第12章

三维模式下数据编辑功能实现

12.1 简介

ArcScene 和 ArcGlobe 控件在第 2 章中已做过介绍。本章直接用一个实例来展示这两个控件在数据编辑中新增功能的实现。更新和删除功能使用的接口和类都一样，在新增中使用 AddElement，在更新和删除中分别替换为 UpdateElement 和 DeleteElement。

12.2 ArcScene 控件下的数据编辑

在 ArcEngine 中，为 3D 应用提供了一个管理图形的容器 IGraphicsContainer3D，类似于两维情形中的 IGraphicsContainer，本节的操作主要也是应用该接口来实现。该接口提供了新增、更新、删除的方法，如 AddElement 方法用于往容器中增加一个元素；DeleteElement 方法用于删除容器中指定的元素；DeleteAllElements 方法用于删除容器中所有的元素，该方法一般用于清空操作。

12.3 ArcGlobe 控件下的数据编辑

GlobeControls 中的图层是依照它们的类型进行分类的，通过它们的 elevation、draped 或 floating 来进行分类。当然充当 Elevation layers 的栅格图层要有高程信息，利用 features 图层或栅格数据的高程数据来运用到 elevation 图层中，运用 IGlobe::AddlayerType 来增加指定类型到 GlobeControl 中，这些类型只是下面几种 esriGlobeLayerType：

esriGlobeLayerTypeUnknown 0 Add layer as base image。

esriGlobeLayerTypeDraped 1 Add layer as base image。

esriGlobeLayerTypeElevation 2 Add layer as elevation。

esriGlobeLayerTypeFloating 3 Add layer as floating。

ArcGlobe 控件完全可以实现 ArcScence 控件的所有功能。需要注意的是：ArcGlobe 控件不能通过 Refresh()方法，也不能使用 Globe.GlobeDisplay.RefreshViewers()方法进行刷新，这两个方

式的刷新都不起作用。而应使用 IGraphicsContainer 的 RefreshLayer 对图层进行刷新。

下面用一个简单的实例来演示这两个控件的使用。

创建一个新 Windows 应用程序项目，命名为“lesson1”，在窗体的左边放一个工具栏“ToolBarControl”控件，一个 SceneControl 控件，设置工具栏控件的“伙伴控件”为 SceneControl。在窗体的右边放一个 ToolBarControl 控件，一个 GlobeControl 控件。设置右边工具栏控件的“伙伴控件”为 GlobeControl。在两个工具栏控件中各添加一些对应的常用操作工具项。在窗体的中部放 5 个 Button 控件，然后修改控件的 Text 属性，如图 12-1 所示。

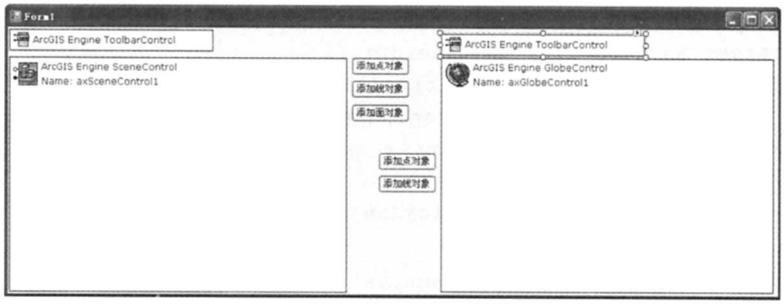


图 12-1　应用程序项目 lesson1 简体

在代码窗口中添加如下程序。

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using System.IO;
using ESRI.ArcGIS.DataSourcesRaster;
using ESRI.ArcGIS.Geodatabase;
using ESRI.ArcGIS.Geometry;
using ESRI.ArcGIS.Analyst3D;
using ESRI.ArcGIS.Carto;
using ESRI.ArcGIS.DataSourcesFile;
using ESRI.ArcGIS.Display;
using ESRI.ArcGIS.GlobeCore;

namespace lesson3
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        //SceneControl 参数
        string geoType = "";                          //类型
        IPointCollection m_pointCollection;           // 新增点集合
        ISegmentCollection m_polyline;                //新增线
```

