



shp文件格式 (转载) 05.08.31 from joey 来源 以文找文 NEW

[上一篇](#) [下一篇](#)

Tag:gis 文件格式

这篇文章不错, [我也要收藏](#)

发信人: liuwenjin (bull), 信区: GIS

标 题: shp文件格式

发信站: 两全其美 BBS (Mon Aug 29 23:36:04 2005), 本站(lqqm.net)

SHAPE 文件格式

SDE,ARC/INFO,PC ARC/INFO,Data Automation Kit(DAK)和ArcCAD软件提供了shape 到coverage的数据转换器, ARC/INFO同样提供了coverage到shape的转换器。为了和其他数据格式交换, shape文件的格式在本报告中被出版。其他数据流, 比如来自全球定位系统(GPS)接收机的数据能同样被存为shape文件或X,Y事件表。

Shape文件技术描述计算机程序能通过使用本节的技术描述来产生, 读, 写shape文件。

一个ESRI的shape文件包括一个主文件, 一个索引文件, 和一个dBASE表。主文件是一个直接存取, 变量记录长度文件, 其中每个记录描述一个有它自己的vertices列表的shape。在索引文件中, 每个记录包含对应主文件记录离主文件头开始的偏移, dBASE表包含一feature一个记录的feature的特征。几何和属性间的一一对应关系是基于记录数目的。在dBASE文件中的属性记录必须和主文件中的记录是相同顺序的。

命名习惯所有文件名都符合8.3命名习惯。主文件, 索引文件和dBASE文件有相同的前缀。前缀必须是由字符或数字(a-Z,0-9)开始, 后跟0到7个字符(a-Z,0-9, _)主文件的后缀是.shp, 索引文件的后缀是.shx, dBASE表的后缀是.dbf。文件名中的所有字母在对文件名敏感的操作系统中都是小写的。

例子

主文件: counties.shp 索引文件:counties.shx dBASE表: ounties.dbf数字类型一个shape文件存储整数和双精度数, 本文档的余数指以下类型:

整数: 有符号32位整数(4字节)

双精度: 有符号64位IEEE双精度浮点数(8字节)浮点数必须是数字的值。负无穷, 正无穷和非数字(NaN)值在shape文件不被允许。然而shape文件支持'没有数据'的值这样的概念, 但是目前只用于衡量。某些小于-1038被shape文件读取程序用来代表'没有数据'的值。

下面的第一节描述shape文件的总体结构和组织。第二节描述shape文件支持的每种shape类型的记录内容。

主文件的组织

主文件(.shp)由固定长度的文件头和接着的变长度记录组成。每个变长度记录是由固定长度的记录头和接着的变长度记录内容组成。图1图解了主文件的结构。

图 1 主文件的结构

文件头

记录头 记录内容

记录头 记录内容

记录头 记录内容

记录头 记录内容

.....

.....

记录头 记录内容

Shape文件中所有的内容可以被分为二类:

与数据相关的:

。主文件记录内容

- 。主文件头的数据描述域（Shape 类型，边界盒等）
- 与文件管理相关的：
 - 。文件和记录长度
 - 。记录偏移等

整数和双精度整数在文件头中组成数据描述域，在主文件的记录内容是小ndian（PC或Intel）字节顺序。组成文件的其余部分和文件管理的整数和双精度浮点数是大endian（Sun或Motorola）字节顺序。

主文件头

主文件头100字节长。表1显示带有字节位置，值，类型和字节顺序的文件头中的域。
在此表中，位置是相对于文件的开始。

表 1 主文件头的描述

位置	域	值	类型	字节顺序
0字节	文件代码	9994	整数	大
4字节	未被使用	0	整数	大
8字节	未被使用	0	整数	大
12字节	未被使用	0	整数	大
16字节	未被使用	0	整数	大
20字节	未被使用	0	整数	大
24字节	文件长度	文件长度	整数	大
28字节	版本	1000	整数	小
32字节	Shape类型	Shape类型	整数	小
36字节	边界盒	Xmin	双精度	小
44字节	边界盒	Ymin	双精度	小
52字节	边界盒	Xmax	双精度	小
60字节	边界盒	Ymax	双精度	小
68字节*	边界盒	Zmin	双精度	小
76字节*	边界盒	Zmax	双精度	小
84字节*	边界盒	Mmin	双精度	小
92字节*	边界盒	Mmax	双精度	小

