

基于三维 GIS 引擎的 CAD 图元数据结构研究及基础绘图功能研发

戴望宇^{1,2} 徐益飞³ 赵飞³ 邱瑞成³ 杨万理^{1,2}

(1. 西南交通大学桥梁智能与绿色建造全国重点实验室, 成都 611756;

2. 西南交通大学土木工程学院, 成都 610031;

3. 四川省交通勘察设计研究院有限公司, 成都 610017)

【摘要】经天路图是基于三维 GIS 系统的公路综合设计平台, 为了使用户在该软件的使用过程中, 准确表达设计理念及思想, 需要开发在三维地球场景下的精确绘制图形功能。本文基于图新地球 LocaSpace 开发工具包, 使用 C# 语言在 Visual Studio 2019 平台上开发了基础图元(点、线、多段线、圆、圆弧)绘制功能, 以及特殊点(端点、图元之间的交点、垂足等)捕捉以及正交绘图、延长线追踪等辅助绘图功能。文中介绍了基础绘图功能的研发思路、工具、流程及关键算法, 并展示了绘图功能的工程应用。结果表明, 本文开发的基础绘图功能可以便捷、快速、准确地帮助用户表达设计理念和思路, 提升设计效率。

【关键词】BIM; GIS; CAD; 二维绘图; 程序开发; 插件; 数据结构

【中图分类号】TU17

【文献标识码】A

引言

经天路图是基于国产三维引擎图新地球(三维 GIS 平台)研发的 BIM 软件, 具备快速 BIM 建模、公路选线设计等功能。但缺乏精确绘制基础图形和辅助绘图功能, 给设计人员带来诸多不便。

国外在绘图软件和绘图插件方面的研究起步很早。例如, Ivan E.Sutherland^[1]开发的 Sketchpad 系统, 它允许用户通过线描与计算机进行快速交流, 对后来的计算机辅助设计(CAD)和人机交互界面的发展产生了深远的影响。国内学者在二维绘图程序开发上同样历经了许多尝试。丁根芽等^[2]基于 Fortran 语言, 开发了交互式二维绘图系统。蔡汉明等^[3]构建了支持多种图元绘制的 IBM-PC 二维绘图系统, 但该系统功能完备性和稳定性仍需继续完善提高。李岩^[4]实现了二维图形参数化设计, 但研究成果尚未得到应用。林仁勇等^[5]研发了集成化闸门 CAD 系统。梅中义等^[6]研究了三维

模型与二维图形之间的双向关联和动态修改能力, 提高了工程图生成的效率和准确性。董树勇等^[7]、王盛智等^[8]、谭姝^[9]、解杨华等^[10]分别基于 AutoCAD 平台开发了铸造工艺、参数化设计及专业工程系统。此外, 叶勇进等^[11]研究了使用 DirectDraw 的二维绘图方法。张玉叶^[12]研究了 Matlab 曲线族绘制技术。李建朋等^[13]开发了机械设计辅助工具。吴凯伟^[14]提出了一种基于 Visual LISP 集成开发环境和 AutoLISP 语言的 AutoCAD 二次开发方案。张亚宁等^[15]实现了数据驱动的二维图形生成系统, 但仍存在交互功能欠缺、工程适配性不足等问题。

综上所述, 在二维绘图程序研发方面, 国外相对成熟, 但国内的二维绘图程序仍然存在短板, 例如辅助绘图功能不够便捷, 且工程应用程度还需进一步提高。本文将为经天路图 BIM 软件开发二维基础绘图和辅助绘图功能插件, 方便设计人员在 BIM 软件中高效开展设计工作。

【基金项目】四川省科技计划项目(2023YFS0429)

【第一作者】戴望宇(2000-), 男, 在读硕士研究生, 主要研究方向: 桥梁BIM, 桥梁水动力学。

【通信作者】杨万理(1979-), 男, 教授, 西南交通大学图学及BIM研究所所长, 主要研究方向为桥梁BIM, 桥梁水动力学等。

1 研究方法与技术路线

1.1 研发思路

本文拟开发的图形插件具体功能为基础图元的绘制,包括点、直线、多段线、圆、圆弧;以及辅助绘图功能,包括正交辅助绘图、N-E 正交辅助绘图、延长线绘制;同时还支持特殊点捕捉功能,包括静态特殊点如端点、中点、圆心和动态特殊点如图元之间的交点;并通过命令输入与快捷键提高用户操作效率。开发过程中重点确保各功能模块的稳定性和交互流畅性,以提供高效、便捷的绘图体验。研发思路如图 1 所示。

