

《GIS 设计与应用》

个人项目总结报告

项目名称： 基于Arc Engine的北京市找房指南

姓名： 黄志成

学号：2000012520

院系： 地球与空间科学学院

专业：地理信息科学

2023 年 2 月 26 日

目录

**[1. 项目概述 1](#_Toc20071)**

[1.1. 项目背景 1](#_Toc13249)

[1.2. 项目目标 1](#_Toc29028)

**[2. 项目完成情况 1](#_Toc12179)**

[2.1. 总体方案 1](#_Toc3883)

[2.2. 项目内容与技术路线 1](#_Toc25510)

[2.3. 完成情况说明 6](#_Toc1886)

**[3. 主要成果介绍 6](#_Toc2024)**

[3.1. 功能展示 6](#_Toc5542)

[3.2. 项目开源 9](#_Toc1804)

**[4. 总结 10](#_Toc18665)**

[4.1. 开发工作评价 10](#_Toc20313)

[4.2. 总结与收获 10](#_Toc28004)

1. **项目概述**
   1. **项目背景**

网络上的房屋信息繁杂，在北京室购房、租房除了需要考虑房价以外，还需要考虑通勤、娱乐设施、环境等因素，找到心仪的房源并非易事。因此，我们决定在《GIS 实验》课程北京市人居适宜性项目的基础上，开发基于 Arc Engine 的北京市找房指南（Apartment Hunting Guide in Beijing Based on Arc Engine），整合房源信息与人居适宜性分析结果，为看房提供更为全面的参考。

* 1. **项目目标**

本项目旨在面向北京市购房者开发一个北京市找房系统，用户可根据个人需求，综合考虑房价、环境、公共设施等因素，寻找适合的房屋。

1. **项目完成情况**
   1. **总体方案**

本系统是基于 Arc Engine 的二次开发，开发语言为 C#，.NET Framework 框架为4.5.2，开发工具为Visual Studio 2015，GIS 数据存储方式选择以文件（ShapeFile、Geodatabase）方式存储，地图文档以.mxd文件的方式存储，最终系统运行于 PC 端。开发过程中，版本控制采用git管理的方式。

* 1. **项目内容与技术路线**

根据系统功能分配，本项目可分为以下五个模块：数据的存取与管理、地图可视化、属性查询、空间查询、路径规划。下面将一一介绍每个模块的内容与技术路线：

1. **数据存取与管理**

系统打开时会加载默认地图，包含北京市的行政区划、路网，以及房源、中小学、医院、公交地铁站、娱乐场所等POI点图层。用户可以自己添加数据到地图中（支持ShapeFile、Geodatabase、Raster、Server Data、Layer File），操作后可对地图进行保存。

技术路线：使用Arc Engine提供的ICommand接口，调用其中的CreateNewDocument、ControlsOpenDocCommandClass、MapDocumentClass、ControlsSaveAsDocCommandClass等子类，实现地图文档的新建、打开、保存；使用AxMapControl控件的LoadMxFile方法实现默认地图文档的加载；使用AxToolbarControl控件中的Add Data工具进行添加数据操作。

示例代码：

            //默认地图路径

            string mxd\_path = @"D:\MyProject\map\";

            string mxd\_filename = @"beijing.mxd";

            //打开默认地图

            axMapControl1.LoadMxFile(mxd\_path + mxd\_filename, 0, Type.Missing);

            axMapControl1.Extent = axMapControl1.FullExtent;

1. **地图可视化**

用户可对地图实现基本的放大、缩小、漫游操作，可识别要素查看对应属性。

技术路线：通过AxToolbarControl控件实现。

1. **属性查询**

用户可以根据小区名称搜索对应的房源，显示小区名称包含输入字符串的房源。系统还支持根据行政区和房价进行筛选，可自定义输入房价范围。查询结果在地图上高亮显示，房源的名称、位置、房价、数量信息在表中显示。

技术路线：使用TextBox、ComboBox控件获取输入信息生成对应SQL语句，使用IQueryFilter的WhereClause属性实现根据SQL查询；使用IActiveView类的PartialRefresh方法实现高亮显示查询结果；通过DataGridView控件显示查询结果信息。

示例代码：

try

            {

                //清除查询结果

                mMapControl.Map.ClearSelection();

                IActiveView pActivaView = mMapControl.Map as IActiveView;