*未被使用，值为0.0，若没有被衡量或是Z轴。

文件长度的值是在16位字下文件的总长度（包括组成文件头的50个16位字）。在shape文件中的所有非空

shape被需要是同种shape类型。Shape类型的值如下：

值	shape类型
0 空shape	
1 点	
3 多线	
5 多边形	
8 多点	
11 点Z	
13 多线Z	
15 多边形Z	
18 多点Z	
21 点M	
23 多线M	
25 多边形M	
28 多点M	
31 多斑块	

没有被定义的Shape 类型值（2， 4， 6等直到33）为将来可能的使用而保留。目前shape文件被局限于包含

以上定义的同种shape类型。在将来shape文件可以被允许包含多于一种shape类型。若混合shape类型被实现，文件头中的shape类型将标识该文件。

主文件头的边界盒存储文件中shape的实际幅度。最小边界X和Y直交的（潜在的M,Z）长方形包含了所有的shape。若shape文件是空的（没有记录），min,Ymin,Xmax,Ymax的值是未被定义的。Mmin和Mmax能包含shape文件用来衡量不包含衡量的shape类型的'没有数据'的值（参见2页的数字类型），记录头每个记录的头存储了记录的数目和记录内容的长度。记录头有一个固定长度8字节。表 2显示文件记录头中域的字节位置，值，类型和字节顺序。在表中，位置是相对于记录的开始的。

表 2 主文件记录头文件的描述

位置	域	值	类型	字节顺序
0字节	记录数目	记录数目	整数	大
4字节	内容长度	内容长度	整数	大

记录数目从1开始。

一个记录的内容长度是按16位字衡量的记录内容长度。每个记录因此为文件的总长度贡献（4+内容长度）个16位字，正如文件头是24字节一样。

主文件记录内容

Shape文件记录内容包含一个shape类型和接着的该shape的几何数据。记录内容的长度依赖于在一个shape中部分和vertices的数目。对每种shape类型，我们首先描述该shape然后是它在磁盘上的存储镜像。在表3到16，位置是相对于记录内容的开始。

空shape

shape类型为0指代一种不带几何数据的空shape，每种要素类型（点，线，多边形等）都支持空，在同一个shape文件中有点和空的点是有效的。常常空shape是放东西的地方；在shape文件产生时被使用和在生产后更被广泛地使用。

表 3 空shape记录内容

位置	域	值	类型	数目	字节顺序
0字节	shape类型	0	整数	1	小

在X,Y位置的Shape类型

点 一个点包括一对以X,Y顺序排列的双精度的坐标

```
Point
{
    Double  X //X坐标
    Double  Y //Y坐标
}
```

表 4 点记录内容

位置	域	值	类型	数目	字节顺序
0字节	shape类型	1	整数	1	小
4字节	X	X	双精度	1	小
12字节	Y	Y	双精度	1	小

多点 一个多点代表一个点的集合：

```
MultiPoint
{
    Double[4]    Box      //边界盒
    Integer      NumPoints //点的数目
    Point[NumPoints]  Points //在集合中的点
}
```

边界盒以Xmin,Ymin,Xmax,Ymax存储。

表 5 多点记录内容

位置	域	值	类型	数目	字节顺序
0字节	shape类型	8	整数	1	小
4字节	Box	Box	双精度	4	小
36字节	NumPoints	NumPoints	整数	1	小
40字节	Points	Points	点	NumPoints	小

PolyLine 一条PolyLine是指一条包含一个或多个部分的有序的**vertices**的集合。一个部分是指二个或多个点彼此连接的顺序。部分间彼此相连或不连。部分间彼此可能相交或不相交。