```
            //GlobeControl参数
            string globeGeoType = "";                    //类型
            IPointCollection m_GlobePointCollection;     // 新增点集合
            ISegmentCollection m_GlobePolyline;          //新增线
            //在SceneControl控件上新增点
            private void btPoint_Click(object sender, EventArgs e)
            {
                geoType = "point";
                m_pointCollection = new MultipointClass();
                IGraphicsContainer3D graphicsContainer3D = this.axSceneControl1.Scene.
BasicGraphicsLayer as IGraphicsContainer3D;
                graphicsContainer3D.DeleteAllElements();
                this.axSceneControl1.Scene.SceneGraph.RefreshViewers();
            }
            //在SceneControl控件上新增线
            private void btPolyline_Click(object sender, EventArgs e)
            {
                geoType = "polyline";
                m_pointCollection = new PolylineClass();
                m_polyline = new PolylineClass();
                IGraphicsContainer3D graphicsContainer3D = this.axSceneControl1.Scene.
BasicGraphicsLayer as IGraphicsContainer3D;
                graphicsContainer3D.DeleteAllElements();
                this.axSceneControl1.Scene.SceneGraph.RefreshViewers();
            }
            //在SceneControl控件上新增面
            private void btPolygon_Click(object sender, EventArgs e)
            {
                geoType = "polygon";
                m_pointCollection = new PolygonClass();

                IGraphicsContainer3D graphicsContainer3D = this.axSceneControl1.Scene.
BasicGraphicsLayer as IGraphicsContainer3D;
                graphicsContainer3D.DeleteAllElements();
                this.axSceneControl1.Scene.SceneGraph.RefreshViewers();
            }
            //SceneControl控件上鼠标单击事件，记录单击位置的坐标，并将其添加到点集中
            private void axSceneControl1_OnMouseDown(object sender, ESRI.ArcGIS.
Controls.IsceneControlEvents_OnMouseDownEvent e)
            {
                ISceneGraph sceneGraph = this.axSceneControl1.SceneGraph;
                ISceneViewer sceneViewer = sceneGraph.ActiveViewer;
                IPoint point;
                object pOwner, pObject;
                object pBefore = Type.Missing;
                object pAfter = Type.Missing;
                //右键取消新增几何对象操作
                if (e.button == 2)
                {
                    geoType = "";
                }
```

```
            //左键，将点添加进点集中，
            else if (e.button == 1)
            {
                //if (geoType == "point")
                //{
                //    sceneGraph.Locate(sceneViewer, e.x, e.y, esriScenePickMode.
esriScenePickAll, true, out  point, out  pOwner, out pObject);
                //    m_pointCollection.AddPoint(point, ref pBefore, ref pAfter);
                //}
                //else if (geoType == "polyline")
                //{
                //    sceneGraph.Locate(sceneViewer, e.x, e.y, esriScenePickMode.
esriScenePickAll, true, out point, out pOwner, out pObject);
                //    m_pointCollection.AddPoint(point, ref pBefore, ref pAfter);
                //}
                //else if (geoType == "polygon")
                //{
                sceneGraph.Locate(sceneViewer,  e.x,  e.y,  esriScenePickMode.
esriScene PickAll, true, out point, out pOwner, out pObject);
                m_pointCollection.AddPoint(point, ref pBefore, ref pAfter);
                //}
            }
            //在控件上画出新增对象
            DrawGeometry();
        }
        //SceneControl 在控件上画出新增对象
        private void DrawGeometry()
        {
            if (geoType == "point")
            {
                IGraphicsContainer3D graphicsContainer3D = this.axSceneControl1.
Scene.BasicGraphicsLayer as IGraphicsContainer3D;
                graphicsContainer3D.DeleteAllElements();
                for (int i = 0; i < m_pointCollection.PointCount; i++)
                {
                    IPoint point = m_pointCollection.get_Point(i);
                    IZAware zAware = point as IZAware;
                    zAware.ZAware = true;
                    IGeometry pointGeometry = point as IGeometry;
                    pointGeometry.SpatialReference = this.axSceneControl1.Scene.
SpatialReference;
                    IElement element = new MarkerElementClass();
                    element.Geometry = pointGeometry;
                    graphicsContainer3D.AddElement(element);
                }
                this.axSceneControl1.Scene.SceneGraph.RefreshViewers();
            }
            else if (geoType == "polyline")
            {
                if (m_pointCollection.PointCount > 1)
                {
```

```
                    IRgbColor lineRgbColor = new RgbColorClass();
                    lineRgbColor.Blue = 255;
                    SimpleLineSymbol simpleLineSymbol = new SimpleLineSymbolClass();
                    simpleLineSymbol.Color = lineRgbColor;
                    simpleLineSymbol.Width = 3;