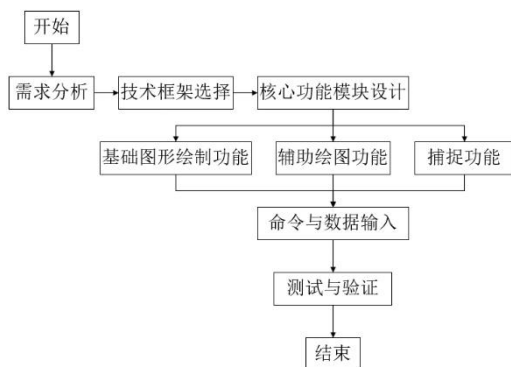


图 1 图形插件研发思路

1.2 开发环境与工具

绘图插件基于 Visual Studio 2019(communit)开发平台,选择 C#语言作为编程语言,采用.NET Framework 4.6.2 环境进行编译和调试。实施过程中借助中科图新 LocaSpaceSDK 进行绘图功能开发,在程序开发前先添加引用 LocaSpaceSDK 到项目中并引用其命名空间,在 WinForms 窗口中添加该控件并编写代码实现其指定功能。

1.3 二维绘图插件与经天路图之间的交互

拟研制的二维绘图插件相对独立,封装了多个图元绘制、辅助绘图以及特殊点计算等代码。插件通过接收外部传入的参数,进行数学计算,并将计算结果返回给外部程序。以绘制圆为例,经天路图将圆心、半径坐标传入绘图插件,绘图插件根据这两个点的坐标,计算出圆弧拟合线段各点坐标,并将其返回给经天路图,随后调用多段线绘制功能依次连接圆上各个点,完成圆的拟合绘制。

1.4 图元数据结构

图元绘制过程中,图元捕捉功能需频繁遍历坐标点数据,采用数组存储可提升访问效率。基础图元由点、直线段(圆和圆弧采用多段线近似)

组成,共同属性包含点坐标、样式如线型、线宽、颜色等。不同图元的数据结构按几何特征差异化定义,如点由一对坐标,直线段由两端点坐标,多段线由有序端点坐标组,圆弧由圆心坐标、开始角度和终止角度表示等。

CDraw 类作为所有图元的基类,描述图元的共同属性,包括点的坐标、图元显示、图元操作如绘制,以及图元操作公共函数如图元之间距离计算、直线斜率计算、图元 ID 获取等,为图元的查询、调用提供支持,也为利用 C#的封装、继承、多态性来管理图元提供方便。从基类 CDraw 中派生出五个图元子类:点类 CPoint,直线类 Cline,多段线类 CPLine,圆类 CCircle,圆弧类 CArc。在各个子类中增加数据特性描述,比如 CPoint 子类增加了 GsoPoint3d 型的成员变量 m_position 来记录点的经纬度坐标;CArc 子类中增加成员变量记录圆弧的圆心坐标、半径、开始角度和结束角度以及用于拟合圆弧的多段线端点的经纬度和笛卡尔坐标。同时,子类除了对基类中虚函数重载之外,还增加了完成该类对象各种操作所必需的成员函数,如每个图元的信息获取与设置,直线段特征边界的获取等。

2 基础图元绘制功能开发

本文使用 LocaSpaceSDK 开发的图形插件中,图元绘制功能的实现流程为:首先,将鼠标所在位置的客户端坐标(Point)转换为数字地球中的经纬度坐标(GsoPoint3d);其次,将经纬度坐标存入点的数组(GsoPoint3ds),并将点的数组(GsoPoint3ds)传入 LocaSpaceSDK 的底层图元对象,如点对象 GSOGeoPoint3d、多段线对象 GSOGeoPolyLine3d 等,随后,将图元对象传入 GsoFeature 对象,并将 GsoFeature 对象添加到图层中,完成图元的显示,如图 2 所示。