因为该定义没有禁止有确定坐标的连续点，**shape**文件的读程序必须掌握这样的情况。在另外，退化(**degenerate**)和可能导致零长度的结果是不被允许的。

```
PolyLine
{
    Double[4]      Box      //边界盒
    Integer        NumParts //部分的数目
    Integer        NumPoints //点的总数目
    Integer[NumParts] Parts  //在部分中第一个点的索引
    Point[NumPoints] Points  //所有部分的点
}
```

PolyLine的域在以下为更详细的描述：

Box 被存储的PolyLine的边界盒，以Xmin,Ymin,Xmax,Ymax的顺序存储。

NumParts 在PolyLine中部分的数目。

NumPoints 所有部分的点的总数目。

Parts NumParts长度的数列。为每条PolyLine存储它在点数列中的第一个点的索引。数列索引是从0开始的。

Points NumPoints长度的数列。在PolyLine中的每一部分的点被尾到尾存储。部分2的点跟在部分1的点之后，如此下去。部分数列对每一部分保持开始点的数列索引。

在部分间点之间没有界限。

表 6 PolyLine记录内容

位置	域	值	类型	数目	字节顺序
0字节	shape类型	3	整数	1	小
4字节	Box	Box	双精度	4	小
36字节	NumParts	NumParts	整数	1	小
40字节	NumPoints	NumPoints	整数	1	小
44字节	Parts	Parts	整数	NumParts	小
X字节	Points	Points	点	NumPoints	小

注意：X=44+4* NumParts 多边形 一个多边形包含一个或多个环。一个环是四或多个点彼此相连组成的一个闭合的彼此不相交的环。一个多边形可能包括多个外环，一个环的**vertices**的顺序和方向指示环的哪一边是多边形的内部。在多边形中的洞的环的**vertices**是逆时针方向的。一个环组成的多边形总是顺时针方向的。一个多边形的环是被做为它的一部分的。因为该定义没有禁止有确定坐标的连续点，**shape**文件读程序必须解决这种情况。在另外，退化(**degenerate**)和可能导致零长度的结果是不被允许的。多边形的结构被定义为PolyLine结构，正如下文：

```
Polygon
{
    Double[4]      Box      //边界盒
    Integer        NumParts //部分的数目
    Integer        NumPoints //点的总数目
    Integer[NumParts] Parts  //在部分中第一个点的索引
    Point[NumPoints] Points  //所有部分的点
}
```

}
多边形的域在以下为更详细的描述:
Box 被存储的多边形的边界盒, 以Xmin,Ymin,Xmax,Ymax的顺序存储。
NumParts 在 多边形中环的数目。
NumPoints 所有环的点的总数目。
Parts **NumParts**长度的数列。为每条环存储它在点数列中的第一个点的索引。数列索引是从0开始的。
Points **NumPoints**长度的数列。在多边形中的每一个环的点被尾到尾存储。环2的点跟在环1的点之后, 如此下去。部分数列对每一环保持开始点的数列索引。在环间点之间没有界限。
图 2 中的例子图演示了多边形的表现。该图中的多边形有一个洞, 总共是8个vertices。

关于多边形shape有以下重要注意事项:
环是闭合的(第一个和最后一个vertex必须是一样的)在点数列中环的顺序是不重要的。
存储在shape文件中的多边形必须是clean, 一个clean的多边形是指这样的一个多边形:
1. 没有自交。这意味着属于一个环的一段可能不与另一个环的一段相交。一个多边形的环可能在vertices处彼此相交, 但不是在沿段处。重合的段被认为是相交的。
2. 在定义多边形的线的正确一边有多边形的内部。一个观察者以vertex顺序沿环走时, 右边的邻居是多边形的内部。一个独环的多边形的vertices因此都是顺时针的。在这些多边形中的洞有一个逆时针的方向。当定义在多边形中的洞同样是顺时针时, 发生"Dirty"多边形错误。这会导致内部的重叠。
图 2 一个多边形的实例
在此例中, NumParts等于2,NumPoints等于10。注意洞多边形的点的顺序是逆序的。

```
0 5
部分: 0 5
0 1 2 3 4 5 6 7 8 9
点: v1 v2 v3 v4 v1 v5 v8 v7 v6 v5
```