                    ILine line = new LineClass();
                    IPoint fromPoint = m_pointCollection.get_Point(m_pointCollection.
PointCount - 2);
                    IPoint toPoint = m_pointCollection.get_Point(m_pointCollection.
PointCount - 1);
                    line.PutCoords(fromPoint, toPoint);
                    object Missing1 = Type.Missing;
                    object Missing2 = Type.Missing;
                    m_polyline.AddSegment(line as ISegment, ref Missing1, ref
Missing2);
                    IZAware zAware = m_polyline as IZAware;
                    zAware.ZAware = true;
                    IGeometry lineGeometry = m_polyline as IGeometry;
                    lineGeometry.SpatialReference = this.axSceneControl1.Scene.
SpatialReference;

                    ILineElement lineElement = new LineElementClass();
                    lineElement.Symbol = simpleLineSymbol;

                    IElement element = lineElement as IElement;
                    element.Geometry = lineGeometry;
                    IGraphicsContainer3D graphicsContainer3D = this.axScene
Control1.Scene.BasicGraphicsLayer as IGraphicsContainer3D;
                    graphicsContainer3D.AddElement(element);
                    this.axSceneControl1.Scene.SceneGraph.RefreshViewers();
                }
            }
            else if (geoType == "polygon")
            {
                if (m_pointCollection.PointCount > 2)
                {
                    IGraphicsContainer3D graphicsContainer3D = this.axScene
Control1.Scene.BasicGraphicsLayer as IGraphicsContainer3D;
                    graphicsContainer3D.DeleteAllElements();
                    IPointCollection polygonCollection = new PolygonClass();
                    object Missing1 = Type.Missing;
                    object Missing2 = Type.Missing;
                    for (int i = 0; i < m_pointCollection.PointCount; i++)
                    {
                        polygonCollection.AddPoint(m_pointCollection.get_Point(i),
ref Missing1, ref Missing2);
                    }

                    IZAware zAware = polygonCollection as IZAware;
                    zAware.ZAware = true;
```

```
                IPolygon polygon = polygonCollection as IPolygon;
                polygon.SimplifyPreserveFromTo();
                IGeometry geometry = polygon as IGeometry;
                geometry.SpatialReference = this.axSceneControl1.Scene.Spatial
Reference;

                IPolygonElement polygonElement = new PolygonElementClass();
                IElement element = polygonElement as IElement;
                elementGeometry = geometry;
                graphicsContainer3D.AddElement(element);
                this.axSceneControl1.Scene.SceneGraph.RefreshViewers();
            }
        }

    }

    //窗体启动，并打开 SXD、3DD 文件初始化两个控件
    private void Form1_Load(object sender, EventArgs e)
    {
        this.openFileDialog1.Title = "sxd 文件";
        this.openFileDialog1.DefaultExt = ".sxd";
        this.openFileDialog1.Filter = "(*.sxd)|*.sxd";
        string fileName = "";
        string globeFileName = "";
        if (this.openFileDialog1.ShowDialog() == DialogResult.OK)
        {
            fileName = this.openFileDialog1.FileName;
            this.axSceneControl1.LoadSxFile(fileName);
        }


        this.openFileDialog1.Title = "sxd 文件";
        this.openFileDialog1.DefaultExt = ".3dd";
        this.openFileDialog1.Filter = "(*.3dd)|*.3dd";
        if (this.openFileDialog1.ShowDialog() == DialogResult.OK)
        {
            globeFileName = this.openFileDialog1.FileName;
            this.axGlobeControl1.Load3dFile(globeFileName);
        }
    }
    //在 GlobeControl 控件上添加点
    private void btAddPoint_Click(object sender, EventArgs e)
    {
        globeGeoType = "point";
        m_pointCollection = new MultipointClass();
        IScene scene = this.axGlobeControl1.Globe as IScene;

        IGraphicsContainer3D graphicsContainer3D = scene.BasicGraphicsLayer as
IGraphicsContainer3D;
        graphicsContainer3D.DeleteAllElements();
```

```
            IGraphicsContainer globeGraphicsLayer = new GlobeGraphicsLayerClass();
            ILayer layer = globeGraphicsLayer as ILayer;
            layer.Name = "point";

            scene.AddLayer(layer, true);
            this.axGlobeControl1.Globe.GlobeDisplay.RefreshViewers();
        }
        //在 GlobeControl 控件上添加线
        private void btAddLine_Click(object sender, EventArgs e)
        {
            globeGeoType = "polyline";
            m_pointCollection = new PolylineClass();
            IScene scene = this.axGlobeControl1.Globe as IScene;

            IGraphicsContainer3D graphicsContainer3D = scene.BasicGraphicsLayer as
IGraphicsContainer3D;
            graphicsContainer3D.DeleteAllElements();
            m_GlobePolyline = new PolylineClass();
            IGraphicsContainer globeGraphicsLayer = new GlobeGraphicsLayerClass();
            ILayer layer = globeGraphicsLayer as ILayer;
            layer.Name = "polyline";

            scene.AddLayer(layer, true);
            this.axGlobeControl1.Globe.GlobeDisplay.RefreshViewers();
        }