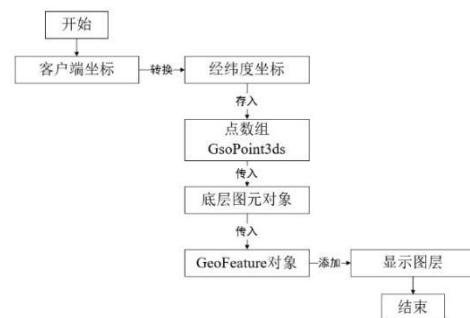


图 2 绘制功能流程

本节将以直线/多段线和圆/圆弧为例,介绍基

基础图元绘制功能的开发过程。

2.1 直线与多段线的绘制

直线和多段线的绘制中调用的基础图元对象是 GSOGeoPolyLine3d。在直线类和多段线类 Cline 以及 CPLine 中,保存了坐标对象 GsoPoint3ds、基础图元对象 GSOGeoPolyLine3d,以及图元显示对象 GSOFeature。图元数据分别存储在 CDraw 类的子类 Cline 类和 CPLine 中。直线和多段线的绘制分别如图 3~图 4 所示。



图 3 直线的绘制



图 4 多段线的绘制

2.2 圆与圆弧的绘制

圆与圆弧的绘制使用多段线进行拟合,分别在 CCircle 和 CArc 类中封装了计算圆与圆弧上各个点坐标的函数。绘制圆时将圆心以及圆上任意一点的坐标,传入 calCircle 函数计算得到多段线各端点的坐标,并使用多段线图元 GSOGeoPolyline3d 进行拟合。圆的绘制如图 5 所示。

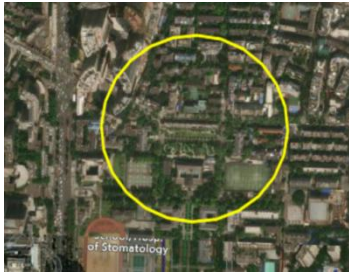


图 5 圆的绘制

绘制圆弧时,将圆心、圆上任一点以及圆心与圆弧起点、圆弧终点相连射线上任意点的坐标,传入 calArc 函数计算得到圆弧上其他点的坐标,并使用多段线图元 GSOGeoPolyline3d 进行拟合。圆弧的绘制如图 6 (a)~(b) 所示。



(a)确定起始角度



(b)确定终止角度

图 6 圆弧的绘制

3 辅助绘图功能开发

3.1 命令与数据的输入

为实现精确绘图,插件通过键盘-鼠标协同交

互完成指令与数据输入。利用 Winform 的消息处理机制,使用函数 GlobeControl_KeyDown (object sender, KeyEventArgs e) 监视键盘非系统键动作,获取键盘命令。在绘图窗口底部增加 textbox 对象 tb_input 的文本框,通过重载函数 tb_input_KeyPress(object sender, KeyPressEventArgs e) 获取文本框输入的数值或坐标数据。输入数据前按下空格键时,通过重载 GlobeControl_KeyDown 函数,将焦点切换至文本框,并将键盘消息传递给 inputDataCal(double inputData) 函数进行处理。

插件对键盘输入信息的判断逻辑如图 7 所示。

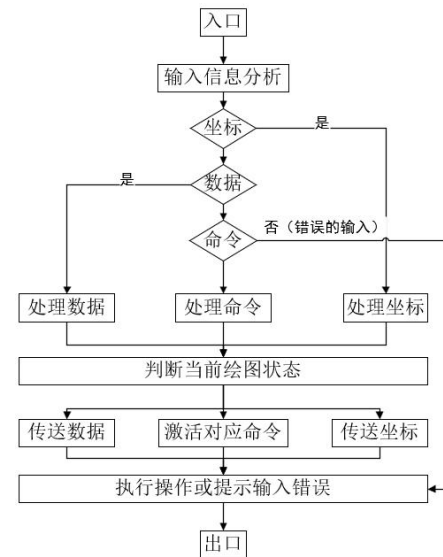


图 7 键盘输入的处理模块

首先对用户输入的信息进行分析,判断是绘图短命令、数据还是坐标。如果是坐标,那么使用文本框的响应函数,根据逗号“,”分割两部分数据,并提取出坐标的 X、Y 值;如果是数据,那么使用函数 inputDataCal(double inputData) 对数据进行判断处理,并根据当前的绘制模式进行响应;如果是命令,那么激活相应的命令。

插件对输入信息的类别进行判定之后,根据当前的绘图状态做出响应。比如正在绘制圆弧,首先通过鼠标点击或坐标点的输入确定圆心,随后根据圆心点与当前鼠标位置确定半径,并绘制橡皮圆,之后输入起始角度和终止角度。这个角度可以是角度制的精确值,也可以是通过鼠标拖动捕捉到的点,最终通过函数 GSOPoint3ds FitArc(GSOPoint3d centerInGeo, GSOPoint3d radiusPtInGeo, GSOPoint3d startInGeo, GSOPoint3d endInGeo) 计算确定。