表 7 多边形记录内容

位置	域	值	类型	数目	字节顺序
0字节	shape类型	5	整数	1	小
4字节	Box	Box	双精度	4	小
36字节	NumParts	NumParts	整数	1	小
40字节	NumPoints	NumPoints	整数	1	小
44字节	Parts	Parts	整数	NumParts	小
X字节	Points	Points	点	NumPoints	小

注意: X=44+4* NumParts

在X,Y空间中衡量的shape类型该种shape有一个额外的坐标 M, 注意"没有数据"的值可以被指定为M的一个值。

PointM 一个PointM包括一对以X,Y顺序排列的双精度的坐标, 再加上衡量M。

```
PointM
{
    Double X //X 坐标
    Double Y //Y 坐标
    Double M //M 衡量
}
```

表 8 PointM记录内容

位置	域	值	类型	数目	字节顺序
0字节	shape类型	21	整数	1	小
4字节	X	X	双精度	1	小
12字节	Y	Y	双精度	1	小

20字节 M M 双精度 1 小
多PointM 一个多PointM代表一个PointM的集合:

```
MultiPoint
{
    Double[4]      Box      //边界盒
    Integer          NumPoints //点的数目
    Point[NumPoints] Points    //在集合中的点
    Double[2]      M Range    //衡量M的上下界
    Double[NumPoints] M Array    //衡量
}
```

多PointM的域在以下为更详细的描述:

Box 被存储的多PointM的边界盒, 以Xmin,Ymin,Xmax,Ymax的顺序存储。

NumPoints 点的总数目。

Points NumPoints长的点的数列。

M Range 以Mmin,Mmax顺序存储的多PointM的最小, 最大值。

M Array NumPoints长的衡量的数组。

表 9 多PointM记录内容

位置	域	值	类型	数目	字节顺序
0字节	shape类型	28	整数	1	小
4字节	Box	Box	双精度	4	小
36字节	NumPoints	NumPoints	整数	1	小
40字节	Points	Points	点	NumPoints	小
X*字节	Mmin	Mmin	双精度	1	小
X+8*字节	Mmax	Mmax	双精度	1	小
X+16*字节	Marry	Marry	双精度	NumPoints	小

注意: X = 40+ (16* NumPoints) *可选的

多LineM 一条shape文件的多LineM包含一个或多个部分。指一条包含一个或多个部分的有序的vertices的集合。一个部分是指二个或多个点彼此连接的顺序。部分间彼此相连或不连。部分间彼此可能相交或不相交。

```
PolyLineM
{
    Double[4]      Box      //边界盒
    Integer          NumParts //部分的数目
    Integer          NumPoints //点的总数目
    Integer[NumParts] Parts    //在部分中第一个点的索引
    Point[NumPoints] Points    //所有部分的点
    Double[2]      M Range    //衡量M的上下界
    Double[NumPoints] M Array    //所有点的衡量
}
```

PolyLineM的域在以下为更详细的描述:

Box 被存储的PolyLineM的边界盒, 以Xmin,Ymin,Xmax,Ymax的顺序存储。

NumParts 在PolyLineM中部分的数目。

NumPoints 所有部分的点的总数目。

Parts NumParts长度的数列。为每条PolyLineV存储它在点数列中的第一个点的索引。数列索引是从0开始的。

Points NumPoints长度的数列。在PolyLineM中的每一部分的点被尾到尾存储。部分2的点跟在部分1的点之后, 如此下去。部分数列对每一部分保持开始点的数列索引。

