

测量全生命周期支持系统 设计 开发 报告

同济大学测绘与地理信息学院

1551126 余周炜

1551140 王雪辰

1551128 江子宇

目 录



1	项目简介	1
1.1	项目背景	1
1.2	开发平台	1
1.3	数据	1
1.4	程序框架	2
2	测量辅助系统	4
2.1	功能	4
2.2	函数说明	5
3	等高线生成系统	11
3.1	功能说明	11
3.1.1	读取 pnt 文件并保存为 shp 文件	11
3.1.2	建立 TIN	13
3.1.3	生成等高线	13
3.1.4	测区面积量算	14
4	小组合作	15
4.1	远程库	15
4.2	本地库	17
5	提交文件	18

第 1 章 项目简介



本项目的名称是“测量全生命周期支持系统”,使用 vs2010+arcEngine10.1 开发,分为两个部分,一部分进行**辅助测量**,是为测量辅助系统,另一部分**生成等高线**。

1.1 项目背景

测量实习中,数字平面地形图测绘部分需要用到 CAD 进行设站点、后视点选择以及地形图展绘;等高线测量部分需要用到 ArcGIS 生成等高线。然而在实际的操作过程中,我们常常为找不到控制点和输入新的控制点而烦恼。例如校庆修路使控制点的位置和数量都发生了一定的变化,此时就需要对控制点进行大批量的更新。此外,在 ArcMap 中生成等高线需要经历生成 TIN、TIN 转栅格、生成等高线的过程,较为繁琐。

为了解决测量实习中的这些问题,开发了本系统,将测量过程中的各种操作用 ArcEngine 开发集成,使测量过程更加简单流畅。

1.2 开发平台

vs2010+ArcEngine10.1

1.3 数据

原始数据 校园 dwg 图 (包含控制点图层以及地物图层)

数据准备阶段将包含控制点的 CAD 图层转为 point 类型的 shp 文件，将包含建筑物的 CAD 图层转为 polygon 类型的 shp 文件。

1.4 程序框架

本项目主要有三个窗体，Form1 为主窗体，如图1.1所示。

图 1.1: 主窗体



第二个窗体是 FormSurvey, 辅助有关测量行为的进行，如图1.2所示。
第三个窗体是 FormCoutour, 进行有关等高线的绘制，如图1.3所示。
此外还有 FormDistance, AddControlPoint 等对话框。



图 1.2: FormSurvey

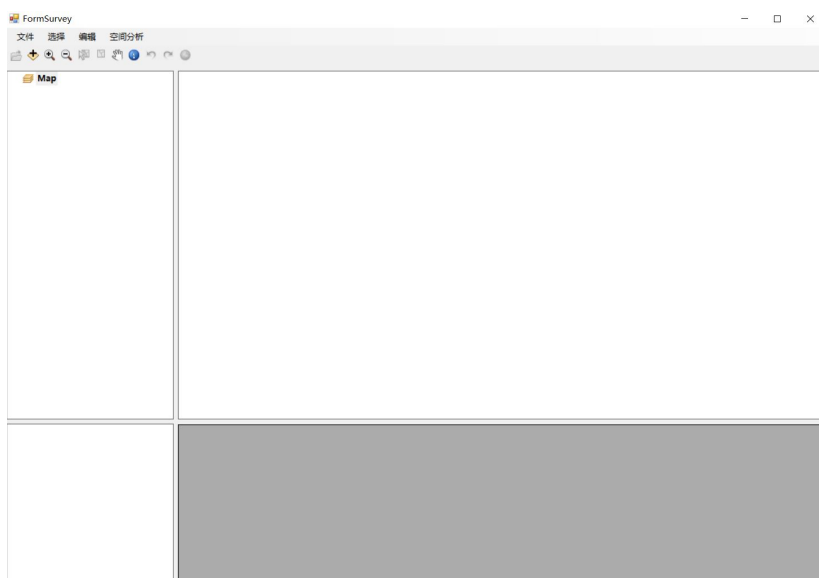
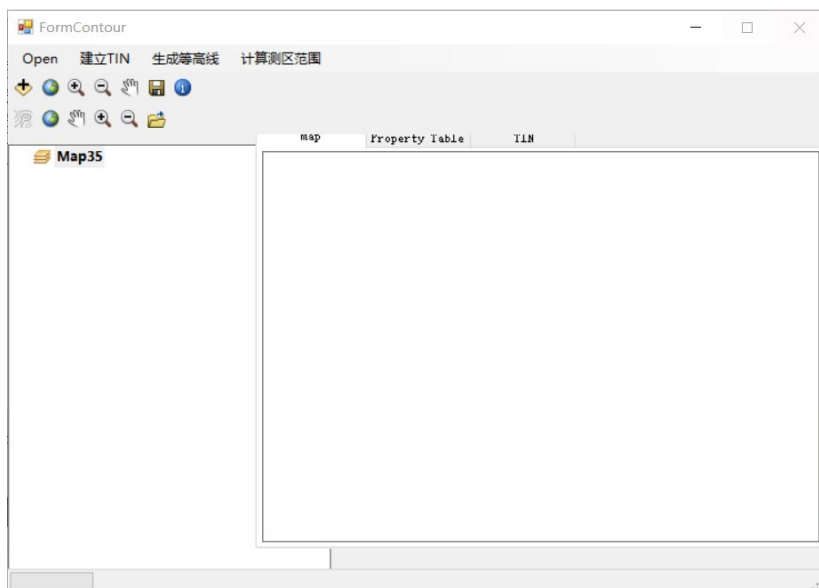


