

# 基于 ArcGIS VBA 的动态插值帧的自动生成

程真 黄成 陈长虹

(上海市环境科学研究院, 上海, 200233)

**摘要** 借助 ArcGIS VBA 二次开发了动态插值帧的自动生成程序, 实现了离散地理数据空间插值的时态 GIS 可视化表达。通过该程序自动获得逐时刻的空间动态插值帧, 为进一步生成动画文件奠定基础, 可作为 ArcGIS9.2 时态 GIS 可视化表达的一个有益补充完善。

**关键词** 动态插值; 时态 GIS; ArcGIS VBA

**中图分类号** TP311

**文献标识码** A

## The Automation of Dynamic Interpolated Frame based on ArcGIS VBA

Cheng Zhen Huang Cheng Chen Changhong

(Shanghai Academy of Environmental Sciences 200233 China)

**Abstract** Realizing the automation of dynamic interpolated frame by using ArcGIS VBA, which enhances the visualization of temporal GIS for the discrete data's interpolation. The program, as a foundation of animation file creation, which can capture the interpolated frame time by time by itself, is a helpful supplement for the ArcGIS 9.2's temporal GIS module.

**Keywords** dynamic interpolation; temporal GIS; ArcGIS VBA

### 1 引言

随着GIS技术的快速发展, 获取和存储空间数据的能力越来越强, 人们已不满足静态的地理数据分析, 而要求GIS能够表达和分析时间上的空间变化, 时态地理信息系统的研究也成为近年来GIS领域的一个研究热点<sup>[1]</sup>。美国ESRI公司, 作为GIS软件领域的引领者, 在其开发的ArcGIS系列产品9.2版本中引入了对时态GIS的支持, 从数据模型的建立、存储到数据分析与表达, 都实现了用户对时态GIS的功能需求。然而, 目前ArcGIS9.2时态GIS模块采用的都是空间地物和属性数据相对应的方法, 通过地物编号的识别, 将不同时刻的属性数据链接到空间地物, 提供的表达结果也只能基于这些具有确定属性数据的地物<sup>[2]</sup>, 而不能先将有限数据空间插值之后再动画显示。

在实际情况中, 我们往往碰到这样的问题: 在某一空间范围内, 通过实地监测或模型获得了有限几个点的不同时刻的属性数据, 然后采用某种插值方法估算整个空间范围内的分布情况。此时, 如果想查看某个时刻的插值结果, 我们可以将该时刻的属性数据更新绑定到地物, 然后选用某种插值方法进行空间插值, 可将得到的结果图导出并保存。但是, 如果时间

---

作者简介: 程真(1983-)男, 江西上饶人, 工程师, 研究方向: 环境信息化与决策支持, Email: [chengz@saes.sh.cn](mailto:chengz@saes.sh.cn).

的长度很长,时刻取样的频率很高,那么上述方法将使工作量成倍上升,有时甚至超出人力所及。

针对上述涉及到空间插值的动画效果的表达, ArcGIS 9.2的时态GIS模块并没有很好的解决办法,不少研究人员在其它GIS软件平台上也实现过类似的功能表达<sup>[3-4]</sup>。本文尝试通过 Desktop VBA的编程<sup>[5]</sup>,实现了不同时刻动态插值帧的自动生成,为实现动画效果表达奠定基础,可作为ArcGIS 9.2时态GIS模块的一个有益的补充。

## 2 技术原理与流程

### 2.1 数据准备

需要准备的原始数据包括矢量格式的、点、线或面状地理图层,以及存储不同时刻属性值的属性数据库。在本研究中,我们以应用最广泛的点状 SHP 图层文件和 Access 数据库为例。表 1 列出了点状 SHP 文件必须包括的字段信息,其中 ID 字段唯一标识地物。

表 1 点状 SHP 文件的字段列表

编号	字段名	字段类型	字段说明
1	OBJECTID	Objcet ID	创建点对象时的自动编号。
2	SHAPE	Point	存储点的空间信息。
3	ID	TEXT	与属性数据连接的唯一标识地物。
4	tempValue	Double	临时数据存储字段。

表 2 为 Access 数据库属性数据表的字段列表,其中 FeatureID 和表 1 中的 ID 字段类型相同,用于连接时的绑定字段,其余字段都是以各时刻命名,存储各个时刻的属性值。值得注意的是,大部分属性数据库的原始格式可能并不是和表 2 结构相同,时间信息和属性值信息分别存储在独立的两个字段,但通过查询 SQL 语句转换为表 2 的格式并不困难。

