“十二五”普通高等教育本科国家级规划教材

地理信息系统教程

Dili Xinxi Xitong Jiaocheng

（第二版）

主编：汤国安

编著：汤国安 刘学军 闾国年
盛业华 王　春 张海平

高等教育出版社

第四章：空间数据结构

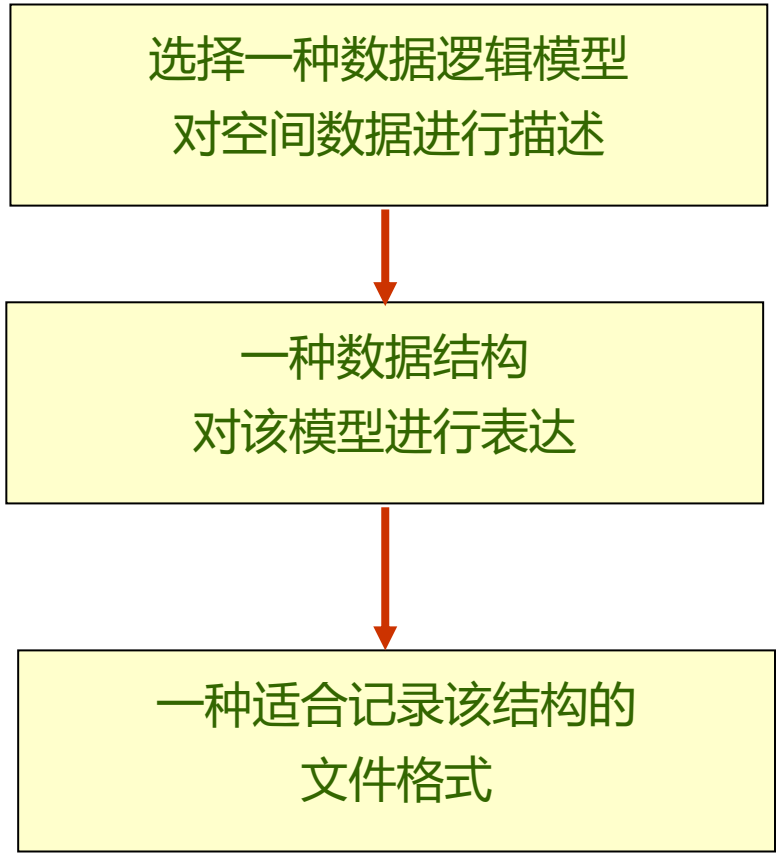
空间数据结构

空间数据结构(spatial data structure)是指对空间数据逻辑模型描述的数据组织关系和编排方式的具体实现，对地理信息系统中数据存储、查询检索和应用分析等操作处理的效率有着至关重要的影响。

空间数据结构是地理信息系统沟通信息的桥梁，只有充分理解地理信息系统所采用的特定数据结构，才能正确有效地使用系统。

空间数据结构

空间数据结构是数据逻辑模型与数据文件格式间的桥梁



空间数据结构

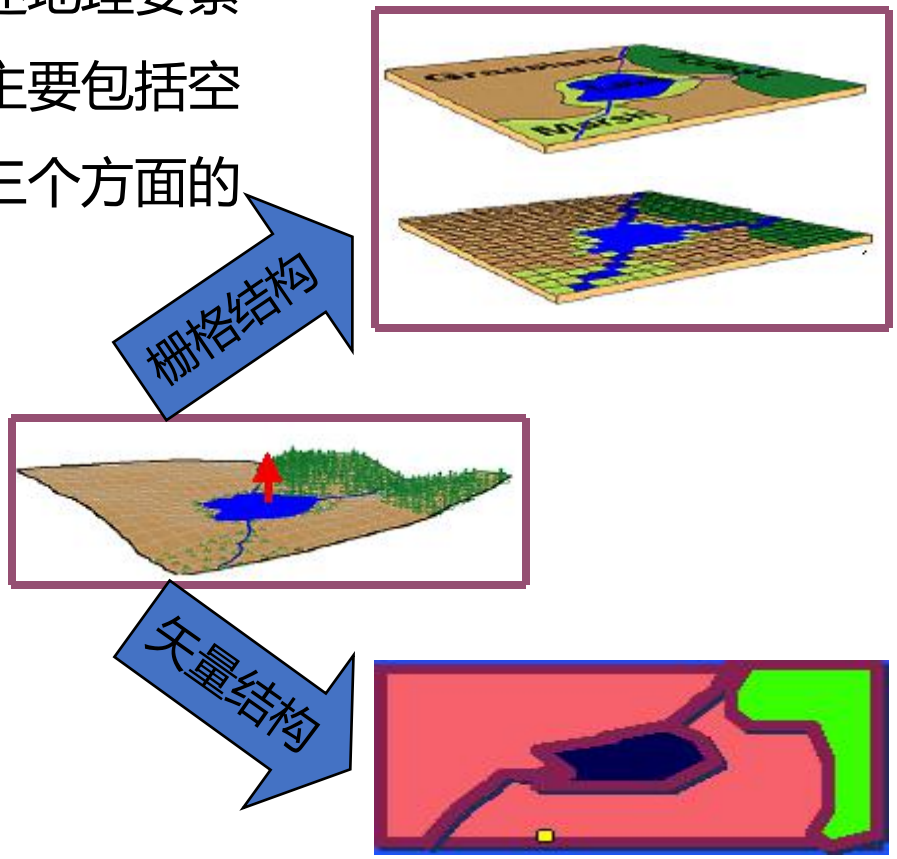
数据结构(data structure)即指数据组织的形式，是适合于计算机存储、管理和处理的数据逻辑结构。对空间数据则是地理实体的空间排列方式和相互关系的抽象描述。