                //设置过滤条件

                IQueryFilter pQueryFilter = new QueryFilterClass();

                pQueryFilter.WhereClause = sql;   //SQL语句

                //查询

                IFeatureCursor pFeatureCursor = mFeatureLayer.Search(pQueryFilter, false);

                //创建一个新的envelope对象用于缩放

                IEnvelope pEnvelope = new EnvelopeClass();

                bool bHasFeatures = false;

                int featureCount = 0;

                //获取查询要素

                IFeature pFeature = pFeatureCursor.NextFeature();

                //判断是否获取到要素

                while (pFeature != null)

                {

                    bHasFeatures = true;

                    featureCount++;

                    mMapControl.Map.SelectFeature(mFeatureLayer, pFeature); //选择要素

                    mMapControl.Extent = pFeature.Shape.Envelope;   //放大到要素

                    //将要素的几何体合并到envelope中

                    IGeometry pGeometry = pFeature.ShapeCopy;

                    pEnvelope.Union(pGeometry.Envelope);

                    pFeature = pFeatureCursor.NextFeature();

                }

                if (bHasFeatures == true)

                {

                    if (featureCount == 1)

                        mMapControl.CenterAt(pEnvelope.LowerLeft);

                    else

                    {

                        //调整envelope的大小，确保要素可见

                        pEnvelope.Expand(1.5, 1.5, true);

                        //缩放到envelope所表示的范围

                        mMapControl.Extent = pEnvelope;

                    }

                }

                pActivaView.PartialRefresh(esriViewDrawPhase.esriViewGeoSelection, null, null);

                pActivaView.Refresh();

            }

            catch (Exception ex)

            {

                MessageBox.Show(ex.Message);

            }

1. **空间查询**

系统支持查询指定小区周围一定范围内的设施点数量，比如“金华园”周围3千米以内的公交地铁站。查询结果在地图上高亮显示，表中显示设施点的名称、类型、位置信息。

技术路线：使用TextBox、NumericUpDown、ComboBox空间获取查询条件；通过ITopologicalOperator类创建缓冲区，ISpatialFilter类设置空间查询属性；使用IActiveView类的PartialRefresh方法实现高亮显示查询结果；通过DataGridView控件显示查询结果信息。

示例代码：

                // 创建缓冲区

                ITopologicalOperator topoOperator = pointFeature.Shape as ITopologicalOperator;

                IGeometry bufferGeometry = topoOperator.Buffer(bufferDistance);

                // 在图层2中查询出距离该点要素一定范围的点

                ISpatialFilter spatialFilter = new SpatialFilterClass();

                spatialFilter.Geometry = bufferGeometry;

                spatialFilter.SpatialRel = esriSpatialRelEnum.esriSpatialRelIntersects;

                IFeatureCursor sFeatureCursor = sFeaturelayer.Search(spatialFilter, false);

                IFeature sFeature = null;

                while((sFeature=sFeatureCursor.NextFeature())!=null)

                {

                    DataRow dRow = dTable.NewRow();

                    dRow["name"] = sFeature.get\_Value(sFeature.Fields.FindField("name"));

                    dRow["adress"] = sFeature.get\_Value(sFeature.Fields.FindField("adress"));

                    dRow["level"] = sFeature.get\_Value(sFeature.Fields.FindField("level"));

                    dRow["area"] = sFeature.get\_Value(sFeature.Fields.FindField("area"));

                    dTable.Rows.Add(dRow);

                }

                //数据在表中显示

                dataPOITable.DataSource = dTable;