图 1.3: FormCoutour



第 2 章 测量辅助系统



2.1 功能

测量辅助系统实现了如下特色功能

1. 鹰眼功能。
2. 鼠标拖动移动图层功能。
3. 输入坐标的方式添加控制点。
4. 拉框选择测区范围，显示测区边界，并将测区范围内的控制点高亮显示。
5. 选择测站所在的控制点、输入测距，在考虑遮蔽和测距的前提下给出可用的后视点。
6. 选中测站所在的控制点，给出测区范围。

下面分别说明这些功能

鹰眼功能 利用 `axMapControl1` 的 `OnExtentUpdated` 事件，当主视图范围改变时，鹰眼视图主视图在鹰眼视图中的范围同时改变。利用 `axMapControl2` 的 `OnMouseDown` 事件，当鼠标在鹰眼视图中点击时，`axMapControl1` 的范围同时改变。

图层移动 利用 `axTOCControl1` 的 `OnMouseDown` 事件和它的 `HitTest` 方法，记录将要移动的图层，利用 `OnMouseUp` 事件和 `IMap` 的 `MoveLayer` 方法，将其移到所属的位置。

选择测区范围 单击该选项时，将 `bool` 变量 `flagSelectFeature` 置为 `Checked`，利用 `axMapControl1` 的 `OnMouseDown` 事件和 `TrackRectangle()` 方法，记录该选框的四个角点坐标，创建一条 `Polyline` 加入 `axMapControl1`

的图形容器中。

坐标添加控制点 单击该选项时，将弹出对话框以输入坐标，单击确定时，将该坐标保存在类 `GlobalData` 的公有静态变量中，找到点图层并添加。

给出后视点 1. 点击输入仪器测量选项输入仪器测程；2. 以该测程为半径生成一个缓冲区；3. 选取该缓冲区内的控制点和建筑物；4. 对每个除测站所在控制点之外的控制点和测站点生成一条 `polyline`，与每个建筑物求交，若该 `polyline` 与所有建筑都不相交，则选择该控制点为后视点。

给出可测范围 选中测站所在控制点，以测程为半径并显示。

此外，单击图层时，可以将属性表显示在下方的 `dataGridView` 中，在 `TOCControl` 中右键单击，会弹出 `menuStrip`，将图层上移/下移或删除。还可以在选项中添加控制点，删除控制点和（随机）修改控制点符号。

2.2 函数说明

在文件 `FormSurvey` 中：

```
public DataTable GetLayerData(IFeatureLayer layer)
```

作用 从 `IFeatureLayer` 中获取属性表，并返回 `DataTable`。

描述 利用到要素层中的要素类的 `Fields` 的 `FieldCount` 字段和 `get_Field(int index)` 方法，得到要素类的属性表的所有字段，利用字段的 `AliasName` 作为 `DataTable` 的字段名，对于每个要素，用 `get_Value(int index)` 方法获取其第 `index` 个字段的值。

注意 使用完 `FeatureCursor` 后要释放指针

```
private void 加载Shp文件ToolStripMenuItem_Click(object sender,
    EventArgs e)
```



作用 打开文件。

描述 运用 AxMapControl 的 AddShapFile 方法，添加文件。

```
private void axMapControl1_OnExtentUpdated(object sender,  
    IMapControlEvents2_OnExtentUpdatedEvent e)
```

作用 更新 axMapControl2 的显示范围，画出 axMapControl1 的显示的边框。

描述 将事件的 e.newEnvelope 强制转换为 IEnvelope，利用 graphicsContainer 的 AddElement 添加元素画出边框。注意 LineSymbol, FillSymbol 和 RgbColor 的建立。

```
private void 删除图层ToolStripMenuItem_Click(object sender,  
    EventArgs e)
```