在地理信息系统中描述地理要素和地理现象的空间数据，主要包括空间位置、拓扑关系和属性三个方面的内容。

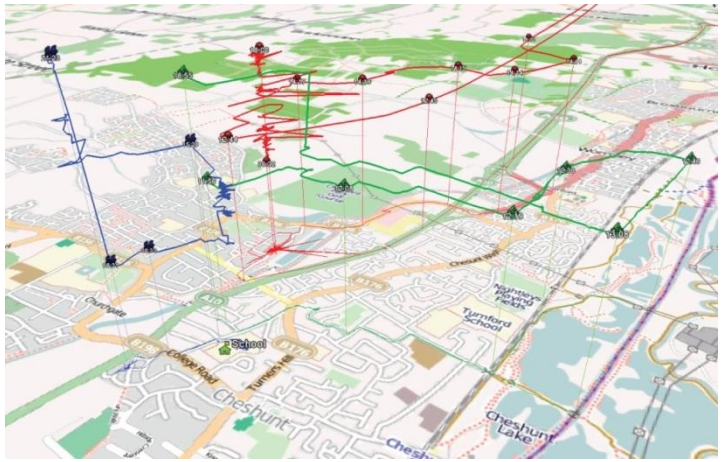
定位

拓扑关系

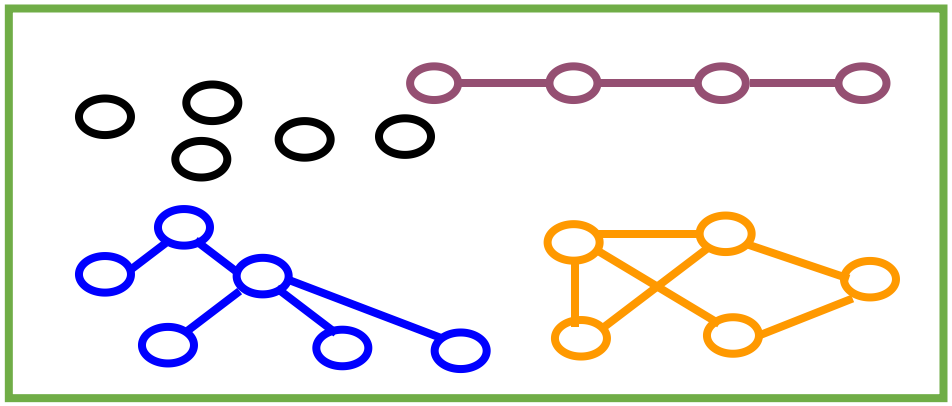
属性



空间数据结构

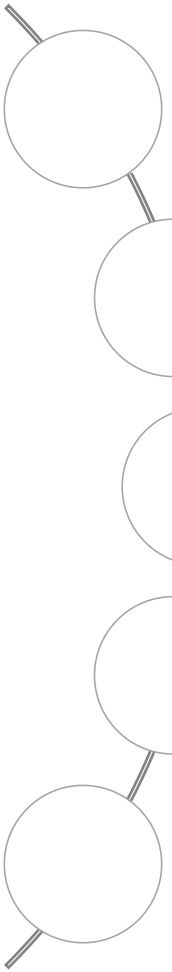


支撑



抽象、基础

本讲大纲



4.1 矢量数据结构

4.2 栅格数据结构

4.3 矢量与栅格数据的融合与转换

4.4 镶嵌数据结构

4.5 多维数据结构

当前内容

4.1 矢量数据结构

4.2 栅格数据结构

4.3 矢量与栅格数据的融合与转换

4.4 镶嵌数据结构

4.5 多维数据结构

4.1 矢量数据结构

当前大纲

4.1.1 实体数据结构

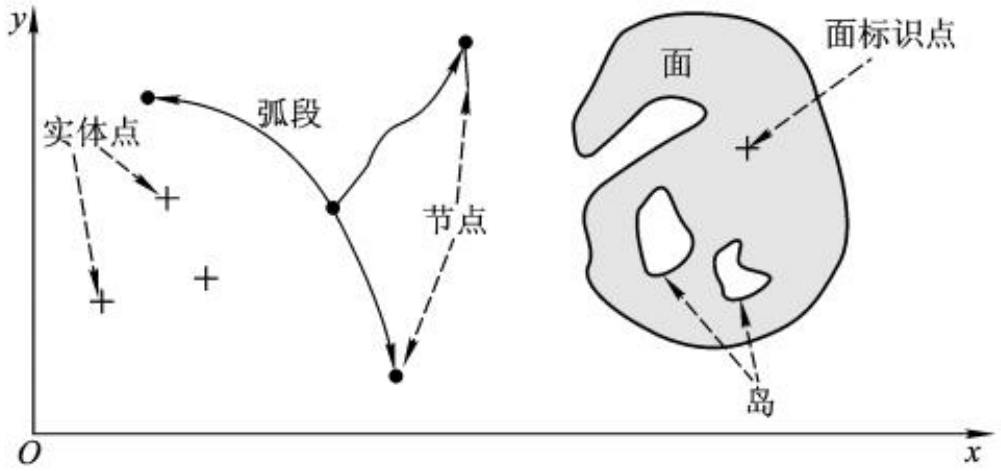
4.1.2 拓扑数据结构

4.1.3 网络数据结构

4.1 矢量数据结构

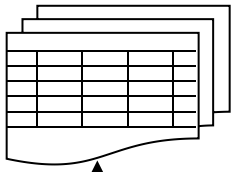
矢量数据结构概述

矢量数据结构(vector data structure)对矢量数据模型进行数据的组织。它通过记录实体坐标及其关系，尽可能精确地表示点、线、面等地理实体，坐标空间为连续空间，允许任意位置、长度和面积的精确定义。



4.1 矢量数据结构

矢量数据结构概述



数据库

独立编码

点: (x , y)
线: (x_1 , y_1) , (x_2 , y_2) , … , (x_n , y_n)
面: (x_1 , y_1) , (x_2 , y_2) , … , (x_1 , y_1)

标识码	属性	几何 位置

空间对象编码
唯一
连接几何和属性数据

点位字典

点: 点号文件
线: 点号串
面: 点号串

点号	X	Y
1	11	22
2	33	44
…	…	…
n	55	66

4.1 矢量数据结构

矢量数据结构概述

矢量数据结构直接以几何空间坐标为基础，记录采样点坐标，通过这种数据组织方式，可以得到精美的地图；另外，该结构

- 可以对复杂数据以最小的数据冗余进行存贮
- 数据精度高
- 存储空间小等特点，是一种高效的图形数据结构。

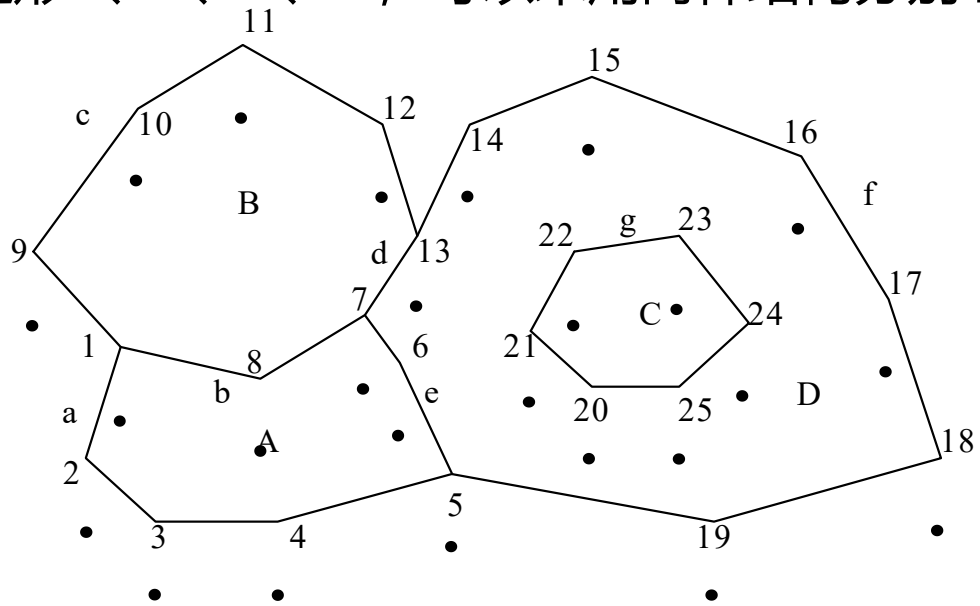
矢量数据结构按其是否明确表示地理实体间的空间关系分为实体数据结构和拓扑数据结构两大类。

4.1 矢量数据结构

4.1.1 实体数据结构

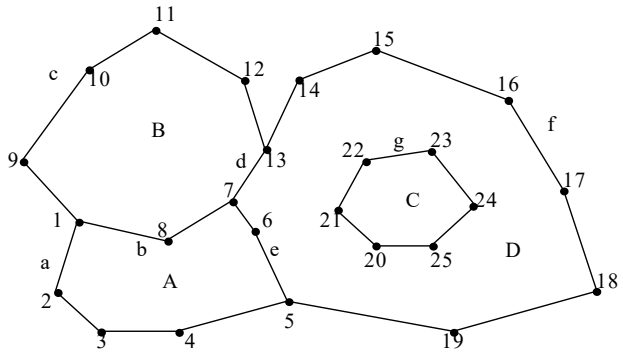
实体数据结构也称**spaghetti数据结构**，是指构成多边形边界的各个线段，以多边形为单元进行组织。按照这种数据结构，边界坐标数据和多边形单元实体一一对应，各个多边形边界点都单独编码并记录坐标。

原始多边形A、B、C、D，可以采用两种结构分别组织。



4.1 矢量数据结构

4.1.1 实体数据结构



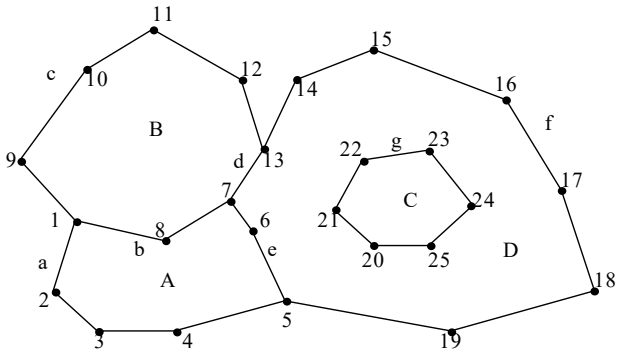
第一种结构：

表 多边形数据文件

多边形 ID	坐标	类别码
A	(x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4), (x_5, y_5), (x_6, y_6), (x_7, y_7), (x_8, y_8), (x_1, y_1)	A102
B	(x_1, y_1), (x_8, y_8), (x_7, y_7), (x_{13}, y_{13}), (x_{12}, y_{12}), (x_{11}, y_{11}), (x_{10}, y_{10}), (x_9, y_9), (x_1, y_1)	B203
C	(x_{20}, y_{20}), (x_{25}, y_{25}), (x_{24}, y_{24}), (x_{23}, y_{23}), (x_{22}, y_{22}), (x_{21}, y_{21}), (x_{20}, y_{20})	A178
D	(x_5, y_5), (x_{19}, y_{19}), (x_{18}, y_{18}), (x_{17}, y_{17}), (x_{16}, y_{16}), (x_{15}, y_{15}), (x_{14}, y_{14}), (x_7, y_7), (x_6, y_6), (x_5, y_5)	C523

4.1 矢量数据结构

4.1.1 实体数据结构



第二种结构：

表 1　点坐标文件

点号	坐标
1	x_1, y_1
2	x_2, y_2
3	x_3, y_3
4	x_4, y_4
.....	.....
25	x_{25}, y_{25}

表 2　多边形文件

多边形 ID	点号串	类别码
A	1,2,3,4,5,6,7,8,1	*A102*
B	1,8,7,13,12,11,10,9,1	*B203*
C	20,25,24,23,22,21,20	*A178*
D	5,19,18,17,16,15,14,7,6,5	*C523*