1. **路径规划**

根据上一步空间查询的结果，用户在表中选择目的地，输入起点可基于对路网的网络分析生成最短路径（阻抗为距离）。

技术路线：先在ArcGIS中创建网络数据集、构建网络；程序中使用NetworkAnalyst扩展实现网络分析、生成最短路径。

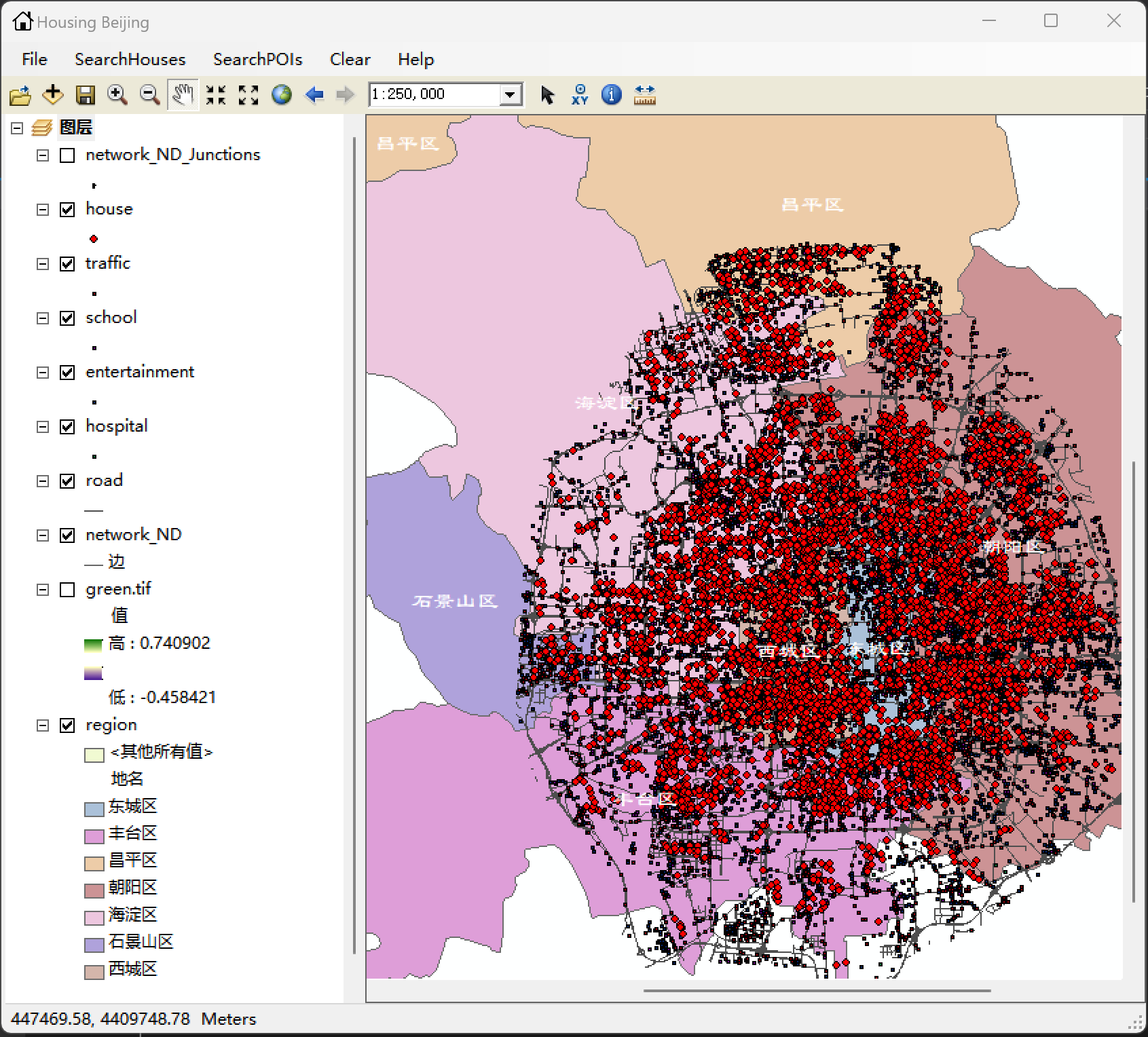
本模块未能成功实现，代码存在诸多bug，不再展示。

* 1. **完成情况说明**

对照项目申请，本项目有两处修改需要说明：一是立项时计划分别对新房、二手房、租房三种看房需求开发，爬取链家网房源数据时发现房源数据多，若全部放在地图中，将极大影响地图可视化效果与用户体验，因此只选择小区的房源数据（包含小区名、房产数、平均价格）；二是空间分析的结果为栅格数据，并线性拉伸到0-100的分数，作为买房参考并不直观，因此改用周围设施点的数目作为购房参考，只保留了拉伸后的NDVI影像作为底图。