作用 删除所选图层。

描述 利用 TOCControl 的 GetSelectedItem 方法获取选中的 TOCControlItem，在 MapControl 里找到对应图层，用 MapControl 的 DeleteLayer 方法删除该图层。

```
private void axTOCControl1_OnMouseDown(object sender,  
    ITOCControlEvents_OnMouseDownEvent e)
```

作用 TOCControl 按下鼠标触发事件

描述 利用 TOCControl 的 HitTest 方法，获取鼠标点击的 item，根据鼠标按键的不同执行不同的操作：左键将 pMoveLayer 置为所点击的图层，为拖动图层做铺垫，另外 dataGridView 显示其属性；右键弹出内容菜单。

```
private void axTOCControl1_OnMouseUp(object sender,  
    ITOCControlEvents_OnMouseUpEvent e)
```



作用 TOCControl 抬起鼠标触发事件

描述 利用 TOCControl 的 HitTest 方法，获取鼠标点击的 item，根据鼠标按键的不同执行不同的操作：左键将 pMoveLayer 置为所点击的图层，为拖动图层做铺垫，另外 dataGridView 显示其属性；右键弹出内容菜单。

```
private void 选择测区范围ToolStripMenuItem_Click(object sender,
    EventArgs e)
```

作用 flagSelectFeature 置为 ToolStripMenuItem 的 Checked 属性值，为 axMapControl 的 OnMouseDown 事件作铺垫。

```
private void axMapControl1_OnMouseDown(object sender,
    IMapControlEvents2_OnMouseDownEvent e)
```

作用 MapControl 的按下鼠标触发该事件

描述 如果 flagSelectFeature 为真，则调用 axMapControl1 的 TrackRectangle 方法，返回一个 geometry，并利用这四个角点新建一个 Polyline，并用 GraphicsContainer 的 AddElement 方法加入 Polyline 元素，即为测区范围；其次，创建一个 spatialFilter，表示与 Polygon 相交，利用点图层的 FeatureClass 的 Search 方法，返回在矩形内的控制点，并将其选中高亮显示。如果 flagCreateFeature 为真，则创建控制点。如果 flagSelectStation 为真，则调用 axMapControl1 的 TrackCircle 方法，选中测站点并高亮显示。

```
private void 图层上移ToolStripMenuItem_Click(object sender,
    EventArgs e)
```

作用 将选中的图层上移一层

描述 利用 axTOCControl1 的 GetSelectedItem 方法找出选中图层，遍历所



有图层，找到该图层的 index，并用 axMapControl1 的 MoveLayerTo 方法将该图层上移一层。

```
private void 图层下移ToolStripMenuItem_Click(object sender,
    EventArgs e)
```

作用 将选中的图层下移一层

描述 利用 axTOCControl1 的 GetSelectedItem 方法找出选中图层，遍历所有图层，找到该图层的 index，并用 axMapControl1 的 MoveLayerTo 方法将该图层下移一层。

```
private void 添加控制点ToolStripMenuItem_Click(object sender,
    EventArgs e)
```

作用 将 flagCreateFeature 置为添加控制点 ToolStripMenuItem 的 Checked 属性。

```
private void 更换控制点符号ToolStripMenuItem_Click(object sender
    , EventArgs e)
```

作用 更换控制点的符号

描述 选择控制点所在的图层，利用随机数，生成不同的颜色和形状，利用 ISimpleMarkerSymbol 和 SimpleRenderer 设置符号和图层的渲染。

```
private void 删除控制点ToolStripMenuItem_Click(object sender,
    EventArgs e)
```

作用 将选中的控制点删除

描述 通过 ISelection 和 IEnumFeature 接口将选中的实体遍历并用 IFeature 的 Delete 方法删除。



```
private void axMapControl2_OnMouseDown(object sender,  
    IMapControlEvents2_OnMouseDownEvent e)
```

作用 在 axMapControl1 的地图的中心移到 axMapControl2 中所点的位置。

描述 利用 axMapControl1 的 CenterAt 方法即可

```
private void 坐标添加控制点ToolStripMenuItem_Click(object sender  
    , EventArgs e)
```

作用 利用弹出对话框的方式，输入坐标添加控制点

描述 首先找到点图层，利用 IPoint 的 PutCoords 方法生成点，利用 Feature-Class 的 CreateFeature() 方法创建点。

```
private void 选择当前测站ToolStripMenuItem_Click(object sender,  
    EventArgs e)
```