4.1 矢量数据结构

4.1.1 实体数据结构

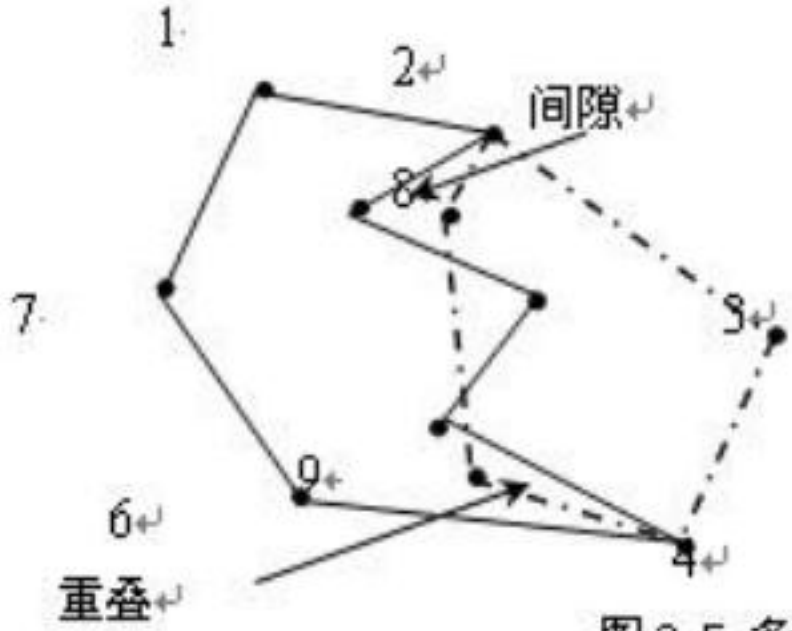
实体数据结构的缺点：

- 相邻多边形的公共边界要数字化两遍，造成数据冗余存储，可能导致输出的公共边界出现间隙或重叠；
- 缺少多边形的邻域信息和图形的拓扑关系；
- 岛只作为一个单个图形，没有建立与外界多边形的联系。

因此，实体式数据结构只适用于简单的系统，如计算机地图制图系统。

4.1 矢量数据结构

4.1.1 实体数据结构



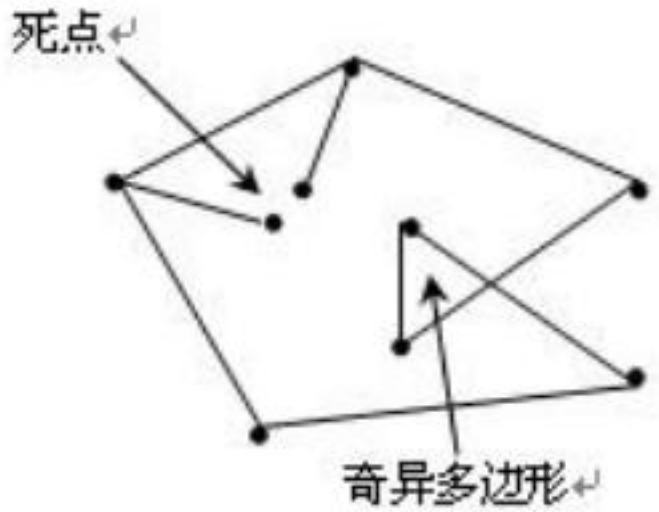


图 3-5 多边形异常

多边形异常

4.1 矢量数据结构

4.1.2 拓扑数据结构

拓扑(topology)关系是一种对空间结构关系进行明确定义的数学方法。具有拓扑关系的矢量数据结构就是拓扑数据结构。

拓扑数据结构没有固定的格式，还没有形成统一标准，但基本原理相同的。

它们的共同的特点是：点是相互独立的，点连成线，线构成面。每条线始于起始结点，止于终止结点，并与左右多边形相邻接。

4.1 矢量数据结构

4.1.2 拓扑数据结构

拓扑数据结构(topological data structure)最重要的特征是具有拓扑编辑功能；这种拓扑编辑功能，不但能够对数字化原始数据进行自动差错编辑，而且可以自动形成封闭的多边形边界，为由各个单独存储的弧段组成的各类多边形及建立空间数据库奠定基础。

拓扑空间数据结构主要有：

- 索引式
- 双重独立编码结构
- 链状双重独立编码结构等

4.1 矢量数据结构

4.1.2 拓扑数据结构

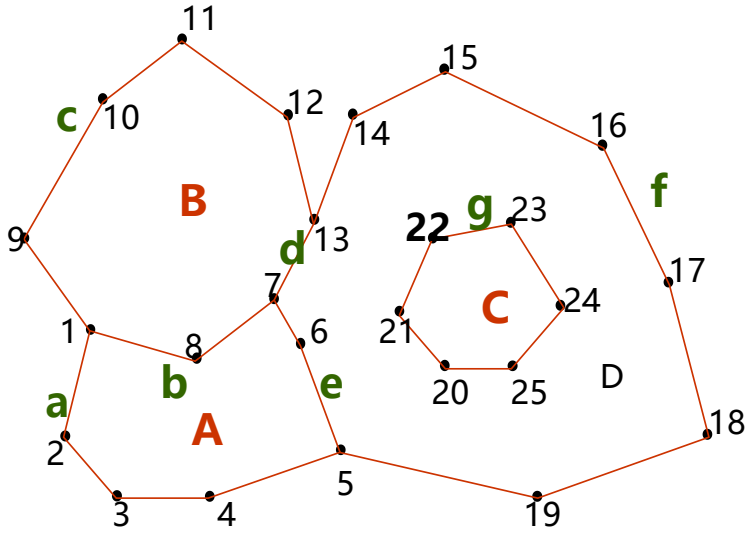
1.索引式结构

索引式数据结构采用树状索引以减少数据冗余并间接增加邻域信息，具体方法是对所有边界点进行数字化，将坐标对以顺序方式存储，由点索引与边界线号相联系，以线索引与各多边形相联系，形成树状索引结构。