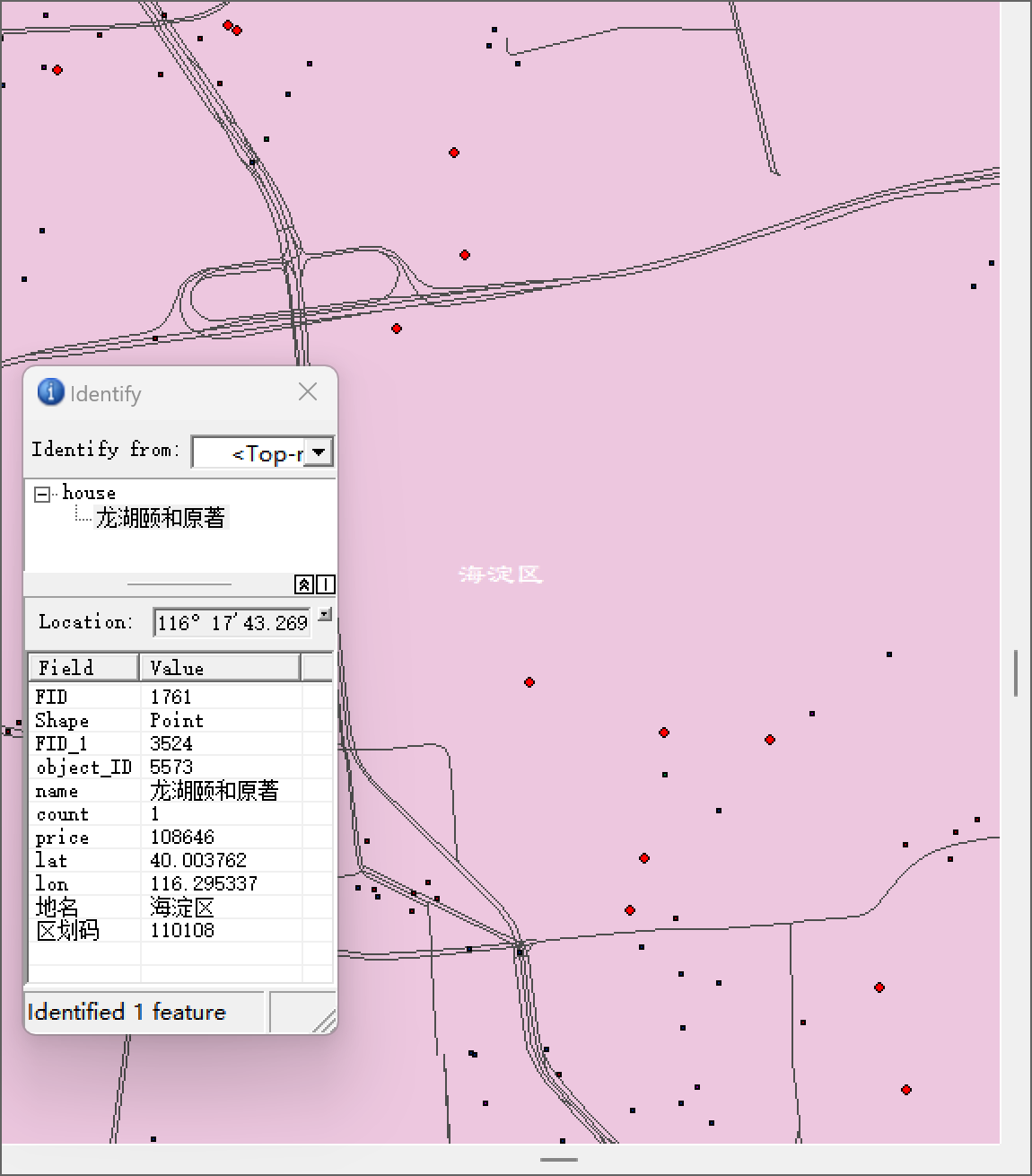
本项目的五个模块：数据的存取与管理、地图可视化、属性查询、空间查询、路径规划，除路径规划无法成功运行外，均已在系统中实现。

