

在OpenGL ES中，只支持三种类型的绘制：点、直线以及三角形。所以需要在绘制图像之前，需要把一个图像分解为这三种图像的组合。

其次，OpenGL作为本地库直接运行在硬件上，没有虚拟机，也没有垃圾回收或者内存压缩。在Java层定义图像的数据需要能被OpenGL存取，因此，需要把内存从Java堆复制到本地堆。使用的方法是通过ByteBuffer:

在进行绘制操作之前，我们需要进行多项渲染设置，调用接口顺序如下：

onSurfaceCreated(GL10 unused, EGLConfig config)

Vertex Shader

顶点着色器

Fragment Shader

片元着色器

Program

包含了用来绘制形状的着色器的OpenGL ES对象

OnDraw( )

在渲染器Render中重写此接口，在其中设置顶点着色器和片元着色器的position和color值，从而实现绘制操作

绘制操作需要至少一个顶点着色器来绘制形状，一个片元着色器来对形状着色。并且这些着色器必须被编译和被添加到OpenGL ES程序中。现在已经准备好执行实际的绘图命令了，使用OpenGL ES绘图需要重写onDraw()方法，在其中指定一些参数来告诉渲染管线画什么和怎么画。

STEP 1 当前已实现移动端异常效果图

STEP 2期望重力异常可视化效果

截至目前，已经实现了重力异常在移动端地图上的等值线可视化，然而本文目的是在移动客户端实现重力异常类似Argics软件中高程图的显示效果，继续设计算法和接口，考虑继续设计算法和接口。

图6 Agcis软件中高程图的展示效果

在底层OpenGL绘图接口之外，百度地图向开发者提供了高度封装的绘制接口，包括但不限于线、面、动画和覆盖物，使开发者能更加容易完成开发。类似Agcis中高程图的效果可以用在底层地图上覆一层自定义覆盖物的方法实现，地形图图层（GroundOverlay），又可叫做图片图层，即开发者可在地图的指定位置上添加图片。该图片可随地图的平移、缩放、旋转等操作做相应的变换。该图层是一种特殊的Overlay， 它位于底图和底图标注层之间（即该图层不会遮挡地图标注信息）。

我们当前获得的，可利用的数据是一个测区插值矩阵，其可以理解为由数个三角网格构成，重力异常图绘制上色的过程本质上是根据每个微型区域数值大小逐一着色，随着网格划分越密集，图片着色的细腻程度越高。

因此我们设计接口：

Void draw[GravityAnomaly](javascript:;)(fLoat [][] value ,**int** ilb,**int** iub,**int** jlb, **int** jub, **float**[ ] x, **float**[ ] y)

|  |  |  |
| --- | --- | --- |
| 变量名 | 类型 | 释义 |
| Value | Float[][] | 已插值测区重力数据 |
| Ilb | Int | 实际应用中很可能只需要绘制部分区域重力等值线，截取数据矩阵中从lib至jub行，jlb至jub列数 |
| Jub | Int |
| Jlb | Int |
| Jub | Int | 据进行等值线构造 |
| X | Float[] | 测区经度坐标集合 |
| Y | Float[] | 测区纬度坐标集合 |

对于每个三角网格设置色值，色值的确定基于以下思想，每个三角网格的三个端点分别对应着相应的重力插值，由于网格数量众多，单个面积微小，可以理解为微分单元，因而将三个端点测值的平均值视为该单元网格重力数值，通过划分区间，对数值设置对应的色值，就能实现类似Agcis中的绘制效果。

以上公式将计算每个三角网格微元三个端点重力插值的均值，并作为下一步选色填充网格的依据，在网格较少，精度设置不高时显示效果如下图所示：

图8 根据每个网格平均测值进行着色

5 基于C/S架构的地勘系统的编程实现

5.1项目总览

移动端：

|  |  |
| --- | --- |
| 开发环境 | Eclipse |
| 编程语言 | JAVA 7.0/XML/SQL |
| 联网协议 | TCP/IP |
| 代码量 |  |
| 最低可运行SDK版本 | API 8:Android 2.2(Froyo) |
| 目标SDK版本 | API 21:Android 5.0(Lollipop) |
| 编译用SDK版本 | API 27:Android 8.1 |
| 数据库 | SQLite |

服务端；

|  |  |
| --- | --- |
| 开发环境 | Eclipse |
| 编程语言 | JAVA 7.0/SQL |
| 联网协议 | TCP/IP |
| 代码量 |  |
| 数据库 | SQL Server |

5.1.1 移动客户端开发环境搭建

就像任何其他软件的开发一样，Android开发同样需要配置好开发环境，尽管可供选择的IDE有很多，但无论选择Eclipse还是Android Studeio，其中有几项组件是通用且必须要配置的：