作用 将 flagSelectStation 置为 true。

```
private void dToolStripMenuItem_Click(object sender, EventArgs e  
    )
```

作用 在考虑建筑物遮挡的情况下给出可以看到的后视点。

描述 首先，建立一个半径为测程的 buffer，第二，得出在 buffer 内的控制点和建筑物图层，第三，对每个在 buffer 内的非测站点，与测站点连一条线，此即为视线，判断该视线是否与建筑物相交（ITopologicalOperator 的 Intersect 方法），如果与所有建筑物均不相交，则将其高亮显示。

```
private void 输入仪器ToolStripMenuItem_Click(object sender,  
    EventArgs e)
```

作用 弹出输入仪器测程对话框。



```
private void 给出可测区域ToolStripMenuItem_Click(object sender,  
    EventArgs e)
```

作用 给出可测区域的范围

描述 先以测程为半径创建一个缓冲区，然后将该缓冲区与测区范围作减运算（clip），得到可测区域范围。



第3章 等高线生成系统



等高线生成系统实现了如下特色功能

1. 建 TIN
2. TIN 转栅格
3. 生成等高线
4. 由 shp 文件建立 TIN

下面这对这些作详细说明。

3.1 功能说明

3.1.1 读取 pnt 文件并保存为 shp 文件

1. 实现的功能

pnt 文件是从全站仪中直接导出来的数据，是一种文本文件，每行是一条记录，记录以逗号分隔。

该步骤将 pnt 文件中的高程点数据读取出来，建立新的 point 类型的 shapefile, 并对其建立属性表。

2. 主要接口

IWorkspaceFactory, IFeatureWorkspace, IFeatureClass, IFeatureCursor, IField, IFieldEdit

3. 函数与数据组织结构

- (a) 点的数据结构

```
struct Point3D
{
```

```
public string ID;  
public double X;  
public double Y;  
public double Z;  
}
```

- (b) 读取 pnt 文件
-

```
Private List<Point3D> Read_ContourLineFile(string  
    FileName)
```

- (c) 创建 shp 文件对应的基本属性表
-

```
private IFields CreateShapeFields(esriGeometryType  
    p_esriGeotype)
```

- (d) 创建 shp 文件
-

```
private IFeatureLayer CreateSHP_Point(List<Point3D>  
    PointList, string FileFullPath)
```

- (e) 根据高程点的信息建立完整属性表
-

```
private void Complete_PropertyTable(ref IFeatureClass  
    pFeatureClass, List<Point3D> PointList)
```

- (f) 显示属性表
-

```
private void Display_PropertyTable(IFeatureLayer  
    pFeatureLayer)
```

- (g) 打开原始测量文件
-

```
private void 原始测量文件ToolStripMenuItem_Click(  
    object sender, EventArgs e)
```



3.1.2 建立 TIN

1. 实现的功能

由 shp 文件中的数据建立 TIN

2. 主要的接口

IWorkspaceFactory, IFeatureWorkspace, ITin, ITinEdit

3. 函数

(a) TIN 生成函数

```
private ITin Create_TIN(IFeatureClass pFeatureClass,  
    IField pField)
```

(b) 建立 TIN

```
private void 建立TINToolStripMenuItem_Click(object  
    sender, EventArgs e)
```

3.1.3 生成等高线

1. 实现的功能

根据 TIN 生成等高线 (polyline 类型的 shapefile)

2. 主要接口

IWorkspaceFactory, IFeatureWorkspace, IFeatureClass, ITinSurface

3. 函数

(a) 等高线生成函数

```
private IFeatureClass Create_ContourLine(ITin pTin,  
    string WorkspaceName, string FileName)
```

(b) 建立 TIN



```
private void 建立TINToolStripMenuItem_Click(object  
sender, EventArgs e)
```

3.1.4 测区面积量算

1. 实现的功能

量测区域的面积（已测得的等高线覆盖的面积）。通过对高程点建立凸包，计算大约的量测面积。

2. 主要接口

IWorkspaceFactory, IFeatureWorkspace, IFeatureClass, ITopologicalOperator（建立凸包），IArea（获取面积）

3. 函数

- (a) 将 Point 类型的 pfeature 转为 MultiPoint, 为建立凸包做准备

```
private IFeature Convert_Point2MultiPoint(  
IFeatureClass PointFeatureClass)
```

- (b) 计算测区范围

```
private void 计算测区范围ToolStripMenuItem_Click(  
object sender, EventArgs e)
```