在部分间点之间没有界限。

M Range 以Mmin,Mmax顺序存储的多PointM的最小, 最大值。

M Array NumPoints长的衡量的数组,在多PointM每部分的衡量被被尾到尾存储部分2的点跟在部分1的点之后, 如此下去。部分数列对每一部分保持开始点的数列索引。在部分间点之间没有界限。

表 10 PolyLineM记录内容

位置	域	值	类型	数目	字节顺序
0字节	shape类型	23	整数	1	小
4字节	Box	Box	双精度	4	小
36字节	NumParts	NumParts	整数	1	小
40字节	NumPoints	NumPoints	整数	1	小
44字节	Parts	Parts	整数	NumParts	小
X字节	Points	Points	点	NumPoints	小
Y*字节	Mmin	Mmin	双精度	1	小
Y+8*字节	Mmax	Mmax	双精度	1	小
Y+16*字节	Marry	Marry	双精度	NumPoints	小

注意: $X=44+4*NumParts$ $Y=40+(16*NumPoints)$ *可选的

多边形M 一个多边形M包含一个或多个环。一个环是四或多个点彼此相连组成的一个闭合的彼此不相交的环。注意相交是在X,Y空间计算, 而不是在X,Y,M空间。一个多边形可能包括多个外环, 一个多边形的环是被做为它的一部分的。

多边形M的结构被定义为PolyLineM结构, 正如下文:

```
PolygonM
{
    Double[4]      Box      //边界盒
    Integer        NumParts //部分的数目
    Integer        NumPoints //点的总数目
    Integer[NumParts] Parts  //在部分中第一个点的索引
    Point[NumPoints] Points  //所有部分的点
    Double[2]      M Range  //衡量M的上下界
    Double[NumPoints] M Array //所有点的衡量
}
```

多边形M的域在以下为更详细的描述:

Box 被存储的多边形的边界盒, 以Xmin,Ymin,Xmax,Ymax的顺序存储。

NumParts 在多边形M中环的数目。

NumPoints 所有环的点的总数目。

Parts NumParts长度的数列。为每条环存储它在点数列中的第一个点的索引。数列索引是从0开始的。

Points NumPoints长度的数列。在多边形中的每一个环的点被尾到尾存储。环2的点跟在环1的点之后, 如此下去。部分数列对每一环保持开始点的数列索引。在环间点之间没有界限。

M Range 以Mmin,Mmax顺序存储的多PointM的最小, 最大值。

M Array NumPoints长的衡量的数组,在多PointM每部分的衡量被被尾到尾存储部分2的点跟在部分1的点之后, 如此下去。部分数列对每一部分保持开始点的数列索引。在部分间点之间没有界限。

关于PolygonM shape有以下注意事项:

环是闭合的(第一个和最后一个vertex必须是一样的) 在点数列中环的顺序是不重要的。

表 11 多边形M的记录内容

位置	域	值	类型	数目	字节顺序
0字节	shape类型	5	整数	1	小
4字节	Box	Box	双精度	4	小

36字节	NumParts	NumParts	整数	1	小
40字节	NumPoints	NumPoints	整数	1	小
44字节	Parts	Parts	整数	NumParts	小
X字节	Points	Points	点	NumPoints	小
Y*字节	Mmin	Mmin	双精度	1	小
Y+8*字节	Mmax	Mmax	双精度	1	小
Y+16*字节	Marry	Marry	双精度	NumPoints	小

注意: X=44+4* NumParts Y = 40+（16* NumPoints） *可选的在X,Y,Z空间中衡量的shape类型该种shape有一个额外的坐标 M, 注意"没有数据"的值可以被指定为M的一个值。

PointZ 一个PointZ包括一对以X,Y,Z顺序排列的双精度的坐标, 再加上衡量M。

```
PointM
{
    Double X //X 坐标
    Double Y //Y 坐标
    Double Z //Z 坐标
    Double M //M 衡量
}
```