第一步：安装JDK。

图9 Android开发环境搭建

JDK是 Java 语言的软件开发工具包，主要用于移动设备、嵌入式设备上的java应用程序。JDK是整个java开发的核心，它包含了JAVA的运行环境（JVM+Java系统类库）和JAVA工具。我们可以打开jdk的安装目录下的Bin目录，里面有许多后缀名为exe的可执行程序，这些都是JDK包含的工具。通过第二步讲到的配置JDK的变量环境，我们可以方便地调用这些工具及它们的命令。JDK包含的基本工具主要有：

javac：Java编译器，将源代码转成字节码。

jar：打包工具，将相关的类文件打包成一个文件。

javadoc：文档生成器，从源码注释中提取文档。

jdb：debugger，调试查错工具。

java：运行编译后的java程序。

第二步：下载安装Android SDK 。

SDK：（software development kit）软件开发工具包。被软件开发工程师用于为特定的软件包、软件框架、硬件平台、操作系统等建立应用软件的开发工具的集合。

因此，Android SDK 指的是Android专属的软件开发工具包,在解压之后我们可以打开SDK manager根据期望开发的Android版本选择相应的SDK组件进行下载。

图10 SDK使用SDK manager下载安装目标版本SDK

第三步：为选择的集成开发环境IDE安装ADT插件。

本文[Android开发](https://www.baidu.com/s?wd=Android%E5%BC%80%E5%8F%91&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1Y3nHR1PHmsnjbsnADkrycv0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3ErHn4njc1nWR3PjckPHczrjbz" \t "_blank)所用的开发工具是Eclipse，在Eclipse编译IDE环境中，安装ADT，ADT是一个用于EclipseIDE的插件，旨在给一个强大的，集成的环境中构建Android应用程序。为[Android开发](https://www.baidu.com/s?wd=Android%E5%BC%80%E5%8F%91&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1Y3nHR1PHmsnjbsnADkrycv0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3ErHn4njc1nWR3PjckPHczrjbz)提供开发工具的升级或者变更，可以理解为在Eclipse下开发工具的升级下载工具。

图10 SDK使用SDK manager下载安装目标版本SDK

5.1.2 移动客户端逻辑结构

是否拥有登录账号

主菜单

注册

读取重力仪测值

查看已有测区

重力场异常模拟

创建新测区

删除测点

设置

获得帮助

关于IGP

我的账户

查看测区信息

修改测区名称

添加测点

就像其他APP一样，本文开发的地勘系统可以以apk安装包的形式，方便地安装在Android版本适配的机器上，在打开应用后，系统将会随着用户具体的操作在数个Activity(活动)和Fragment之间切换。

运行后，首先系统将会启动相应活动加载后续运行资源，包括将来可能会用到的图片、音频和必要的组件验证。加载完毕之后，系统将自动切换到登录界面，此时用户将会面临选择，如果当前用户拥有一个可使用的账户，那么他可以选择登录，在此时后台将会与服务器交互并下载对应账户中存储的所有测区信息，这是非常高效的，不仅是对于个人用户而言，对于团队工作更是如此，这表示一个团队可以及时共享地质勘探信息。 而如果当前用户是新用户，并没有可登陆的账户，或者很不幸，尽管用户有可使用的账户，但他所处的偏远环境不允许他使用网络连接，那么他可以选择离线使用该系统，并将数据离线存储在本地。幸运的是Android系统自带SQLite数据库，本次客户端设计也是基于该数据库实现。

图11 客户端运行界面—登录/注册

设计编程实现SQLite数据库操作类DBHelepr，可以注意到其中实现了极为丰富的数据库操作接口，该类的调用将贯穿应用的整个生命周期。

不过本文强烈建议使用者申请一个注册账户，在没有商业应用之前，这是完全免费的，使用者仅需要提交必要的信息，比如用户名、密码、邮箱和联系电话，当使用者听从建议补全信息并按下注册按钮，后台将启动一个新线程并在其中创建一个封装了TCP/IP协议的套接字Socket尝试向服务器发送注册请求，此时创建一个新线程是必要的，Android系统不允许在主线程中创建套接字，因为申请操作的延时将会导致潜在的程序崩溃可能。刚好，在服务器端程序已经有Server-Socket在持续等待申请的到来，在接收到来自客户端的请求之后，根据标记来判断请求类型并调用相应方法，就像你想的那样，对于注册请求，服务端进程首先遍历数据库中是否存在重复的用户名，如果是这样，那么立即向客户端发送拒绝信息并标记错误类型，这是很有用的，因为错误类型远远不止一种，例如用户名重复、用户不存在或密码错误，而客户端在收到相应的反馈之后在屏幕上显示提示信息，方便使用者后续做出正确的操作。

图11 客户端运行界面—登录/注册

假设使用者完成了注册操作，那么在以用户的身份登录之后，他将会注意到以下主菜单界面，该界面简洁的展示了用户可以执行的操作，就像你看到的那样，自上而下屏幕依次显示了各一个网格和列表，网格中展示了这套地勘系统中四种主要功能，分别是创建新测区、查看已有测区、重力异常模拟和读取重力仪测值；而列表则由主菜单、关于IGP、获得帮助、设置和我的账户五个栏目构成，相信用户点击之后将会将感到熟悉和亲切，因为这与微信功能栏的切换操作是一致的。