3.2 光标靶区的建立

为实现无极缩放场景下的精确捕捉,研究针对

LocaSpaceSDK 数字地球的经纬度坐标特性 (double 类型), 优化靶区判定规则。原 Microsoft Rectangle 类的 Contains(int x, int y)函数, 因整型参数限制, 无法适配经纬度微小变化 (1°≈88-110km) 导致的高精度需求。为此, 重构靶区构建逻辑: 以屏幕光标为中心生成边长为 20 像素 (2n, 默认 n=10) 的正方形区域, 并通过自定义的 bool contains(GSOPoint3d ptInGeo, GSOPoint3d ptLB, GSOPoint3d ptRT)函数直接对比 double 型经纬度坐标, 判定目标点是否位于靶区范围内, 从而解决缩放过程中的捕捉精度与灵敏度矛盾问题。

3.3 特殊点捕捉

特殊点分为两类, 静态特殊点: 图元固有特征点 (如端点、中点、圆心、象限点及几何交点/切点/垂足); 动态特殊点: 绘制过程中实时生成的关联点 (如橡皮筋线与现有图元的交点/切点)。

捕捉时, 程序遍视口内所有特殊点坐标 (存储于 GsoPoint3d[] m_PLEnd 数组), 特殊点被靶区覆盖, 则以特定符号高亮标记 (例如端点使用正方形框、中点使用三角形框等), 并通过 int n_End 计数器与布尔变量 (如 bool_EndPoint) 管理捕捉状态。

3.4 正交绘图

正交绘图功能通过限制光标沿 X/Y 轴移动, 实现图元轴向对齐。以直线段绘制为例, 算法以鼠标首次单击时, 光标所在点(x1,y1)为起点, 建立直角坐标系, 如图 8 所示; 两条角平分线(虚线)将坐标系分为四个区域, 光标移动时, 根据所在区域向对应坐标轴投影生成直线终点(x2,y1)。正交绘制模式通过函数 GSOPoint3d InterCrossPoint(GSOPoint3d PointOrign, GSOPoint3d Point)实现, 实现代码图 9 所示。

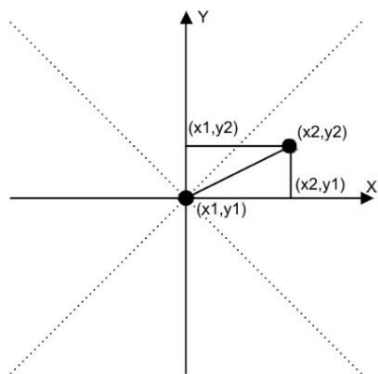


图 8 正交绘图原理

```
InterAng = GetAngle(PointOrign.X, PointOrign.Y, Point.X, Point.Y);
if (InterAng >= pi / 4 && InterAng <= pi * 3 / 4
|| InterAng >= pi * 5 / 4 && InterAng <= pi * 7 / 4)
{
    Point.X = PointOrign.X;
    if (m_InterceptLine == true)
    {
        if (Point.Y - PointOrign.Y > 0)
            Point.Y = PointOrign.Y + m_InterceptLen;
        else
            Point.Y = PointOrign.Y - m_InterceptLen;
    }
}
else
{
    Point.Y = PointOrign.Y;
    if (m_InterceptLine == true)
    {
        if (Point.X - PointOrign.X > 0)
            Point.X = PointOrign.X + m_InterceptLen;
        else
            Point.X = PointOrign.X - m_InterceptLen;
    }
}
```

图 9 正交绘图判断代码

4 求解交点关键算法

求解图元集合 P {直线段, 多段线, 圆, 圆弧} 中任意两个图元的交点, 共 10 种情况, 只需调用 3 个函数就可以完全解决。

设集合 $A=\{\text{直线段, 多段线}\}$; 若 CP 表示求解两个图形元素的交点, 数组 GSOPoint3d[] CP_point 用于存放交点坐标, 则 $CP_point=A_i CP A_j$, (i, j 为大于 0 小于 3 的整数) 表示集合 A 中任意两个图形元素的交点, 那么编写直线交点判断函数 CP_Line_Line(.....), 即把集 A 中元素都看作为直线段组成的, 集合中任意两个元素的交点归结为直线段之间的交点, 直线交点求解代码如图 10 所示。

```
k1 = (y2 - y1) / (x2 - x1);
k2 = (y4 - y3) / (x4 - x3);
b1 = (y1 * x2 - y2 * x1) / (x2 - x1);
b2 = (y3 * x4 - y4 * x3) / (x4 - x3);
if (k1 != k2)
{
    x = (b1 - b2) / (k2 - k1);
    y = k1 * x + b1;
    bool OnL1 = IsCPInLineZone(x1, y1, x2, y2, x, y);
    bool OnL2 = IsCPInLineZone(x3, y3, x4, y4, x, y);
    if (OnL1 && OnL2)
    {
        CP.X = x;
        CP.Y = y;
        return 1;
    }
    else
        return 0;
}
```