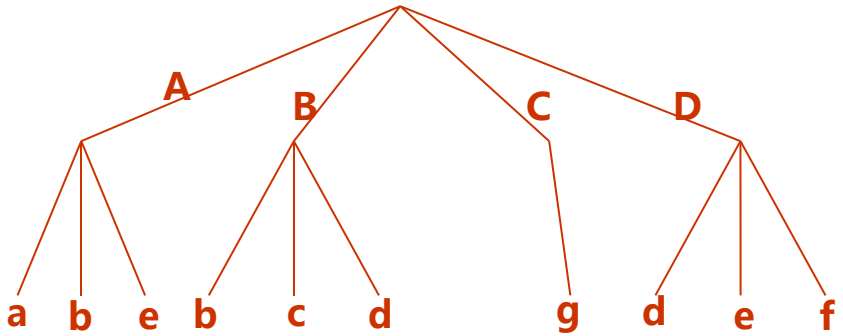
4.1 矢量数据结构

4.1.2 拓扑数据结构

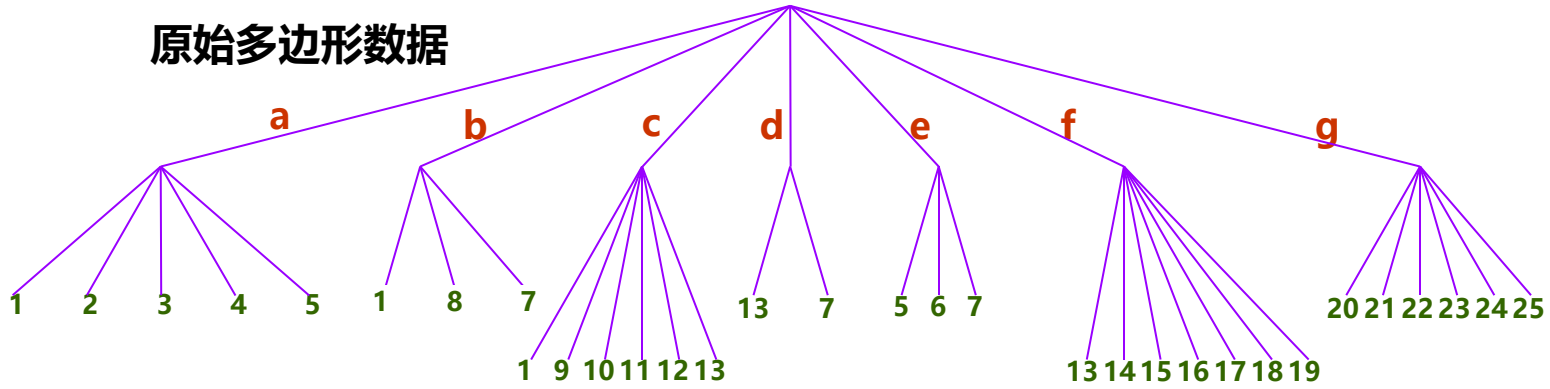
1.索引式结构



原始多边形数据



多边形与线之间索引

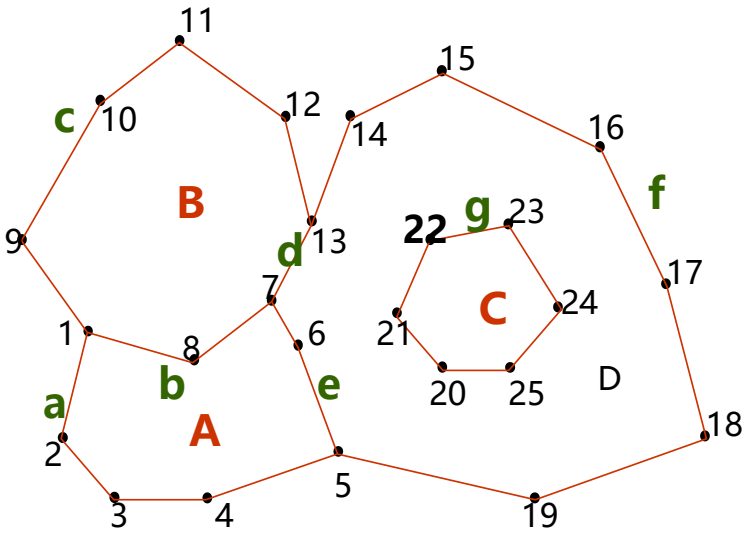


点与线之间的树状索引

4.1 矢量数据结构

4.1.2 拓扑数据结构

1.索引式结构



原始多边形数据

点文件

点ID	坐标
1	x_1,y_1
……	……

边文件

边ID	组成的点ID
a	1,2,3,4,5
……	……

多边形文件

多边形ID	组成的边ID
A	a,b,c
……	……

4.1.2 拓扑数据结构

2.双重独立编码结构

双重独立编码结构最早是由美国人口统计系统采用的一种编码方式，简称**DIME**（Dual Independent Map Encoding）编码系统，它是以城市街道为编码主体，它的特点是采用了拓扑编码结构，这种结构最适合于城市信息系统。

双重独立编码结构是对图上网状或面状要素的任何一条线段，用顺序的两点定义以及相邻多边形来予以定义。

4.1 矢量数据结构

4.1.2 拓扑数据结构

2.双重独立编码结构

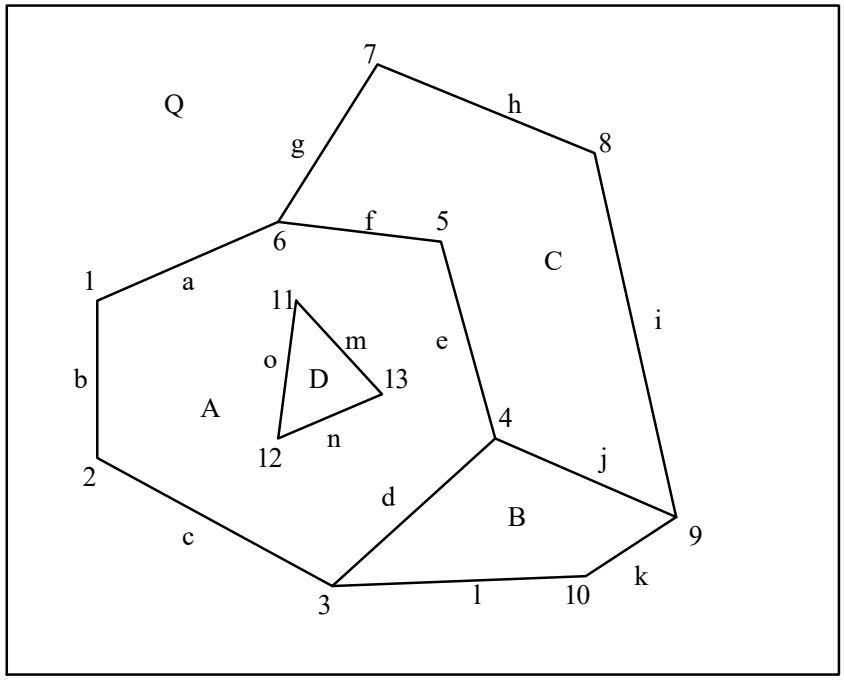


图 多边形原始数据

表 双重独立式编码线文件结构

线号	起点	终点	左多边形	右多边形
a	1	6	Q	A
b	2	1	Q	A
c	3	2	Q	A
d	4	3	B	A
e	5	4	C	A
f	6	5	C	A
g	6	7	Q	C
h	7	8	Q	C
i	8	9	Q	C
j	9	4	B	C
k	9	10	Q	B
l	10	3	Q	B
m	11	13	A	D
n	13	12	A	D
o	12	11	A	D

4.1.2 拓扑数据结构

3.链状双重独立编码结构

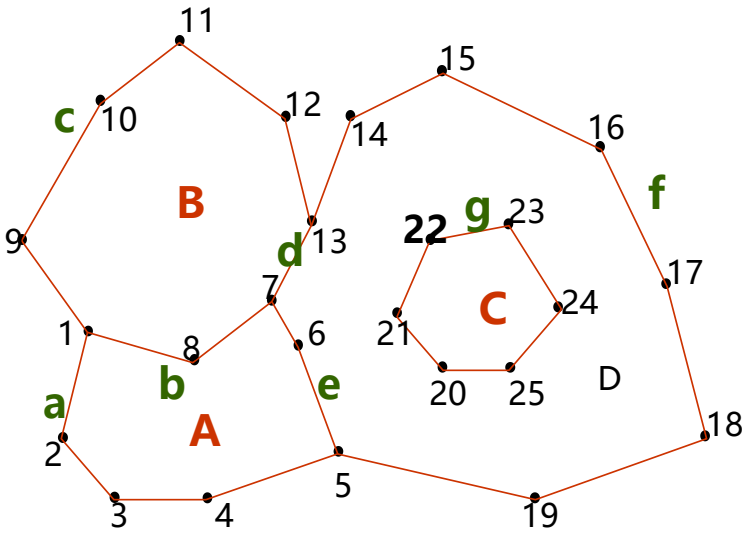
链状双重独立式数据结构是对DIME数据结构的一种改进。在DIME中，一条边只能用直线两端点的序号及相邻的多边形来表示，而在链状数据结构中，将若干直线段合为一个弧段（或链段），每个弧段可以有许多中间点。

国际著名GIS软件平台开发商美国ESRI公司的ARCGIS产品中的COVERAGE数据模型就是采用链状双重独立编码数据结构的。

4.1 矢量数据结构

4.1.2 拓扑数据结构

3.链状双重独立编码结构



原始多边形数据

点号	坐标
1	(x_1, y_1)
2	(x_2, y_2)
……	……
25	(x_{25}, y_{25})

弧段ID	起始点	终结点	左多边形	右多边形
a	5	1	Q	A
b	7	1	A	B
c	1	13	Q	B
d	13	7	D	B
e	7	5	D	A
f	13	5	Q	D
g	25	25	D	C

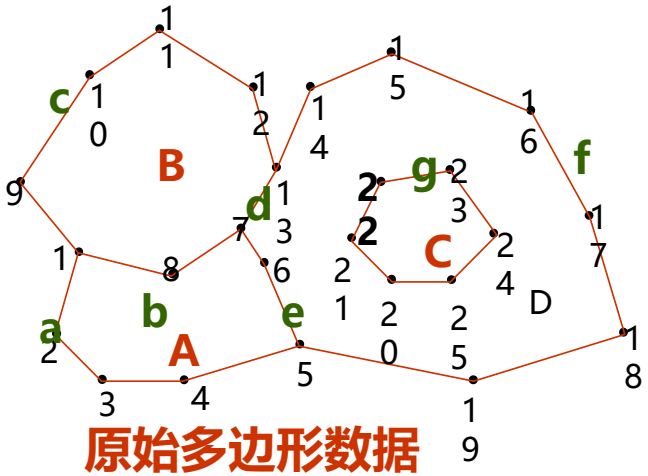
多边形ID	弧段号	属性
A	a,b,e	…
B	c,d,b	…
C	g	…
D	f,e,d,-g	…

弧段ID	点号	弧段ID	点号
a	5,4,3,2,1	e	7,6,5
b	7,8,1	f	13,14,15,16,17,18,19,5
c	1,9,10,11,12,13	g	25,20,21,22,23,24,25
d	13,7		

4.1 矢量数据结构

4.1.2 拓扑数据结构

3.链状双重独立编码结构



原始多边形数据

多边形拓扑文件

多边形ID	弧段号	属性
A	a,b,e	…
B	c,d,b	…
C	g	…
D	f,e,d, -g	…

弧段坐标文件

弧段ID	坐标串	弧段ID	坐标串
a	……	e	……
b	……	f	……
c	……	g	……
d	……		

弧段拓扑文件

弧段ID	起始点	终结点	左多边形	右多边形
a	5	1	Q	A
b	7	1	A	B
c	1	13	Q	B
d	13	7	D	B
e	7	5	D	A
f	13	5	Q	D
g	25	25	D	C