        //GlobeControl 鼠标单击时间，将点添加进点集
        private void axGlobeControl1_OnMouseDown(object sender, ESRI.ArcGIS.
Controls.IGlobeControlEvents_OnMouseDownEvent e)
        {
            IGlobeViewUtil globeViewUtil = this.axGlobeControl1.GlobeCamera as
IGlobeViewUtil;
            IGlobeDisplay globeDisplay = this.axGlobeControl1.GlobeDisplay;


            IPoint point;
            object pOwner, pObject;
            object pBefore = Type.Missing;
            object pAfter = Type.Missing;
            double dLon, dLat, dAlt;


            if (e.button == 2)
            {
                globeGeoType = "";
            }
            else if (e.button == 1)
            {
                if (globeGeoType == "point")
                {
                    point = new PointClass();
```

```
                    globeViewUtil.WindowToGeographic(globeDisplay, globeDisplay.
ActiveViewer, e.x, e.y, true, out dLon, out dLat, out dAlt);
                    point.PutCoords(dLon, dLat);
                    m_pointCollection.AddPoint(point, ref pBefore, ref pAfter);
                }
                else if (globeGeoType == "polyline")
                {
                    point = new PointClass();
                    globeViewUtil.WindowToGeographic(globeDisplay, globeDisplay.
ActiveViewer, e.x, e.y, true, out dLon, out dLat, out dAlt);
                    point.PutCoords(dLon, dLat);
                    m_pointCollection.AddPoint(point, ref pBefore, ref pAfter);
                }
            }
            //在 GlobeControl 控件上画出新增对象
            GlobeDrawGeometry();
        }
        //在 GlobeControl 控件上画出新增对象
        private void GlobeDrawGeometry()
        {
            IRgbColor rgbColor = new RgbColorClass();
            rgbColor.Red = 255;

            IGraphicsContainer globeGraphicsLayer = null;
            ILayer layer = null;
            IScene scene = this.axGlobeControl1.Globe as IScene;
            if (globeGeoType == "point")
            {
                for (int k = 0; k < scene.LayerCount; k++)
                {
                    if (scene.get_Layer(k).Name == "point")
                    {
                        globeGraphicsLayer = scene.get_Layer(k) as IGraphics
Container;
                    }
                }
                layer = globeGraphicsLayer as ILayer;
                IElement markerElement = new MarkerElementClass();
                ISimpleMarker3DSymbol simpleMarker3DSymbol = new SimpleMarker
3DSymbolClass();
                simpleMarker3DSymbol.Style = esriSimple3DMarkerStyle.esriS 3DMSCone;

                simpleMarker3DSymbol.ResolutionQuality = 1;

                IMarkerSymbol markerSymbol = simpleMarker3DSymbol as Imarker Symbol;
                markerSymbol.Size = 1000000;
                markerSymbol.Color = rgbColor;

                for (int i = 0; i < m_pointCollection.PointCount; i++)
```

```
                {
                    markerElement.Geometry = m_pointCollection.get_Point(i);
                    IMarkerElement markerElement2 = markerElement as Imarker Element;
                    markerElement2.Symbol = markerSymbol;
                    globeGraphicsLayer.AddElement(markerElement, 1);
                }
                IGlobeDisplayLayers2  pGDispLyrs  =  this.axGlobeControl1.Globe
Display as IglobeDisplayLayers2;
                pGDispLyrs.RefreshLayer(layer);


            }
            else if (globeGeoType == "polyline")
            {
                if (m_pointCollection.PointCount > 1)
                {
                    for (int k = 0; k < scene.LayerCount; k++)
                    {
                        if (scene.get_Layer(k).Name == "polyline")
                        {
                            globeGraphicsLayer = scene.get_Layer(k) as Igraphics
Container;
                        }
                    }
                    layer = globeGraphicsLayer as ILayer;

                    IElement lineElement = new LineElementClass();
                    ISimpleLine3DSymbol  simpleLine3DSymbol  =  new  SimpleLine3D
SymbolClass();
                    simpleLine3DSymbol.Style = esriSimple3DLineStyle.esri S3DLSStrip;

                    simpleLine3DSymbol.ResolutionQuality = 1;

                    ILineSymbol lineSymbol = simpleLine3DSymbol as ILineSymbol;
                    lineSymbol.Color = rgbColor;
                    lineSymbol.Width = 5;