图 10 直线交点算法

设集合 $B=\{\text{圆, 圆弧}\}$, $CP_point= A_i CP B_j$, (i, j 为大于 0 小于 3 的整数) 表示集合 A 和 B 中, 两个图形的交点, 那么编写直线段与圆的交点函数 CP_Line_Circle(.....), 求解集合 A 、 B 中任意两个元素的交点归结为求解直线段和圆的交点。对于求解直线段和圆弧的交点, 先求解直线段和圆的交点, 再判断交点是否在弧上即可。

设集合 B 有子集 $C=\{\text{圆, 圆弧}\}$, 则 $CP_point= C_i CP C_j$ (i, j 为大于 0 小于 3 的整数) 表示集合 C 中

任意两个图形元素的交点，同理编写圆与圆的交点函数：CP_Circle_Circle(.....)，即求圆和圆的交点的函数是求解集合 C 中任两个图形素交点的通用函数。

5 工程应用

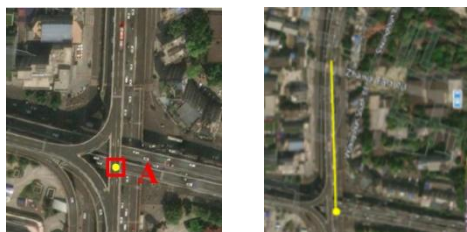
以成都二环高架路成温立交段为例，展示本文研制的基础图形绘制功能在公路设计过程的应用。

首先打开程序，顶部是功能按钮，从左到右依次为点、直线、多段线、圆、圆弧的绘制按钮，以及辅助绘图功能如正交、N-E 正交的勾选按钮，延长线的开启按钮，捕捉功能及具体捕捉点的勾选按钮，如图 11 所示。



图 11 程序界面

视口移动到成温立交段后，在拟建设起点 A 和终点 B 各绘制一个点，随后打开捕捉并勾选节点，即可将接下来要绘制的直线段的起点设置到已经绘制的点 A 上，如图 12 (a) 所示。然后绘制路线中的直线部分，如图 12 (b) 所示。



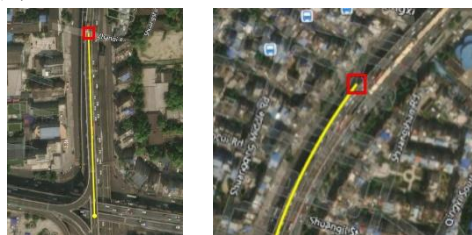
(a) 捕捉起点 (b) 绘制直线

图 12 直线段绘制

由于之后的路线是带有弧度的，所以需要打开圆弧绘制，并且勾选捕捉端点，即可将捕捉到的直线的端点设置为圆弧的起点，如图 13 (a) 所示。

圆弧绘制完成后是一段直线，所以选择直线绘制，同样保持勾选端点捕捉，将捕捉到的圆弧的端点设置为直线的起点，如图 13 (b) 所示。最后将直线终点捕捉到终点 B 上，即可完成绘制，如

图 14 所示。



(a) 捕捉直线终点 (b) 捕捉圆弧终点

图 13 圆弧段绘制



图 14 完整路线

绘制总用时在一分钟以内，为经天路图提供了有精度要求的基础图形绘制功能，让用户能够便捷地表达自己的设计理念和设计思路。

6 结论

基于三维 GIS 引擎的 CAD 图元数据结构研究及基础绘图功能开发，使用中科图新 LocaSpaceS DK，采用 C#语言在 VisualStudio 2019 平台上研发了经天路图的图形绘制功能插件。为经天路图提供了绘图功能。绘制功能包括点、直线、多段线、圆、圆弧的绘制；屏幕正交绘制、N-E 正交绘图及延长线绘制；特殊点的捕捉，包括端点、中点、圆心、象限点、交点、垂足、最近点及节点。