1. **主要成果介绍**
   1. **功能展示**
2. **主界面**



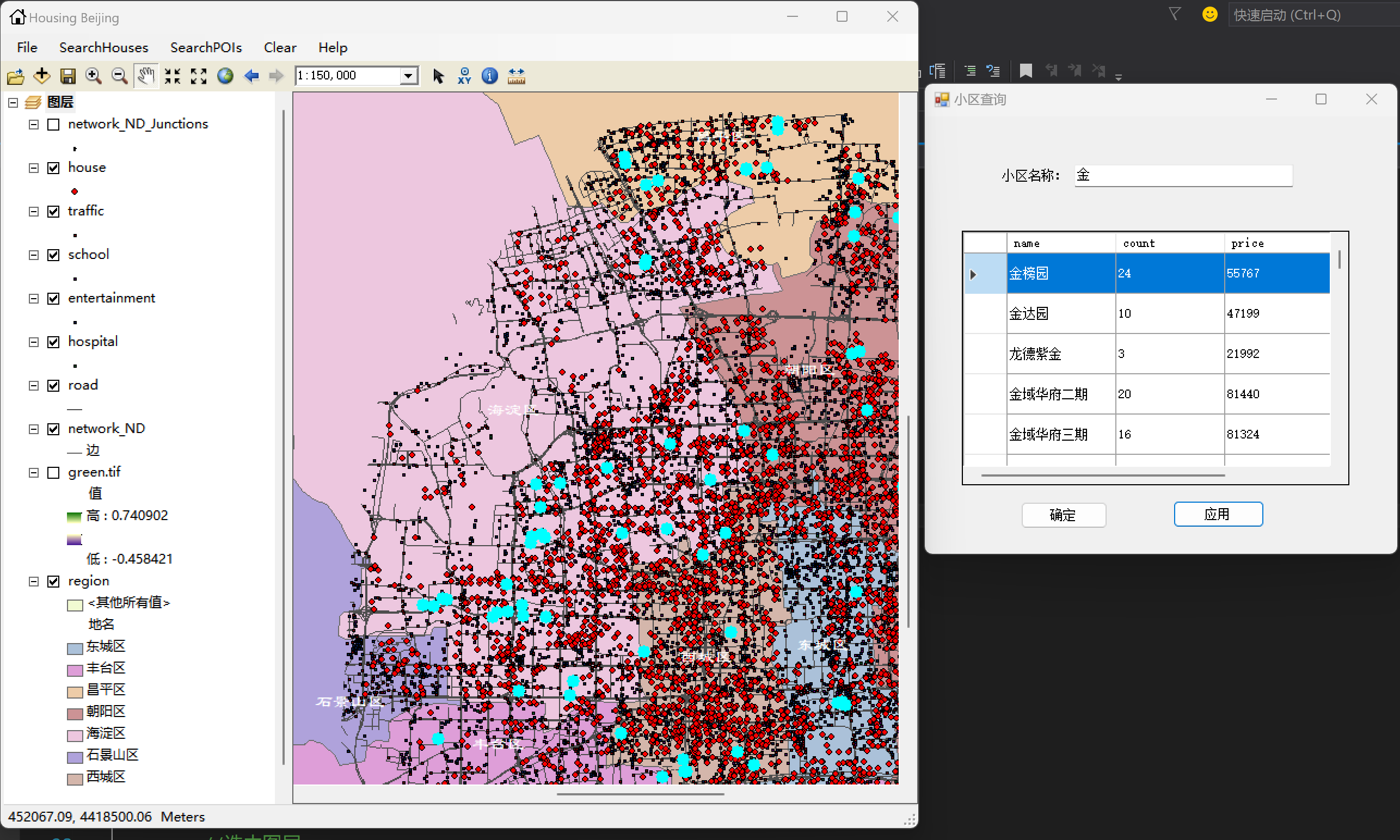
主界面

1. **查看某房源信息**



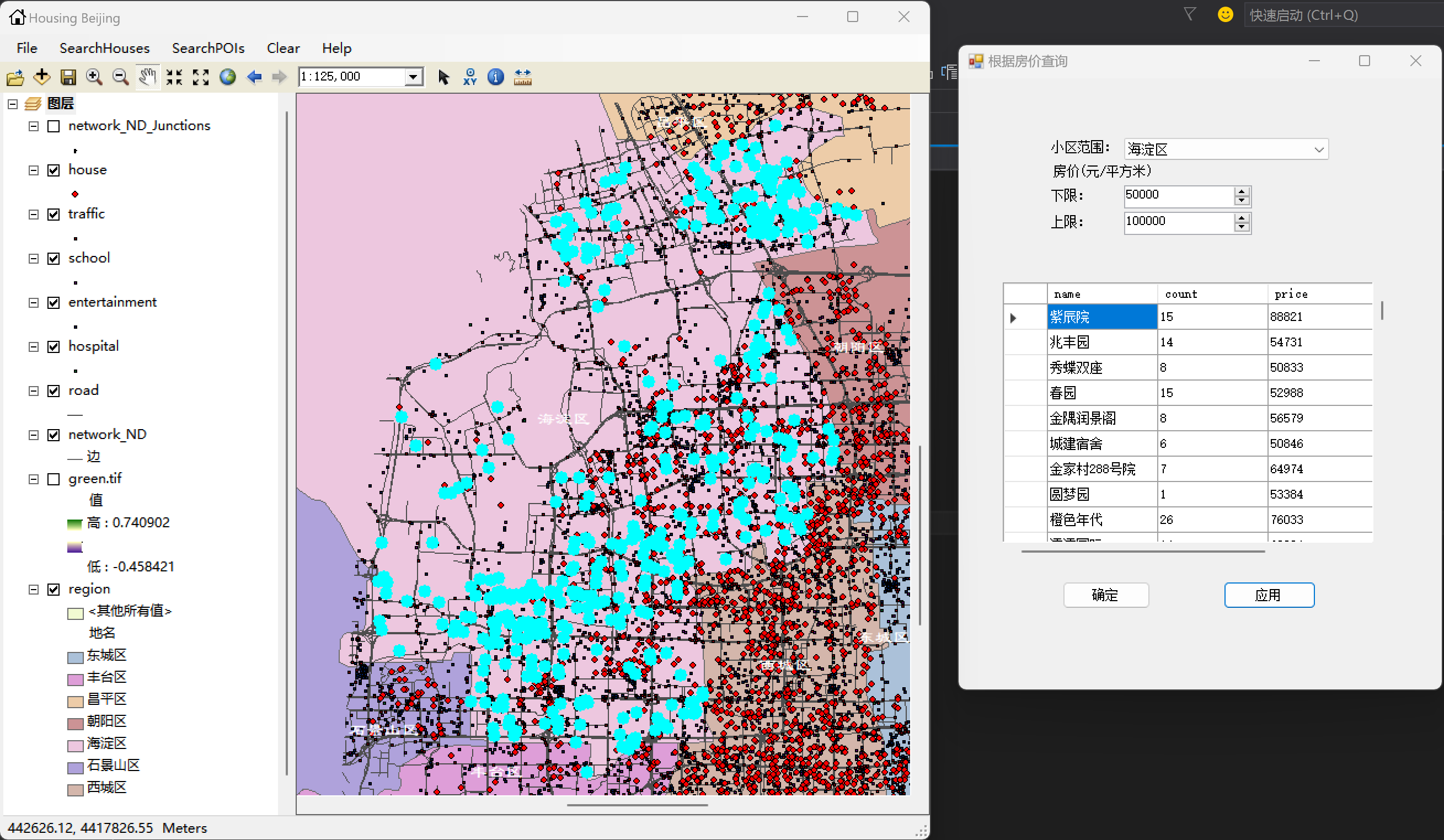
某小区房源信息

**（3）根据小区名称查询**



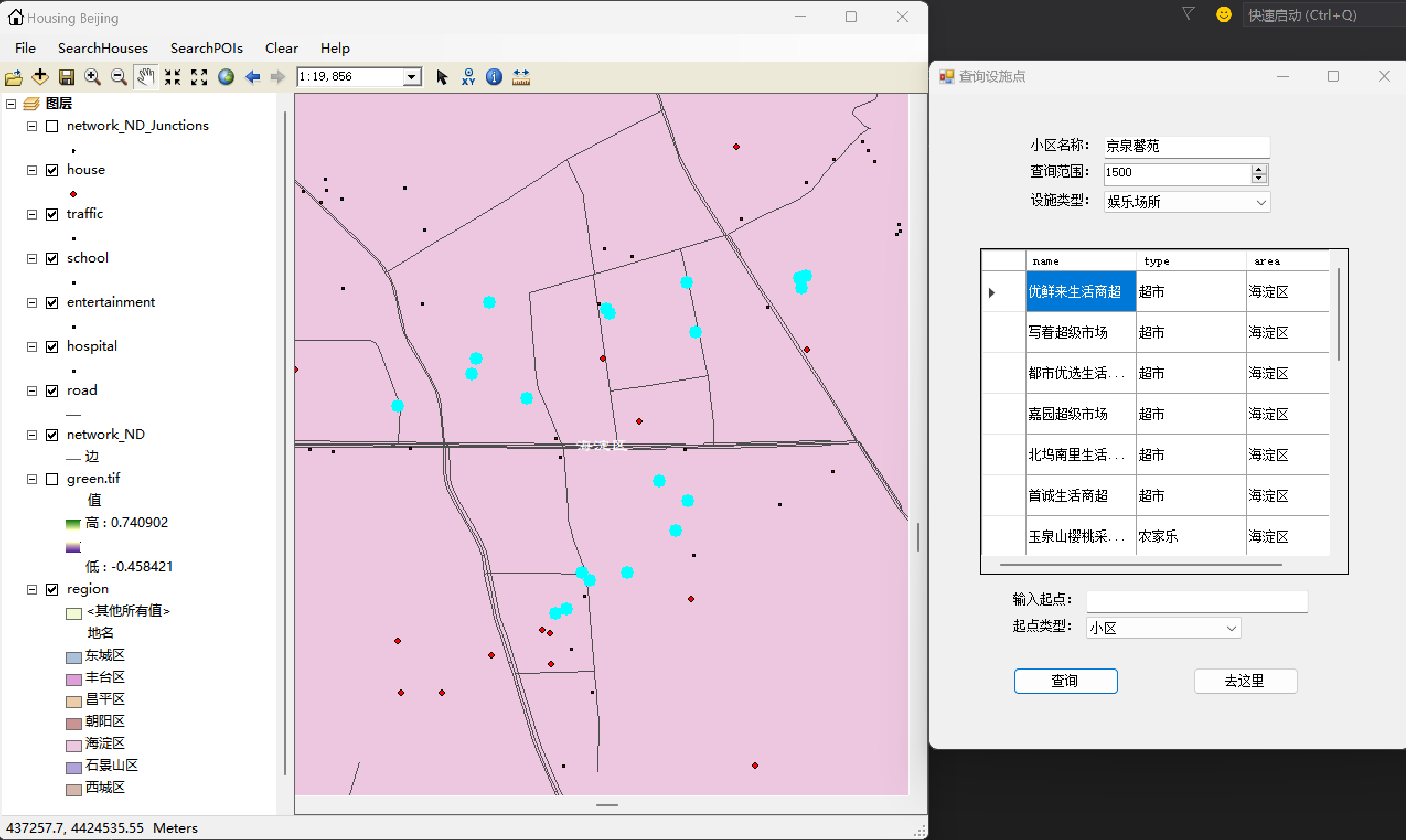
名称含“金”的小区

**（4）根据小区房价查询**



海淀区房价在5-10万的小区

**（5）查询周边设施点**



京泉馨苑1.5千米以内的娱乐场所

* 1. **项目开源**

本项目源码与数据已上传至<https://github.com/PKUtheshy/HousingBeijing>，读者可自行参考。

1. **总结**