图11 客户端运行界面—主菜单界面

现在让我们来依次体验各项功能，就一个地勘系统而言，测区信息的记录是极其重要的，作为其余功能实现的基础，我们首先尝试进行创建新测区操作。在触碰屏幕相应区域之后，用户将会看到一个输入框，现在输入测区名称并点击确定，在这个过程背后，客户端将开辟子线程再次向服务器发出请求，不过这次有些不同，在这套地勘系统中服务器实际上拥有两个数据库，一个User库和一个District库，从名字上我们就可以大致判断他们各自的作用，User库存储着所有用户的信息，包括名称、密码和邮箱等，而District库则存储着所有测区的信息，有人可能会问该如何实现仅更新特定账户名下的测区信息，对于这种情况，我们在District库的存储结构中添加用户标记，从而保证不会有混淆的情况发

**//这里添加一个数据库数据结构的可视化**

生。回到刚才，如果遍历该用户名下所有测区之后发现，之前输入的测区名称已经存在，则立即反馈出错信息给客户端，要求用户修改新建测区的名称，实际上这个过程的描述隐藏了很多细节，类似客户端信息即时更新Handler中的回调处理等等。如果一切操作正确，那用户将会看到测区信息输入界面。

图11 客户端运行界面—主菜单

这个界面非常简洁，在这里地图组件占了屏幕绝大部分比例，让用户可以更好的观察自己所处的地理位置，这里有一个不容易被注意到的细节处理，在获取定位获取经纬度的时候，客户端可以选择基于基站定位或是基于GPS定位，但无论是哪种定位获取的位置数据都无法直接在百度地图上正确定位，实际上出于某些安全原因也无法在任何国内地图上正确显示，而百度地图使用的坐标系是自有的bd09ll（百度经纬度坐标）坐标系，幸运的是本次已经编程完成了后台两种坐标的转换，所以用户在此界面上可以准确的看到自己目前所处的位置。当用户准备记录该点地质信息，他可以点击添加测点按钮切换到下一个活动，就像你看到那样，这个页面提供了丰富的数据记录组件，包括经纬度、测值、测点名称和备注等等，其中经纬度已经由系统自动填写正确，无需用户再次手动输入。当用户完成某点的记录，他可以再次添加测点信息，或者返回上个界面查看整个测区内测点的布设情况，没错，借助于重写OnResume()方法并重读数据库，有所得测点信息都会在地图重绘时更新，显示所有已记录测点，以方便地质工作者查看整个测区的工作进展。

在用户退出测区地图界面之后，也就意味着本次新建测区人物的结束，在这之后如果用户希望查看或修改已有测区的信息，那么他可以在主界面上点击查看已有测区，就像下图展示的一样，但用户实际体验中将会看到比想象中更多的测区，因为移动客户端进程已经从服务器数据库下载了所有该用户名下的测区信息，如果用户希望对某个测区做出一些修改，类似修改名称，添加测点，甚至将测区删除，那么他可以通过简单的操作实现，请注意这些修改都是与服务器数据库操作同步的，在完成修改之后，新的测区信息会将服务器中已有数据进行替换。有人可能会觉得这将导致某些冲突，但本文已经注意并解决了这一点问题，无论当用户尝试修改测区名称还是向测区中添加新测点，这些可能导致关键字索引冲突的行为都必须通过检查，如果更新的测区或测点名称与已有重复，那么用户将看到相应的提示信息，有利于后续做出正确的操作。

图11 客户端运行界面—主菜单界面

图11 客户端运行界面—主菜单界面

以上便是测区详细信息界面，在这个界面上左上部分显示了一个地图组件，用户会看到所有的测点都会在这个区域内显示；右上角是一个可滑动列表，包含了测区内所有测点名称；当用户选择希望查看的测点之后，具体的信息将会自动填写到界面下部的栏目中，并且同时地图将会自动定位到以该测点为中心，方便用户确定测点在测区中的相对位置。假使用户希望修改该测点的某些信息，他可以简单点击一下修改测点信息按钮，在这之后他将看到一个测点信息添加界面，用户可能会感觉熟悉，因为这实际上是一个复用的addPointer\_Activity，任何

图11 客户端运行界面—主菜单界面

添加或修改测点信息的操作都可以用它来实现，代码复用大大简化了开发步骤，

实际上这也是OOP面向对象编程的基础思想。在完成对测点信息的更新之后，返回上一级测区概况界面，用户会发现上一步对测点所做的修改已经实时显示在地图组件上——修改经纬度会导致测点位置变化，删除则会在将测点在地图上移除。另外就像你想的一样，所有这些操作已经同步对服务器数据库做了修改，保证无论是单人用户或者团队用户都能在登录后准确的获得最新的测区信息。

接下来我们将要介绍的是在前文中介绍了实现过程的区域重力异常可视化功能，

图11 客户端运行界面—主菜单界面

在使用这个功能之前，我们假设用户已经做好了准备，拥有了一个存储着合适(已经经过重力勘探方法校正)数据的测区，现在让我们点击功能网格中的第三个选项——重力异常模拟，就像上图展示的一样，用户首先要从所有的测区列表中选择目标测区，这个过程中在移动客户端程序中创建Intent实例，Intent可以理解为在Activtiy间传递参数的数据结构，之后