表 12 PointZ记录内容

位置	域	值	类型	数目	字节顺序
0字节	shape类型	21	整数	1	小
4字节	X	X	双精度	1	小
12字节	Y	Y	双精度	1	小
20字节	Z	Z	双精度	1	小
28字节	Measure	M	双精度	1	小

多PointZ 一个多PointZ代表一个PointZ的集合:

```
MultiPoint
{
    Double[4] Box //边界盒
    Integer NumPoints //点的数目
    Point[NumPoints] Points //在集合中的点
    Double[2] Z Range //Z的上下界
    Double[NumPoints] Z Array //Z的值
    Double[2] M Range //衡量M的上下界
    Double[NumPoints] M Array //衡量
}
```

边界盒是以Xmin,Ymin,Xmax,Ymax的顺序存储。Z Range是以Mmin,Mmax顺序存储的, M Range是以Mmin,Mmax顺序存储的。

表 9 多PointM记录内容

位置	域	值	类型	数目	字节顺序
0字节	shape类型	28	整数	1	小
4字节	Box	Box	双精度	4	小
36字节	NumPoints	NumPoints	整数	1	小
40字节	Points	Points	点	NumPoints	小
X字节	Zmin	Zmin	双精度	1	小
X+8字节	Zmax	Zmax	双精度	1	小
X+16字节	Zarry	Zarry	双精度	NumPoints	小
Y*字节	Mmin	Mmin	双精度	1	小

Y+8*字节 Mmax Mmax 双精度 1 小
Y+16*字节 Marry Marry 双精度 NumPoints 小

注意: $X = 40 + (16 * \text{NumPoints})$ $Y = X + 16 + (8 * \text{NumPoints})$ *可选的

多LineZ 一条shape文件的多LineZ包含一个或多个部分。指一条包含一个或多个部分的有序的vertices的集合。一个部分是指二个或多个点彼此连接的顺序。部分间彼此相连或不连。部分间彼此可能相交或不相交。

```
PolyLineZ
{
    Double[4]      Box      //边界盒
    Integer        NumParts //部分的数目
    Integer        NumPoints //点的总数目
    Integer[NumParts] Parts //在部分中第一个点的索引
    Point[NumPoints] Points //所有部分的点
    Double[2]      Z Range  //Z的上下界
    Double[NumPoints] Z Array //Z的值
    Double[2]      M Range  //衡量M的上下界
    Double[NumPoints] M Array //所有点的衡量
}
```

PolyLineZ的域在以下为更详细的描述:

Box 被存储的PolyLineZ的边界盒, 以Xmin,Ymin,Xmax,Ymax的顺序存储。

NumParts 在PolyLineZ中部分的数目。

NumPoints 所有部分的点的总数目。

Parts NumParts长度的数列。为每条PolyLineZ存储它在点数列中的第一个点的索引。数列索引是从0开始的。

Points NumPoints长度的数列。在PolyLineZ中的每一部分的点被尾到尾存储。部分2的点跟在部分1的点之后, 如此下去。部分数列对每一部分保持开始点的数列索引。在部分间点之间没有界限。

ZRange 以Zmin,Zmax顺序存储的多PointZ的最小, 最大值。

Z Array NumPoints长的衡量的数组,在多PointZ每部分的衡量被被尾到尾存储部分2的点跟在部分1的点之后, 如此下去。部分数列对每一部分保持开始点的数列索引。在部分间点之间没有界限。

M Range 以Mmin,Mmax顺序存储的多PointZ的最小, 最大值。

M Array NumPoints长的衡量的数组,在多PointZ每部分的衡量被被尾到尾存储部分2的点跟在部分1的点之后, 如此下去。部分数列对每一部分保持开始点的数列索引。在部分间点之间没有界限。