                    ILine line = new LineClass();
                    IPoint fromPoint = m_pointCollection.get_Point(m_point
Collection. PointCount - 2);
                    IPoint toPoint = m_pointCollection.get_Point(m_pointCollection.
PointCount - 1);
                    line.PutCoords(fromPoint, toPoint);
                    object Missing1 = Type.Missing;
                    object Missing2 = Type.Missing;
                    m_GlobePolyline.AddSegment(line as ISegment, ref Missing1, ref
Missing2);
                    IPolyline polyline = m_GlobePolyline as IPolyline;
                    lineElement.Geometry = polyline;

                    ILineElement lineElement2 = lineElement as ILineElement;
```

```
                    lineElement2.Symbol = lineSymbol;
                    globeGraphicsLayer.AddElement  (lineElement, 1);
                    IGlobeDisplayLayers2 pGDispLyrs = this.axGlobeControl1.Globe
Display as IGlobeDisplayLayers2;
                    pGDispLyrs.RefreshLayer(layer);


                }
            }
        }
    }
}
```

程序运行效果如图 12-2、图 12-3、图 12-4、图 12-5 和图 12-6 所示。

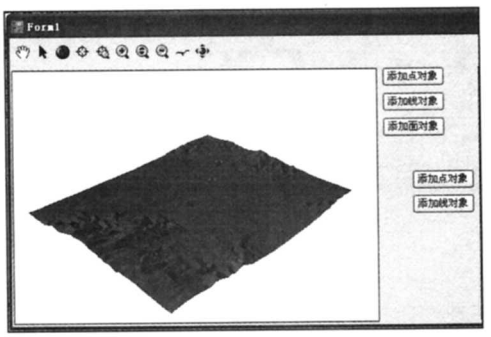


图 12-2　在 SceneControl 上添加点对象

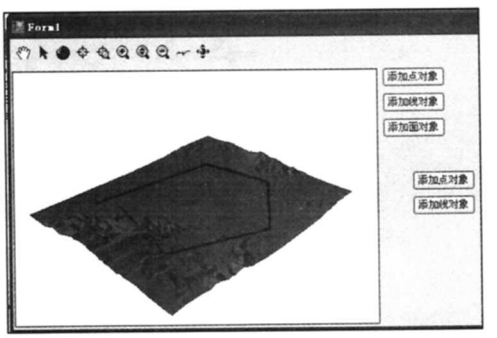


图 12-3　在 SceneControl 上添加线对象

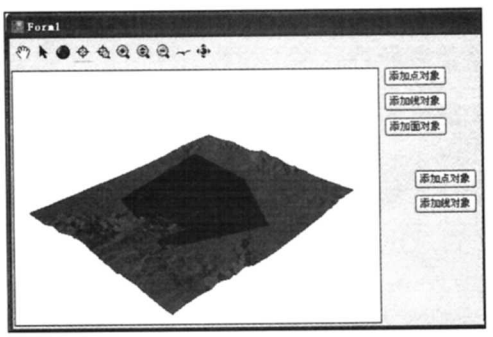


图 12-4　在 SceneControl 上添加面对象

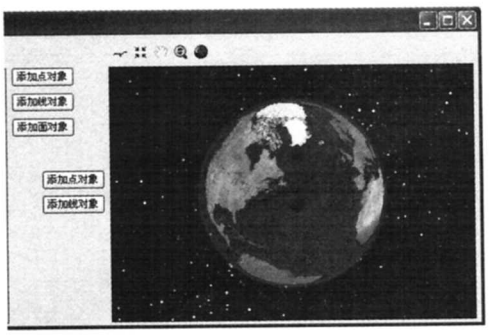


图 12-5　在 GlobeControl 控件上添加点

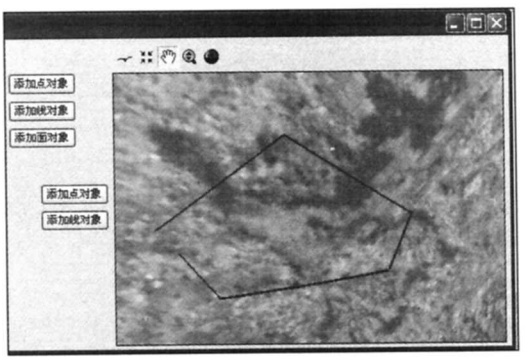


图 12-6　在 GlobeControl 控件上添加线

12.4 本章小结

本章介绍了三维控件的编辑，通过一个实例介绍了两个三维空间数据编辑的功能，三维编辑不是本书的重点，这里只做了介绍，需要深入了解的读者可查看相关书籍。