4.1.3 网络数据结构

网络数据结构（network data structure）由一组相连的边和交汇点以及连通性规则组成，用于表示现实世界中的网状线性系统。

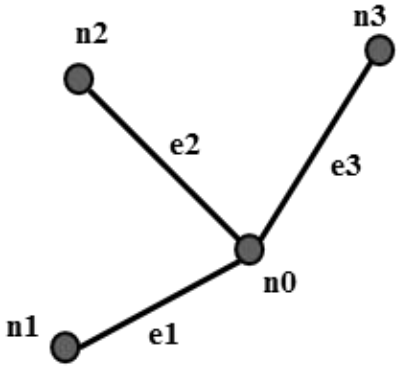
网络数据结构是GIS数据建模和空间分析所必须的一种常用数据结构。采用网络结构进行建模的常见实体包括交通网络系统、电力系统、地下管网系统和河网系统等。

在GIS网络建模中，根据网络是否记录位置特征可以将其分为：

- 几何网络
- 逻辑网络

4.1 矢量数据结构

4.1.3 网络数据结构



网络原始数据

表 节点文件

	节点	节点几何
1	n0	(x0, y0)
2	n1	(x1, y1)
3	n2	(x2, y2)
4	n3	(x3, y3)

表 边文件

	边	节点几何
1	e1	(x0, y0) (x1, y1)
2	e2	(x0, y0) (x2, y2)
3	e3	(x0, y0) (x3, y3)

表 节点-边关系文件

	节点	节点和边的关系		
1	n0	(n1, e1)	(n2, e2)	(n3, e3)
2	n1	(n0, e1)		
3	n2	(n0, e2)		
4	n3	(n0, e2)		

4.1.3 网络数据结构

以上只是一个简单网络的几何数据结构，在实际应用中，网络建模要复杂得多。

在较为复杂的网络中，可能需要考虑方向，还可能包含一级或多级的子节点和边，甚至还需要对转弯规则进行建模，如在道路建模中，某个道路交叉口只能左转弯，这都需要在网络结构中添加相应的规则。

对于简单网络数据结构，其编码方式同实体数据结构和拓扑数据结构的编码方式类似，但在复杂网络数据结构中，考虑的规则较多，其建模的形式也有所不同。

当前内容

4.1 矢量数据结构

4.2 栅格数据结构

4.3 矢量与栅格数据的融合与转换

4.4 镶嵌数据结构

4.5 多维数据结构

4.2 栅格数据结构

当前大纲

4.2.1 完全栅格数据结构

4.2.2 压缩栅格数据结构

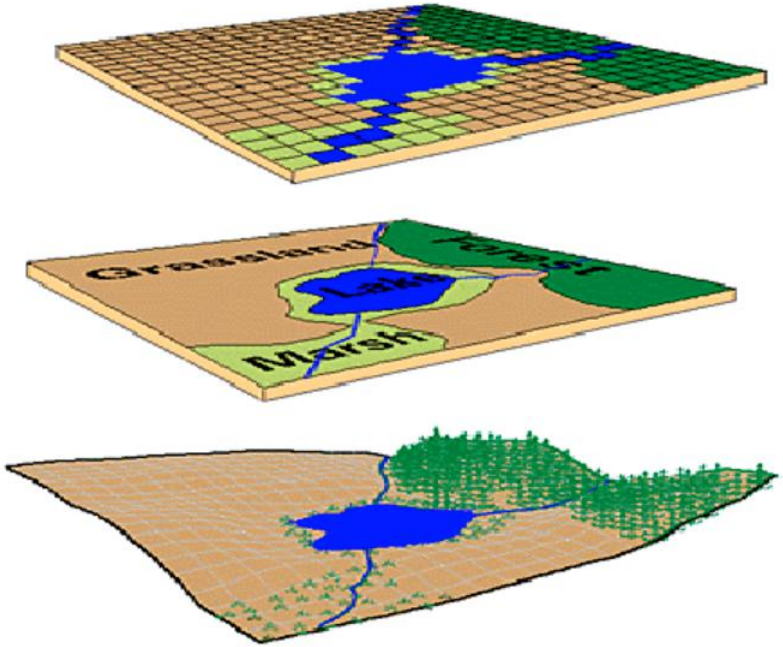
4.2.3 链码结构

4.2.4 影像与瓦片金字塔结构

4.2 栅格数据结构

栅格数据结构概述

以规则栅格阵列表示空间对象的数据结构称为**栅格数据结构**(raster data structure)。阵列中每个栅格单元上的数值表示空间对象的属性特征。即栅格阵列中每个单元的行列号确定位置，属性值表示空间对象的类型、等级等特征。每个栅格单元只能存在一个值。



4.2 栅格数据结构

栅格数据结构概述

栅格数据结构的显著特点是：属性明显，定位隐含，即数据直接记录属性的指针或属性本身，而所在位置则根据行列号转换为相应的坐标给出，也就是说定位是根据数据在数据集中的位置得到的。

栅格数据的优缺点：

- 优点：数据结构简单、数学模拟方便
- 缺点：数据量大、难以建立实体间的拓扑关系、通过改变分辨率而减少数据量时精度和信息量同时受损等。

4.2 栅格数据结构

4.2.1 完全栅格数据结构

完全栅格数据结构（也称编码）将栅格看作一个数据矩阵，逐行逐个记录栅格单元的值。

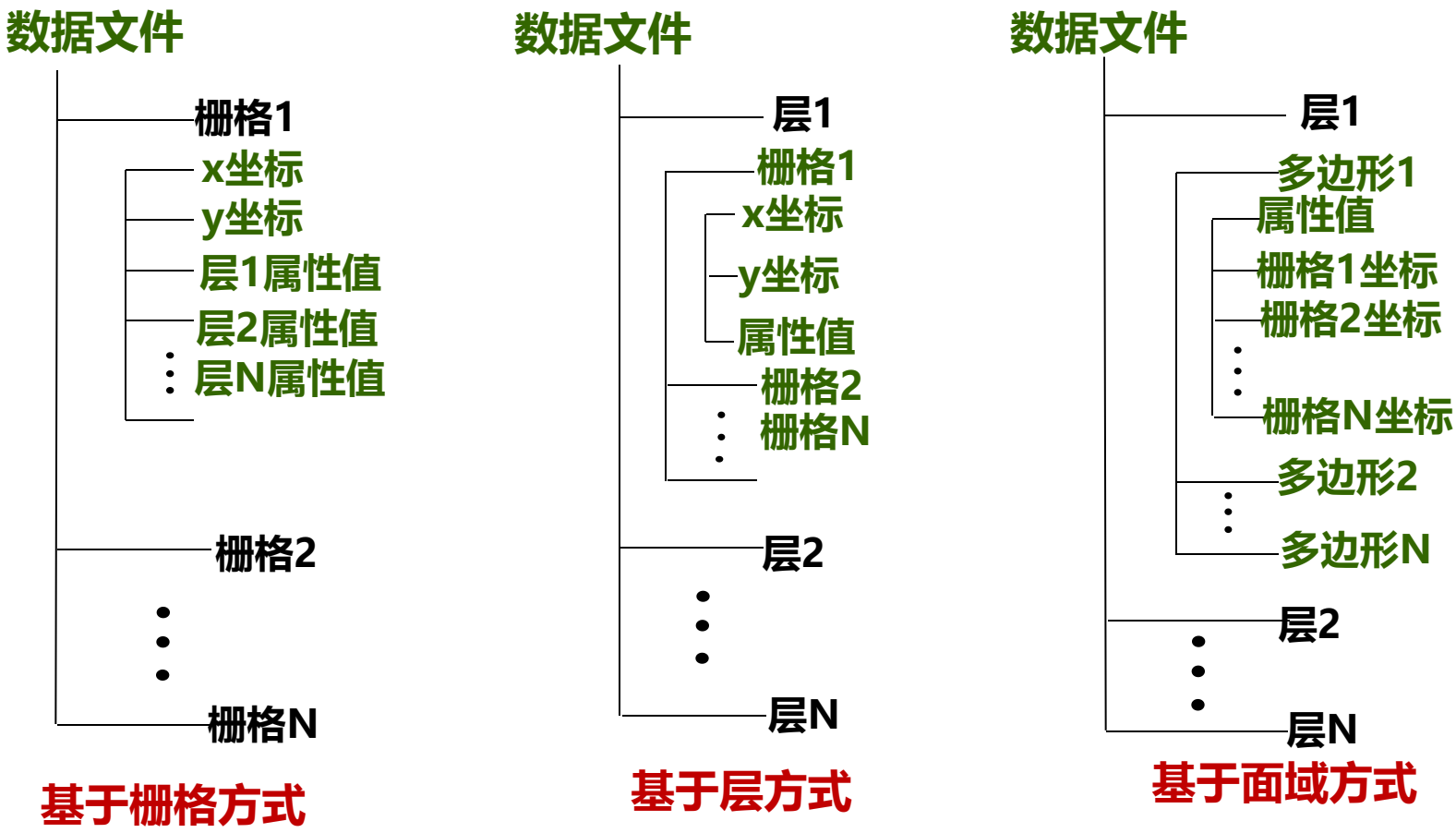
这是最简单最直接的一种栅格编码方法。通常这种编码为栅格文件或格网文件，它不采用任何压缩数据的处理。