表 14 PolyLineZ记录内容

位置	域	值	类型	数目	字节顺序
0字节	shape类型	23	整数	1	小
4字节	Box	Box	双精度	4	小
36字节	NumParts	NumParts	整数	1	小
40字节	NumPoints	NumPoints	整数	1	小
44字节	Parts	Parts	整数	NumParts	小
X字节	Points	Points	点	NumPoints	小
Y字节	Zmin	Zmin	双精度	1	小
Y+8字节	Zmax	Zmax	双精度	1	小
Y+16字节	Zarry	Zarry	双精度	NumPoints	小
Z*字节	Mmin	Mmin	双精度	1	小
Z+8*字节	Mmax	Mmax	双精度	1	小
Z+16*字节	Marry	Marry	双精度	NumPoints	小

注意: $X = 44 + 4 * \text{NumParts}$ $Y = X + (16 * \text{NumPoints})$ $Z = Y + 16 + (8 * \text{NumPoints})$ *可选的多边形Z 一个多边形Z包含一个或多个环。一个环是四或多个点彼此相连组成的一个闭合的彼此不相交的环。一个多边形可

能包括多个外环，一个多边形Z的环是被做为它的一部分的。
多边形Z的结构被定义为PolyLineZ结构，正如下文：

```
PolygonZ
{
    Double[4]      Box      //边界盒
    Integer        NumParts //部分的数目
    Integer        NumPoints //点的总数目
    Integer[NumParts] Parts  //在部分中第一个点的索引
    Point[NumPoints] Points  //所有部分的点
    Double[2]      Z Range  //Z的上下界
    Double[NumPoints] Z Array //Z的值
    Double[2]      M Range  //衡量M的上下界
    Double[NumPoints] M Array //所有点的衡量
}
```

多边形Z的域在以下为更详细的描述：

Box 被存储的多边形的边界盒，以Xmin,Ymin,Xmax,Ymax的顺序存储。

NumParts 在多边形Z中环的数目。

NumPoints 所有环的点的总数目。

Parts NumParts长度的数列。为每条环存储它在点数列中的第一个点的索引。数列索引是从0开始的。

Points NumPoints长度的数列。在多边形中的每一个环的点被尾到尾存储。环2的点跟在环1的点之后，如此下去。部分数列对每一环保持开始点的数列索引。在环间点之间没有界限。

ZRange 以Zmin,Zmax顺序存储的多PointZ的最小，最大值。

Z Array NumPoints长的衡量的数组,在多PointZ每部分的衡量被被尾到尾存储部分2的点跟在部分1的点之后，如此下去。部分数列对每一部分保持开始点的数列索引。在部分间点之间没有界限。

M Range 以Mmin,Mmax顺序存储的多PointZ的最小，最大值。

M Array NumPoints长的衡量的数组,在多PointZ每部分的衡量被被尾到尾存储部分2的点跟在部分1的点之后，如此下去。部分数列对每一部分保持开始点的数列索引。在部分间点之间没有界限。

关于PolygonZ shape有以下注意事项：

环是闭合的（第一个和最后一个vertex必须是一样的）在点数列中环的顺序是不重要的。

表 15 多边形Z的记录内容

位置	域	值	类型	数目	字节顺序
0字节	shape类型	15	整数	1	小
4字节	Box	Box	双精度	4	小
36字节	NumParts	NumParts	整数	1	小
40字节	NumPoints	NumPoints	整数	1	小
44字节	Parts	Parts	整数	NumParts	小
X字节	Points	Points	点	NumPoints	小
Y字节	Zmin	Zmin	双精度	1	小
Y+8字节	Zmax	Zmax	双精度	1	小
Y+16字节	Zarry	Zarry	双精度	NumPoints	小
Z*字节	Mmin	Mmin	双精度	1	小
Z+8*字节	Mmax	Mmax	双精度	1	小
Z+16*字节	Marry	Marry	双精度	NumPoints	小

注意：X=44+4* NumParts Y = X+（16* NumPoints）Z=Y+16+(8* NumPoints) *可选的多Patch 一个MultiPatch 包括许多表面斑块。每个表面斑块描述了一个表面。MultiPatch的表面斑块是指比如它的部分，控制一个MultiPatch部分的vertices顺序是如何被解释的部分类型。一个MultiPatch的部分可以是以下类型：