表 2 属性数据表的字段列表

编号	字段名	字段类型	字段说明
1	FeatureID	TEXT	标识地物的唯一编号。
2	[时刻 1]	Double	时刻 1 的属性值。
3	[时刻 2]	Double	时刻 2 的属性值。
...	...	Double	某个时刻的属性值。
N	[时刻 N]	Double	时刻 N 的属性值。

在空间数据和属性数据都准备完毕后,还有一个必须的步骤是在 Arcmap 软件中将空间数据连接到属性数据表,连接对应字段分别是表 1 的 ID 字段和表 2 的 FeatureID 字段。

### 2.2 实现流程

#### 2.2.1 界面设计

为了增加程序的通用性,让其它项目中也能方便修改、使用,在 Arcmap 工具栏中新建一个功能按钮,命名“动态插值导出图片”,运行界面的设计窗口如图 1 所示。从图 1 中看到,运行的界面包括“开始时间”、“结束时间”、“分辨率”、“表头”四个文本输入框以及一个“插值并导出图片”按钮。前两个输入框用于用户设置需要提取数据的开始时间和结束时间,“分辨率”文本框用于设置需要导出的图片的分辨率大小,“表头”文本框用于设置属性数据表的表名,因为当空间数据和属性数据相连接时,表名也会显示在字段名称中。