完全栅格数据的组织有三种基本方式：

- 基于象元
- 基于层
- 基于面域

4.2 栅格数据结构

4.2.1 完全栅格数据结构

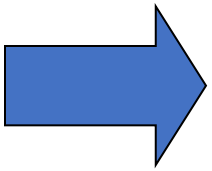


4.2 栅格数据结构

4.2.2 压缩栅格数据结构

1.游程长度编码结构（一）

0	0	0	0	0	4	4	4
0	0	0	4	4	4	4	4
0	0	4	4	4	4	8	8
0	0	4	4	4	8	8	8
2	2	4	4	8	8	8	8
2	2	2	4	8	8	8	8
2	2	2	2	8	8	8	8
2	2	2	2	8	8	8	8



(s_i, l_i)

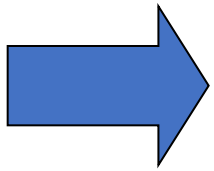
(0,5),(4,3)
(0,3),(4,5)
(0,2),(4,4),(8,2)
(0,2),(4,3),(8,3)
(2,2),(4,2),(8,4)
(2,3),(4,1),(8,4)
(2,4),(8,4)
(2,4),(8,4)

4.2 栅格数据结构

4.2.2 压缩栅格数据结构

1.游程长度编码结构（二）

0	0	0	0	0	4	4	4
0	0	0	4	4	4	4	4
0	0	4	4	4	4	8	8
0	0	4	4	4	8	8	8
2	2	4	4	8	8	8	8
2	2	2	4	8	8	8	8
2	2	2	2	8	8	8	8
2	2	2	2	8	8	8	8



(s_i, pos_i)

(0,5),(4,8)
(0,3),(4,8)
(0,2),(4,6),(8,8)
(0,2),(4,5),(8,8)
(2,2),(4,4),(8,8)
(2,3),(4,4),(8,8)
(2,4),(8,8)
(2,4),(8,8)

4.2.2 压缩栅格数据结构

1.游程长度编码结构

我们可以发现，压缩比的大小与图的复杂程度成反比的，在变化多的部分，游程数就多，变化少的部分游程数就少，原始栅格类型越简单，压缩效率就越高。

因此这种数据结构最适宜于类型面积较大的专题要素、遥感图像的分类结构，而不适合于类型连续变化或类型分散的分类图。

4.2 栅格数据结构

4.2.2 压缩栅格数据结构

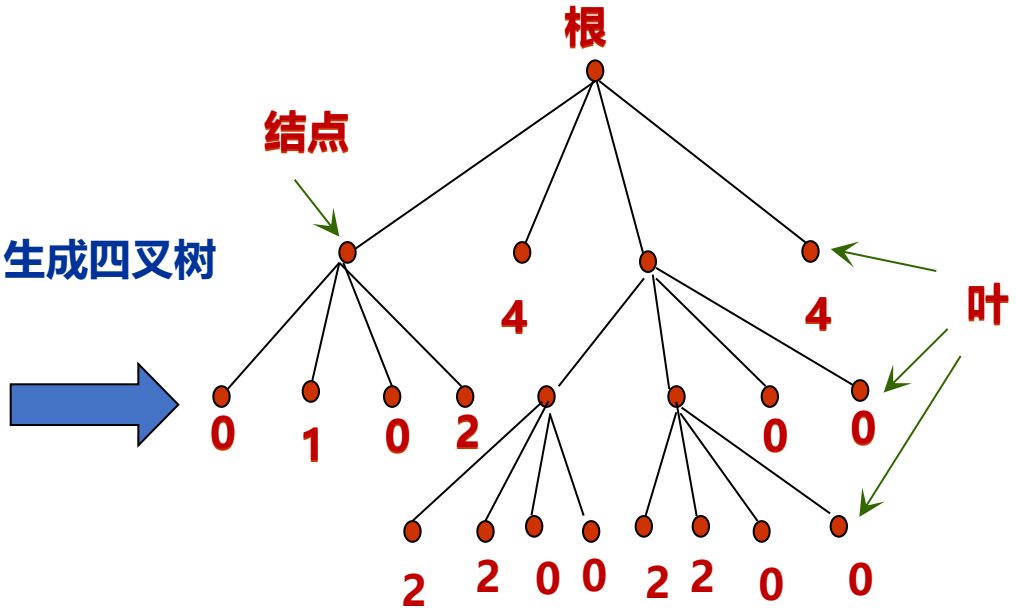
2.四叉树数据结构

栅格单元数满足$2^n \times 2^n$，递归分割，直到每一区间相同或不可再分割。

- 常规四叉树

0	0	1	1	4	4	4	4
0	0	1	1	4	4	4	4
0	0	2	2	4	4	4	4
0	0	2	2	4	4	4	4
2	2	2	2	4	4	4	4
0	0	0	0	4	4	4	4
0	0	0	0	4	4	4	4
0	0	0	0	4	4	4	4

生成四叉树

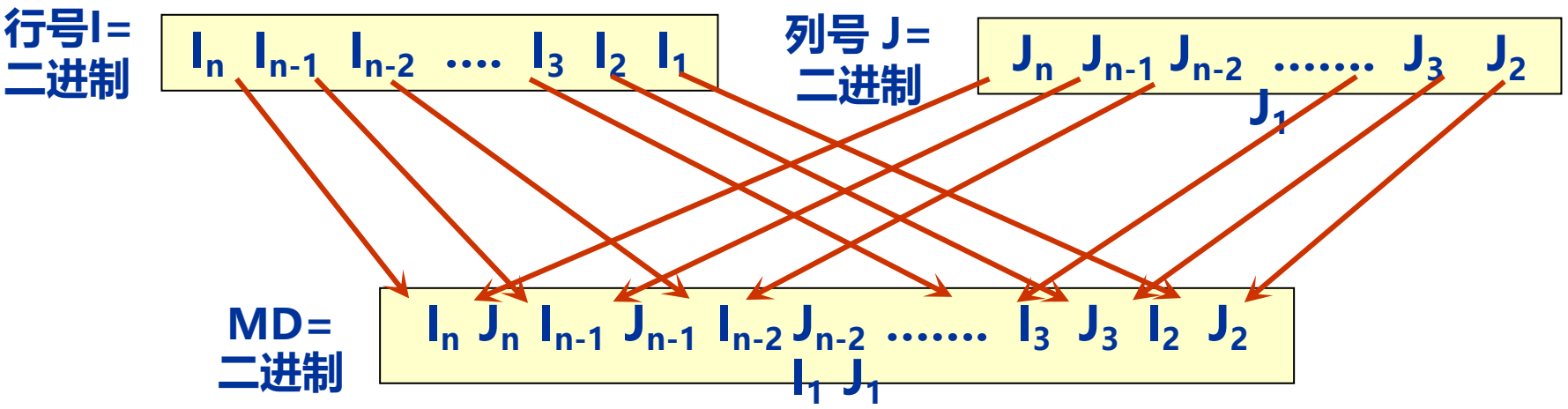


4.2 栅格数据结构

4.2.2 压缩栅格数据结构

●线性四叉树

- 线性四叉树则只存贮最后叶结点的信息，包括叶结点的位置编码/地址码、属性或灰度值；
- 线性四叉树地址码，通常采用十进制Morton码（M_D码）：



M_D码的“位”运算生成

4.2 栅格数据结构

4.2.2 压缩栅格数据结构

列方向 →

行方向 ↓

	0	1	2	3	4	5	6	7	8
0	0	1	4	5	16	17	20	21	64
1	2	3	6	7	18	19	22	23	66
2	8	9	12	13	24	25	28	29	72
3	10	11	14	15	26	27	30	31	74
4	32	33	36	37	48	49	52	53	96
5	34	35	38	39	50	51	54	55	98
6	40	41	44	45	56	57	60	61	104
7	42	43	46	47	58	59	62	63	106
8	128	129	132	133	144	145	148	149	192

M_D码实例

4.2 栅格数据结构

4.2.2 压缩栅格数据结构

0	0	1	1	4	4	4	4
0	0	1	1	4	4	4	4
0	0	2	2	4	4	4	4
0	0	2	2	4	4	4	4
2	2	2	2	4	4	4	4
0	0	0	0	4	4	4	4
0	0	0	0	4	4	4	4
0	0	0	0	4	4	4	4