第 4 章 小组合作



本小组的成员通过 `github` 进行合作及版本控制，方式是组长把程序推到 `github` 上，小组成员从组长的库中 `fork` 到自己的库中，更改后组长提出拉取请求 (`pull request`)，组长审核后将该提交合并 (`merge`) 到主分支。

需要说明一下，`git` 是一个分布式版本控制系统，`Github` 是一个代码托管平台，用 `git` 进行版本控制。

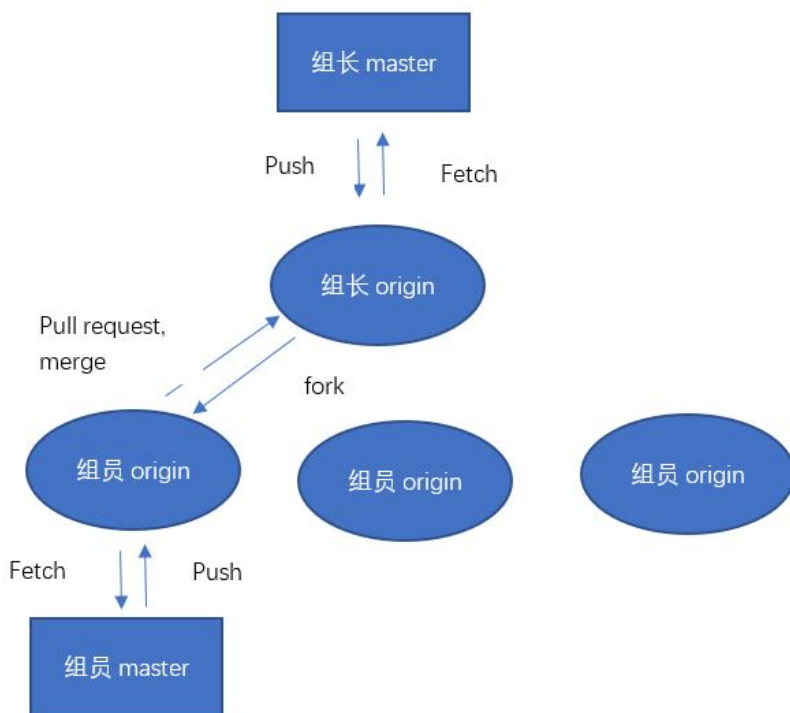
`Git` 版本控制系统中，将本地库叫 `master`（主分支），将远程库叫 `origin`。其他的分支这里还没有用到不讲。

4.1 远程库

`github` 上提交的流程如图4.1所示

需要说明的是，因为主场的远程库在不停变动，所以组员在每次修改之前，需要先 `merge` 组长的远程分支到 `master`，然后才能进行修改。

图 4.1: 提交流程

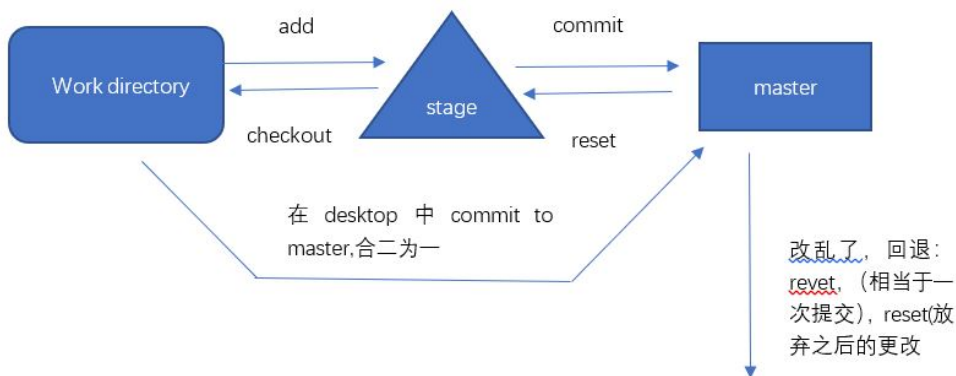


4.2 本地库

下面针对本地库作讲解。在本地库中，将你选择建立 git 的文件夹称为工作区，**add** 之后到暂存区，**commit** 之后到 **master**（主分支）。

git 本地的操作如图所示4.2所示

图 4.2: git 本地操作



第 5 章 提交文件



表 5.1: 文件

项目文件	ArcEngineProgram/ ArcEnginePro- gram.sln
源代码	ArcEngineProgram/ ArcEngineProgram/ *.cs
编译好的程序	ArcEngineProgram/ ArcEngineProgram/ bin/Debug/ArcEngineProgram
说明文件	document.tex,document.pdf
数据	gis/ pointt.shp,polygon.shp, 高程数据/ CL05.pnt