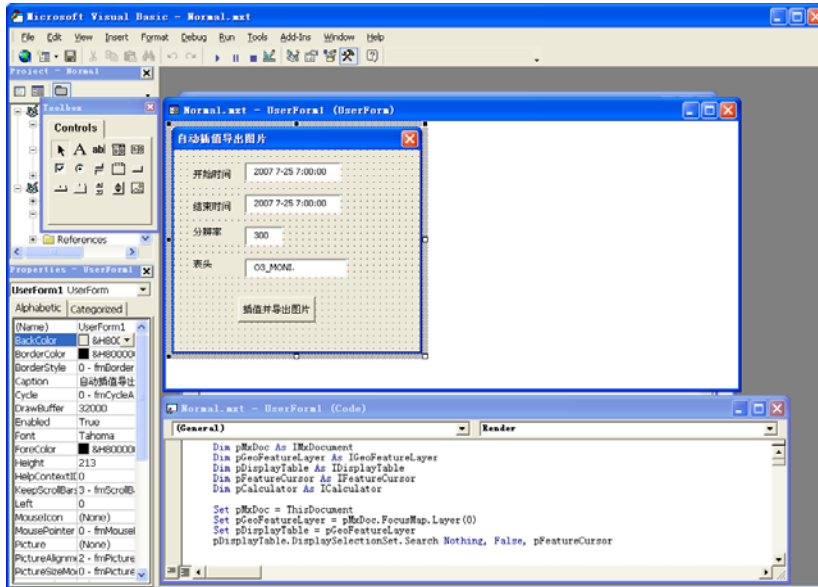


图 1 运行界面设计窗口

## 2.2.2 算法实现

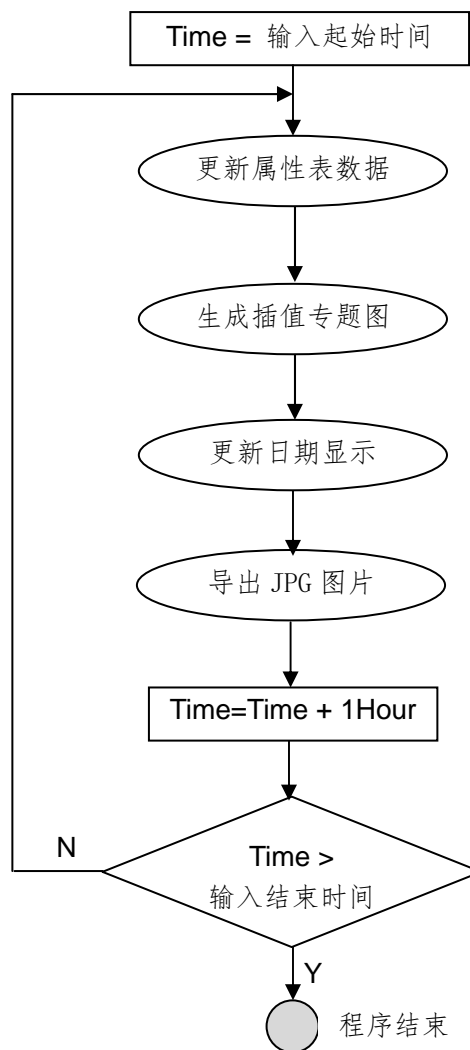


图 2 程序算法流程图

图 2 为整个程序算法流程图，本质上是一个循环生成图片的过程。每个时刻，首先更新空间图层的属性数据表，将该时刻的属性数据库的值更新到空间地理图层的 **tempValue** 字段；然后根据 **tempValue** 字段值，选择合适的插值方法及相关参数，对当前空间图层进行插值；同时更新当前日期显示标签的值；最后将当前的插值结果按设定的相关参数以图片的形式导出并保存。单个过程结束后，时间加 1，判断时间是否超过结束时间，如没有则又进入一个新的图片生成过程。

以下是四个主要过程的详细实现方法。

### (1) 更新属性数据

该子过程的主要代码及说明：

```
Sub UPDATE_FIELD (ByVal ValueField As String) ' ValueField 为属性数据库存储某时刻数值的字段名称
    Dim pMxDoc As IMxDocument
    Dim pGeoFeatureLayer As IGeoFeatureLayer
    Dim pDisplayTable As IDisplayTable
    Dim pFeatureCursor As IFeatureCursor
    Dim pCalculator As ICalculator
    Set pMxDoc = ThisDocument
    Set pGeoFeatureLayer = pMxDoc.FocusMap.Layer(0) ' 假定空间数据图层位于第一层
    Set pDisplayTable = pGeoFeatureLayer
    pDisplayTable.DisplaySelectionSet.Search Nothing, False, pFeatureCursor
    Set pCalculator = New Calculator
    With pCalculator ' 字段计算更新器
        Set .Cursor = pFeatureCursor
        .Expression = "[" + ValueField + "]" ' 直接用属性数据库的字段更新替换
        .Field = "Point_Feature.tempValue" ' Point_Feature 为空间图层名称
        .Calculate
    End With
End Sub
```

### (2) 生成插值专题图<sup>[6]</sup>

该子过程代码较复杂，在此列出关键代码及说明：

```
Sub Render (ByVal layername As String) ' layername 为插值所得新图层的名称
    Dim pInterpolationOp As IInterpolationOp
    Set pInterpolationOp = New RasterInterpolationOp
    Dim penv As IRasterAnalysisEnvironment
    Set penv = pInterpolationOp
    penv.SetCellSize esriRasterEnvValue, 400 ' 设定栅格图层的精度
    penv.SetExtent esriRasterEnvValue, pREnv
    ' 假定采用的插值方法为克里格插值
    Set pOutputDataset = pInterpolationOp.Krige(pFDescr, esriGeoAnalysisCircularSemiVarioqram,
    pRadius, True)
    Set pOutRaster = pOutputDataset
    pOutRasLayer.CreateFromRaster pOutRaster
    pOutRasLayer.Name = layername
    Set pClassRen = New RasterClassifyColorRampRenderer
```

```

Set pRasRen = pClassRen
Set pRasRen.Raster = pOutRaster      ' 设置栅格的专题渲染
pClassRen.ClassCount = 6
pRasRen.Update
Dim mycolors(6) As IColor             ' 依次设定插值各等级的颜色
Dim pFSymbol As IFillSymbol          ' 创建等级显示符号
Set pFSymbol = New SimpleFillSymbol
For i = 0 To pClassRen.ClassCount - 1 ' 设定各等级的颜色、标签及阈值等
    pFSymbol.Color = mycolors(i)
    pFSymbol.Color.Transparency = 50
    pClassRen.Symbol(i) = pFSymbol
    pClassRen.Break(i) = 10 + 10 * i
    pClassRen.Label(i) = "Class" & CStr(i)
Next
pRasRen.Update                       ' 更新专题图到图层
Set pOutRasLayer.Renderer = pClassRen
.....                                ' 添加生成的栅格图层到当前地图
End Sub