传统 CAD 二次开发插件中，几何计算受限于软件内核调度，自定义算法集成困难，本文通过独立封装图元基类及派生类，实现几何计算内核与主程序解耦，提升计算效率；开源 GIS 工具依赖于通用几何库的圆弧离散化算法，本文自主开发圆心角等分策略，通过多段线拟合圆弧，提升了绘制精度；同时本文直接基于 LocaSpaceSDK 开发，保留了原始设计参数，与商业 GIS 插件相比，避免了传统格式转换可能导致的数据丢失问题。

实例验证表明，本文开发的图形绘制程序能够便捷、快速、准确地帮助用户能表达设计理念和思路。但是由于绘制坐标系为经纬度坐标系，绘制的便捷程度不如二维笛卡尔坐标系，后续可在该方面开展进一步研究。

参考文献

- [1] Sutherland I E. Sketch pad a man-machine graphical communication system[C]//Proceedings of the SHARE design automation workshop. 1964: 6.329-6.346.
- [2] 丁根芽,潘杰.交互式二维绘图系统的实现[J].微电子学

与计算机,1989(11):9-13.

[3] 蔡汉明,宋晓梅.IBM-PC 二维绘图系统的建立[J].青岛化工学院学报,1991(02):87-92.

[4] 李岩,王东戈,陈宜东. GEODRAW 二维图形软件剖析[J].哈尔滨科学技术大学学报,1996(06):64-66.

[5] 林仁荣,范永法,吴玉光,等.平面闸门 CAD 系统的模块化设计[J].河海大学常州分校学报,1999(04):15-18.

[6] 梅中义,曾令卫,吴斌.基于三维特征设计的二维工程图的自动生成[J].北京航空航天大学学报,2000(01):103-106.

[7] 董树勇,张希俊,柳百成.铸钢件铸造工艺工装 CAD 软件研究与开发[J].铸造技术,2001(01):29-31.

[8] 王盛智.基于 AutoLISP 语言的参数化设计绘图系统[J].抚顺石油学院学报,2003(01):69-72.

[9] 谭姝.参数化设计绘图系统[J].机械设计与制造,2005(08):

66-68.

[10] 解杨华.水环真空泵参数化设计与 CAD 系统开发[D].东北大学,2009.

[11] 叶勇进.在 C++ Builder 6.0 中使用 DirectDraw 进行二维绘图[J].电脑编程技巧与维护,2010(19):73-74.

[12] 张玉叶.Matlab 二维绘图数据源分析及曲线族绘制[J].重庆科技学院学报(自然科学版),2011,13(06):189-192.

[13] 李建朋.基于 ObjectARX 的机械设计辅助工具系统的开发[D].青岛大学,2013.

[14] 吴凯伟, 石玮莹. 基于 Visual LISP 的地铁轨道综合图辅助设计程序开发[J]. 铁道标准设计, 2017, 61(5): 27-31.

[15] 张亚宁,刘春光,吴诗帆.基于 VC++的二维绘图程序设计[J].科技与创新,2019(17):4849+51.

Research on CAD Primitive Data Structure Based on 3D GIS Engine and Development of Basic Drawing Functions.

Dai Wangyu^{1,2}, Xu Yifei³, Zhao Fei³, Qiu Ruicheng³, Yang Wanli^{1,2}

(1. State Key Laboratory of Bridge Intelligent and Green Construction, Southwest Jiaotong University, Chengdu 611756, Sichuan, China;

2. School of Civil Engineering, Southwest Jiaotong University, Chengdu 610031, Sichuan, China;

3. Sichuan Communications Surveying and Design Institute Co., Ltd., Chengdu 610017, Sichuan, China)

Abstract: Jingtianlutu is a comprehensive road design platform based on a 3D GIS system. To enable users to accurately express their design concepts and ideas during the use of the software, it is necessary to develop drawing functions with precision requirements in a 3D Earth environment. In this study, based on the LocaSpace development toolkit of Tuxin, the basic primitive drawing functions (points, lines, polylines, circles, arcs) and special point snapping functions (e.g., endpoint snapping, intersection snapping, perpendicular snapping), as well as auxiliary drawing functions such as orthogonal drawing and extension line tracking are developed on the platform of Visual Studio 2019 by using C# language. This study introduces the research and development content, detailed design ideas, development tools, development process, and key algorithms behind the drawing functions. The results show that the developed basic drawing functions enable users to express design concepts conveniently, quickly, and accurately, thereby improving design efficiency.

Keywords: BIM; GIS; CAD; 2D Drawing; Program Development; Plugin; Data Structure