按M_D码顺序
依次提取4个
栅格单元

MD	属性
28	4
29	4
30	4
31	4

栈顶指针

比较

入栈

栈

28	4
24	4
20	4
16	4
12	2
8	0
4	1
0	0
MD	属性

栈顶指针

判别

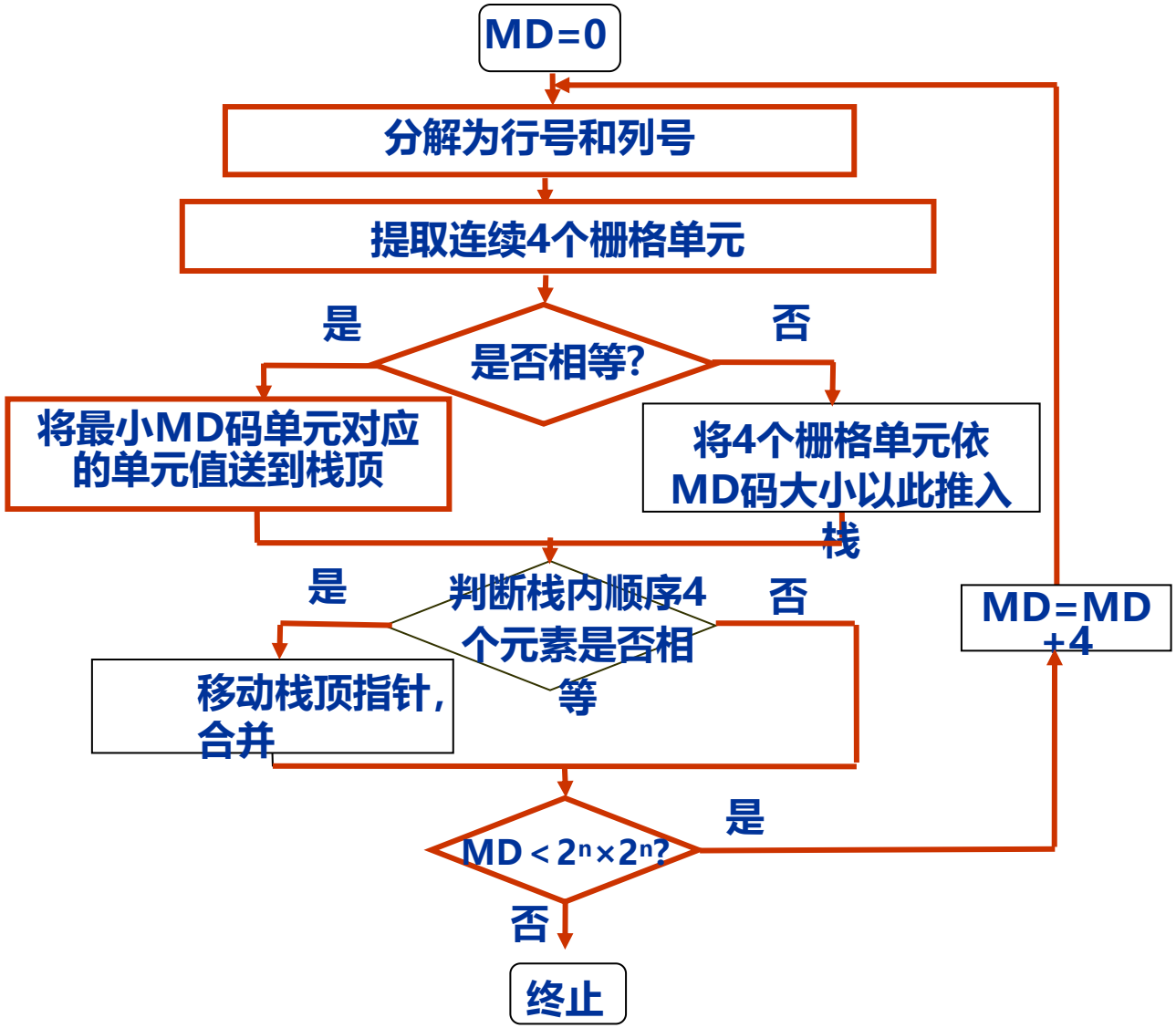
压栈

栈

16	4
12	2
8	0
4	1
0	0
MD	属性

4.2 栅格数据结构

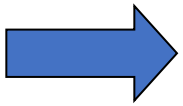
4.2.2 压缩栅格数据结构



4.2 栅格数据结构

4.2.2 压缩栅格数据结构

0	0	1	1	4	4	4	4
0	0	1	1	4	4	4	4
0	0	2	2	4	4	4	4
0	0	2	2	4	4	4	4
2	2	2	2	4	4	4	4
0	0	0	0	4	4	4	4
0	0	0	0	4	4	4	4
0	0	0	0	4	4	4	4



M_D码	属性值
0	0
4	1
8	0
12	2
16	4
32	2
33	2
34	0
35	0
36	2
37	2
38	0
39	0
40	0
44	0
48	4

4.2 栅格数据结构

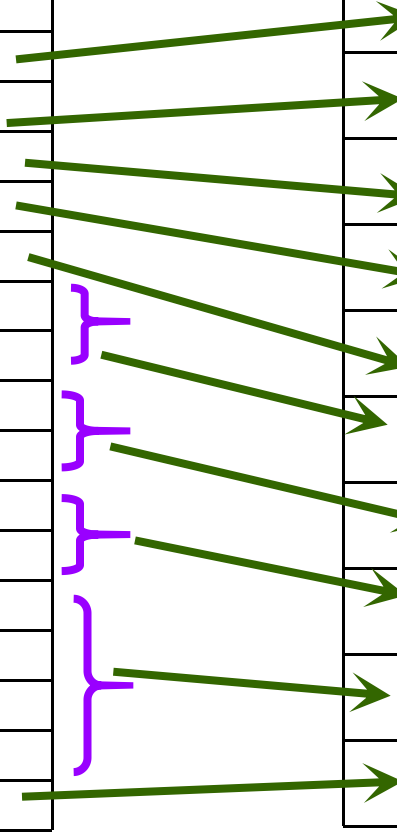
4.2.2 压缩栅格数据结构

3、二维行程编码结构

对线性四叉树中仍存在前后叶结点相同值的情况，

进一步压缩数据，将前后值相同的叶结点归并：

M_D码	属性值
0	0
4	1
8	0
12	2
16	4
32	2
33	2
34	0
35	0
36	2
37	2
38	0
39	0
40	0
44	0
48	4

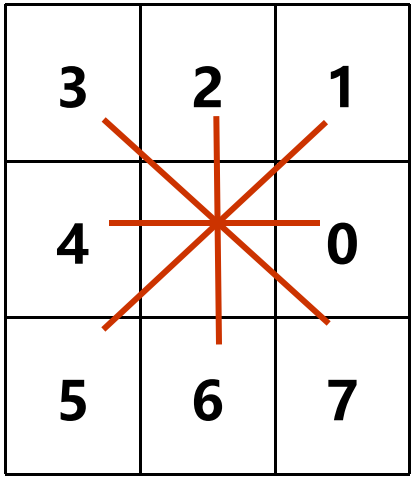


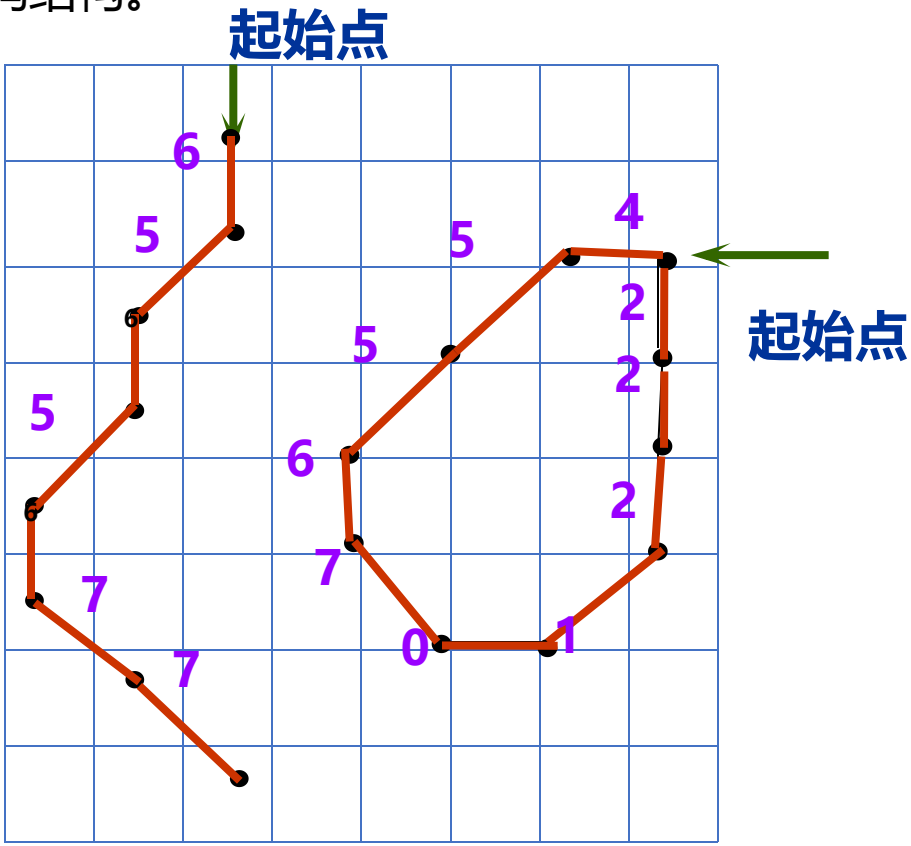
二维行程码	属性值
0	0
4	1
8	0
12	2
16	4
32	2
34	0
36	2
38	0
48	4