```

### (3) 更新日期显示标签的值

```

Public Sub UpdateTEXT (ByVal strDT As String)      ' strDT 为要显示的文本
    Set pGraphicsCont = pActiveView.GraphicsContainer
    pGraphicsCont.Reset
    Set pElement = pGraphicsCont.Next
    Do Until pElement Is Nothing                  ' 遍历 layout 中的 element
        Set pElementProp = pElement
        If pElementProp.Name = "DT" Then        ' 查找名字为 DT 的 element
            If TypeOf pElement Is ITextElement Then
                Set pTextElement = pElement
                pTextElement.Text = strDT
            End If
        End If
        Set pElement = pGraphicsCont.Next
    Loop
    pActiveView.PartialRefresh esriViewGraphics, Nothing, Nothing ' 更新显示
End Sub

```

### (4) 导出 JPG 图片

```

Sub Output_JPG (ByVal fileName As String)
    iScreenResolution = 96      ' 默认分辨率为 96dpi
    iOutputResolution = Val(tbReso.Text) ' 输出分辨率由用户设定
    pExport.Resolution = iOutputResolution
    With exportRECT
        .Left = 0
    End With

```

```

.Top = 0
.Right = pActiveView.ExportFrame.Right * (iOutputResolution / iScreenResolution)
.Bottom = pActiveView.ExportFrame.bottom * (iOutputResolution / iScreenResolution)
End With
Set pPixelBoundsEnv = New Envelope
pPixelBoundsEnv.PutCoords exportRECT.Left, exportRECT.Top, exportRECT.Right,
exportRECT.bottom
pExport.PixelBounds = pPixelBoundsEnv
hDC = pExport.StartExporting
pActiveView.Output hDC, pExport.Resolution, exportRECT, Nothing, Nothing
pExport.FinishExporting
pExport.Cleanup
pMxDoc.FocusMap.DeleteLayer pMxDoc.FocusMap.Layer(8) ' 将新生成的图层删除
End Sub

```

### 3 应用举例

将上述程序应用于上海市臭氧逐小时演变动画的制作，极大地提高了生成插值图片的效率，取得了良好效果。原始数据来自空气质量模型模拟得到的上海几个离散点的 2007 年 7 月臭氧的逐小时浓度，希望通过空间插值得到 7 月份全市范围的臭氧的污染分布演变趋势。图 3 为程序运行时的输入参数，我们选取臭氧高污染的 7 月 24 日 9 时到 7 月 29 日 9 时作为研究时段，分辨率设定 300dpi，属性数据表的表名为 O3\_MONI，点击“插值并导出图片”按钮后，程序将这一时段内逐小时的臭氧浓度插值图片导出并保存到硬盘。

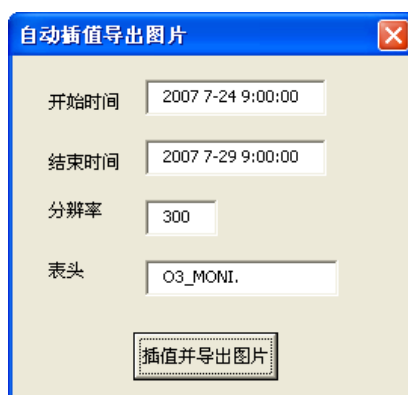


图 3 程序运行输入参数

图 4 显示了程序运行得到的动态帧的部分结果。通过空间插值，可以看到 7 月 24 日上海东北部的臭氧污染有一个明显的从上午开始到下午 14 时逐步加重和扩散的演变过程。在得到这些比例、背景、渲染方法都相同的单帧之后，借助一些常用的动画生成软件，如 GIF Movie Gear 等，可以很容易地生成如 GIF、AVI 等格式的动画文件。