三角形条带 三角形的连接条带, 此处每个顶点 (在开始二个后) 完成一个新三角形。一个新三角形总是通过把新顶点和二个临近的原有点相连而得到。

三角形扇 三角形的连接扇, 此处每个顶点 (在开始二个后) 完成一个新三角形。一个新三角形总是通过把新顶点和一个个临近的原有点及这部分的第一个顶点相连而得到。

外环 多边形的外环。

内环 多边形的洞。

第一环 未定义类型的多边形的第一个环。

环 未定义类型的多边形的环。

一个简单的三角形条带或三角形扇代表一个简单的表面斑块。参见图 3 看这些部分类型的例子。带环的部分的序列能描述有洞的多边形表面。该序列典型包括一个外环, 代表斑块的外边界, 用许多内环代表洞。当用代表表现带洞的多边形斑块环的集合的个体环是不知道, 整个序列以第一环开始, 接着是其他的环。不跟着第一环后的环的序列被认为是没有洞的外环的序列。

图 3 MultiPatch 部分例子

用来对部分类型进行编码的值有:

值 部分类型

0 三角形条带

1 三角形扇

2 外环

3 内环

4 第一环

5 环

MultiPatch

```
{
    Double[4]      Box      //边界盒
    Integer        NumParts //部分的数目
    Integer        NumPoints //点的总数目
    Integer[NumParts] Parts //在部分中第一个点的索引
    Integer[NumParts] PartsTypes //部分类型
    Point[NumPoints] Points //所有部分的点
    Double[2]      Z Range //Z的上下界
    Double[NumPoints] Z Array //Z的值
    Double[2]      M Range //衡量M的上下界
    Double[NumPoints] M Array //所有点的衡量
}
```

多边形Z的域在以下为更详细的描述:

Box 被存储的多边形的边界盒, 以Xmin,Ymin,Xmax,Ymax的顺序存储。

NumParts 在多边形Z中环的数目。

NumPoints 所有环的点的总数目。

Parts NumParts长度的数列。为每条环存储它在点数列中的第一个点的索引。数列索引是从0开始的。

PartsType NumParts长度的数列, 存储每一部分的类型。

Points NumPoints长度的数列。在多边形中的每一个环的点被尾到尾存储。环2的点跟在环1的点之后, 如此下去。部分数列对每一环保持开始点的数列索引。在环间点之间没有界限。

ZRange 以Zmin,Zmax顺序存储的多PointZ的最小, 最大值。

Z Array NumPoints长的衡量的数组, 在多PointZ每部分的衡量被被尾到尾存储部分2的点跟在部分1的点之后, 如此下去。部分数列对每一部分保持开始点的数列索引。在部分间点之间没有界限。

M Range 以Mmin,Mmax顺序存储的多PointZ的最小, 最大值。

M Array NumPoints长的衡量的数组,在多PointZ每部分的衡量被被尾到尾存储部分2的点跟在部分1的点之后, 如此下去。部分数列对每一部分保持开始点的数列索引。在部分间点之间没有界限。

--

踏踏实实做事

VRMap(三维地理信息系统)

#attach imageAbelllandscape.jpg

※ 来源:. 两全其美 BBS lqqm.net. [FROM: 61.135.21.130]

[Need a Digitizing tablet?](#)

Buy a new or used digitizer for use with ESRI
ArcInfo, ArcView & ArcGIS

www.logicgroup.com

[A world-class MSc in GIS](#)

Study with UCL's renowned faculty and get
the career you want in GIS

www.gisatucl.info

[Mobile GIS Toolkit](#)

SpatialAce LE - GIS for handheld dev
networked applications

www.spatialace.com

Gooooooooogle 提供的广告

在本网

[上一篇](#) [下一篇](#) (joey的分类目录[gis]中共2篇)

以文找文 **NEW**: [点击这里](#)可搜索互联网上的相关文章

发表评论

游客请[登录](#)后再发表评论!