   1. **开发工作评价**
2. **对生产效率的评价**

前期通过python爬虫获取链家房源数据、高德地图POI数据，程序简单但由于不稳定性时常中断，五一假期前从开始编写到完全爬取用时两周完成。

本系统以 VisualStudio 为开发平台，基于ArcEngine提供的控件与接口、.NetFramework 框架进行编写 Winform 项目。空间查询模块代码量为500行，其他模块代码量在300行左右，总体项目代码量在1500行左右，平均每个模块用一周完成编写、测试。整体调试在最后一周完成。因时间仓促，未能成功实现路径规划模块。总体来说，代码编写工作多从五一假期开始，生产效率中等。

1. **对产品质量的评价**

本系统为桌面程序，在测试过程中，没有发现明显的程序卡顿和崩溃现象。这得益于两方面的原因。一方面程序中的对象都及时通过Dispose方法释放内存；另一方面，经过预处理后数据本身规模不大，运行速度较快。但在其他设备使用本系统时，因地图存储采用绝对路径，需要修改默认地图路径。

1. **对技术方法的评价**

本项目采用基于ArcEngine的C# .Net Framework开发，相关组件可直接调用。网络上有许多相关的开发教程与资源，技术路线比较成熟，开发比较顺利。

* 1. **总结与收获**

ArcEngine提供了成熟的控件与类库，但实际开发中，我细读了官方文档与相关教程。课堂实践中许多类的编写与命名都能在ArcObjects封装的类中看到参考，比如Geometry、Feature、Layer等，二次开发的经历对于自主开发GIS系统也有很大的帮助。

虽然之前有ArcEngine的使用经验，但处理的是栅格数据，对于矢量数据的处理还是走了不少弯路。开发过程中，应该作好开发计划、合理安排时间与控制进度，预留出系统测试的时间，才能产出成熟的产品。