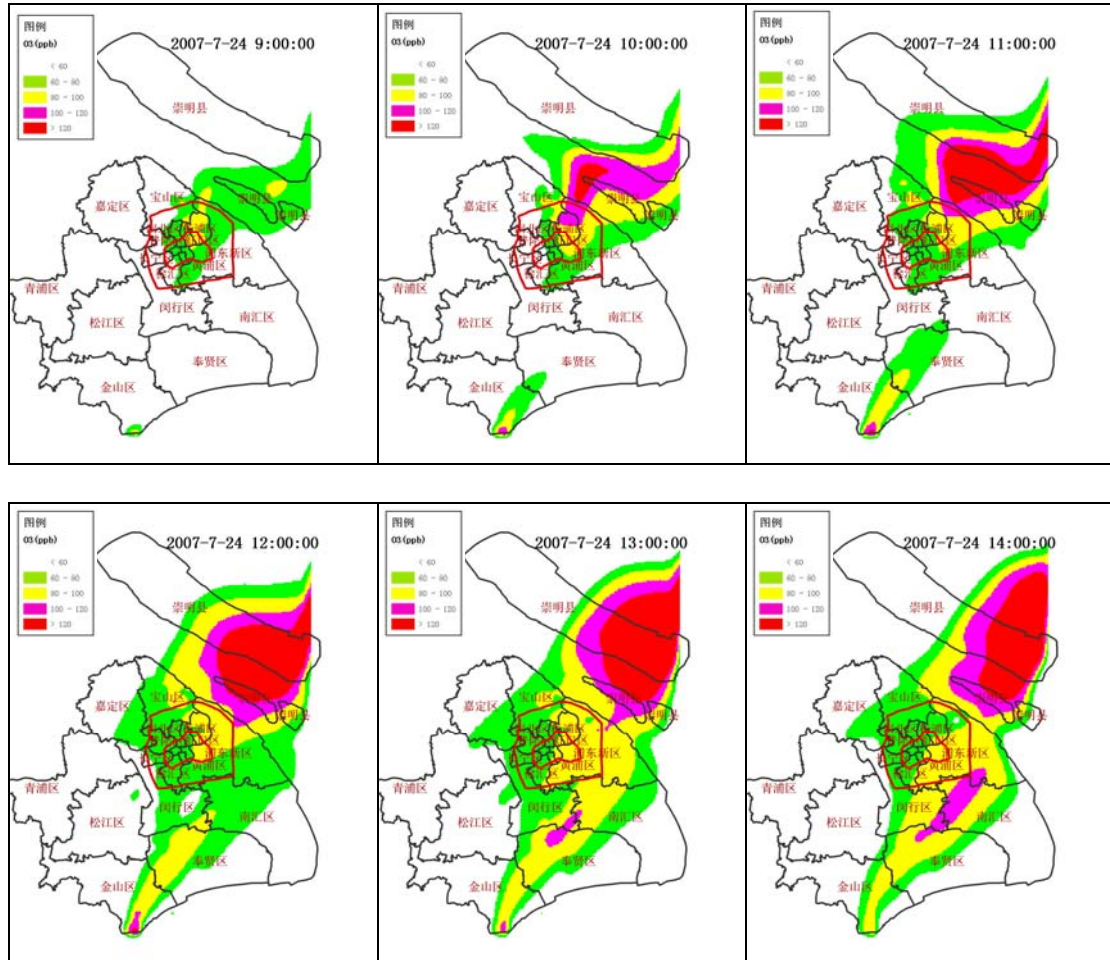


图4 程序自动生成的部分动态帧

## 4 结论与建议

- (1) 通过 ArcGIS VBA 程序的开发, 实现了离散点的时间序列属性数据的动态插值帧的自动生成, 对目前 ArcGIS 产品中的时态 GIS 的表达作了有益的补充。
- (2) 本文仅仅是实现了一个简单的案例。在实际运用中, 数据存储格式、插值方法的选择和插值参数设定、时间步长的动态设置、显示效果和范围的参数设置都不尽相同, 在后续的补充完善开发中, 只有把这些问题都解决, 才能增强程序的功能性和通用性。
- (3) 本文得到的结果是各个时刻的动态插值帧, 仍然还需要借助外在的动画生成软件实现动画文件的生成。在后续的工作中, 可以进一步借助第三方组件的嵌入使用, 实现程序内即直接生成动画文件。

## 参考文献

- [1] 陈荣清, 谢刚生, 邹时林. 时态GIS中时空数据表达综述[J]. 华东地质学院学报, 2001, 24(4): 320-322.
- [2] ArcGIS Developer 联机帮助[EB/OL]. 2008.
- [3] 张圣微, 雷玉平, 郑力等. 用空间数据库及COM技术实现GIS空间插值运算[J]. 哈尔滨工程大学学报, 2006, 27(6): 858-861, 877.
- [4] 喻永平, 陈晓勇, 都洁. 基于数学模糊逻辑的土地利用动态可视化[J]. 城市勘测, 2007, (5): 39-41.
- [5] 党安荣, 贾海峰, 易善帧等. ArcGIS 8 Desktop地理信息系统应用指南[M]. 北京: 清华大学出版社, 2003. 582-647.

[6] 朱求安,张万昌,余钧辉. 基于GIS的空间插值方法研究[J]. 江西师范大学学报(自然科学版),2004,28(2):183-188.