4.2 栅格数据结构

4.2.3 链码结构

链码数据结构首先采用弗里曼（Freeman）码对栅格中的线或多边形边界进行编码，然后再组织为链码结构。





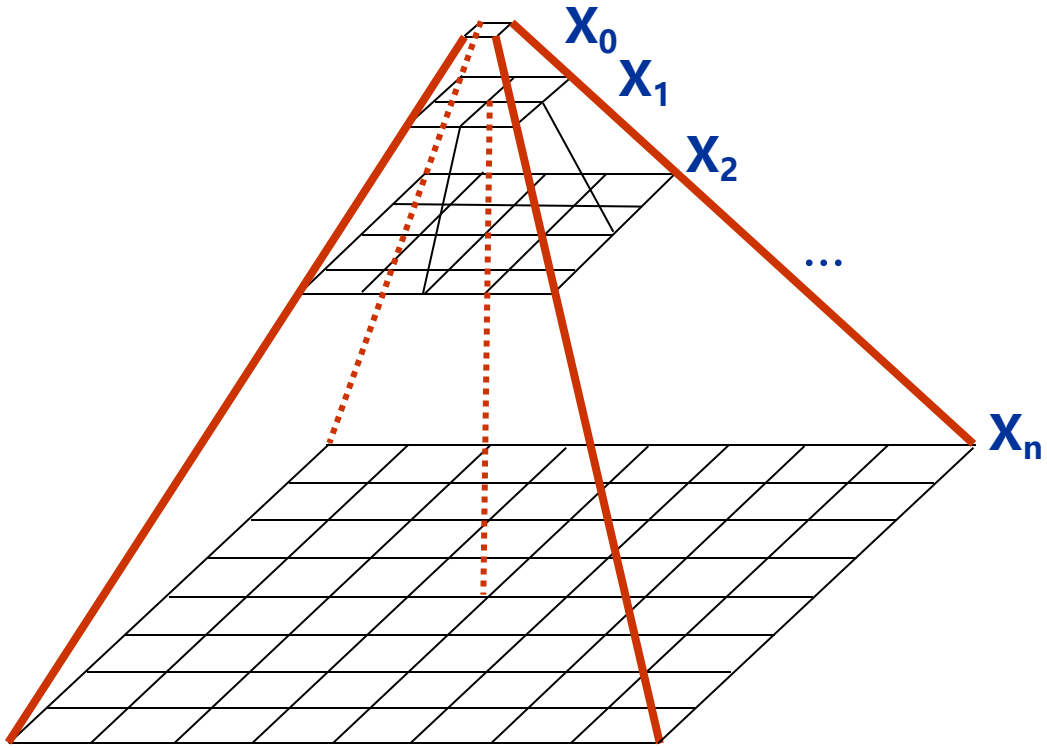
链码结构文件

特征码	起点行	起点列	链码
2	1	3	6,5,6,5,6,7,7
7	2	8	4,5,5,6,7,0,1,2,2,2

4.2 栅格数据结构

4.2.4 影像与瓦片金字塔结构

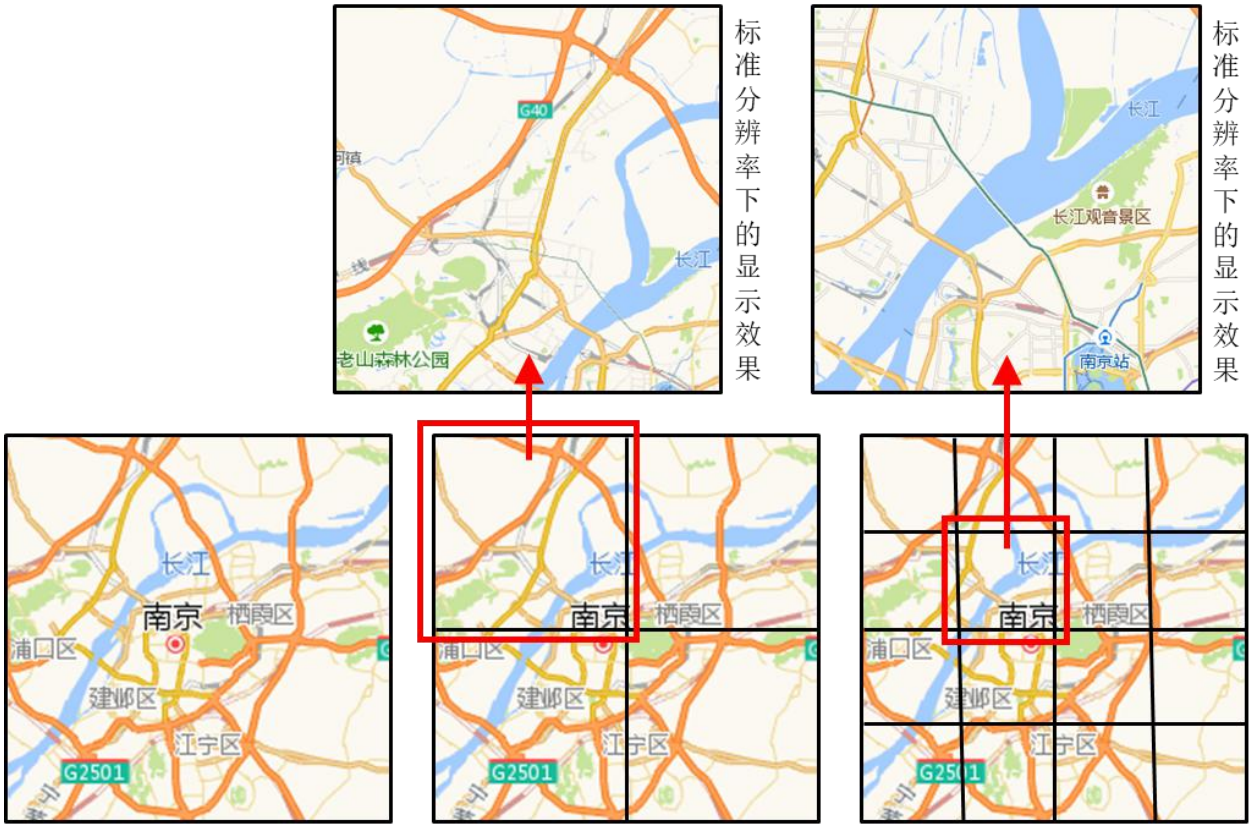
影像金字塔结构用于图像编码和渐进式图像传输，是一种典型的分层数据结构形式，适合于栅格数据和影像数据的多分辨率组织，也是一种栅格数据或影像数据的有损压缩方式，有M-金字塔，T-金字塔等。



M-金字塔结构

4.2 栅格数据结构

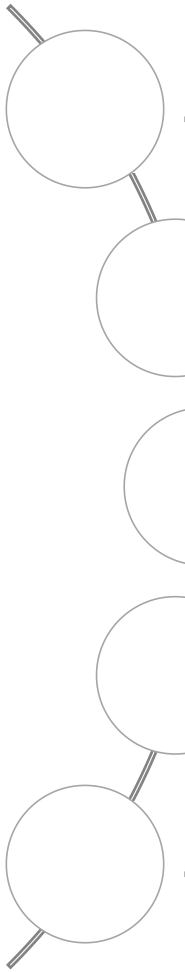
4.2.4 影像与瓦片金字塔结构



(a) 第N级瓦片　(b) 第N+1级瓦片　(c) 第N+2级瓦片

无论由多少切片组成，其分辨率都保持不变，因此任何级别的请求都可以看到相同分辨率的切片服务。

当前内容



4.1 矢量数据结构

4.2 栅格数据结构

4.3 矢量与栅格数据的融合与转换

4.4 镶嵌数据结构

4.5 多维数据结构

4.3 矢量与栅格数据的融合与转换

当前大纲

4.3.1 矢量数据与栅格数据结构的比较

4.3.2 矢栅一体化数据结构

4.3.3 矢量数据与栅格数据结构的转化

4.3 矢量与栅格数据的融合与转换

4.3.1 矢量数据与栅格数据结构的比较

	优点	缺点
矢量数据结构	➢ 数据结构严密，冗余度小，数据量小； ➢ 空间拓扑关系清晰，易于网络分析； ➢ 面向对象目标的，不仅能表达属性编码，而 且能方便地记录每个目标的具体的属性描述 信息； ➢ 能够实现图形数据的恢复、更新和综合； ➢ 图形显示质量好、精度高。	➢ 数据结构处理算法复杂 ➢ 叠置分析与栅格图组合比较难； ➢ 数学模拟比较困难； ➢ 空间分析技术上比较复杂，需要更复杂的软、硬件条件； ➢ 显示与绘图成本比较高。
栅格数据结构	➢ 数据结构简单，易于算法实现； ➢ 空间数据的叠置和组合容易，有利于与遥感数据的匹配应用和分析； ➢ 各类空间分析，地理现象模拟均较为容易； ➢ 输出方法快速建议，成本低廉。	➢ 图形数据量大，用大像元减小数据量时，精度和信息量受损失； ➢ 难以建立空间网络连接关系； ➢ 投影变化实现困难； ➢ 图形数据质量低，地图输出不精美。

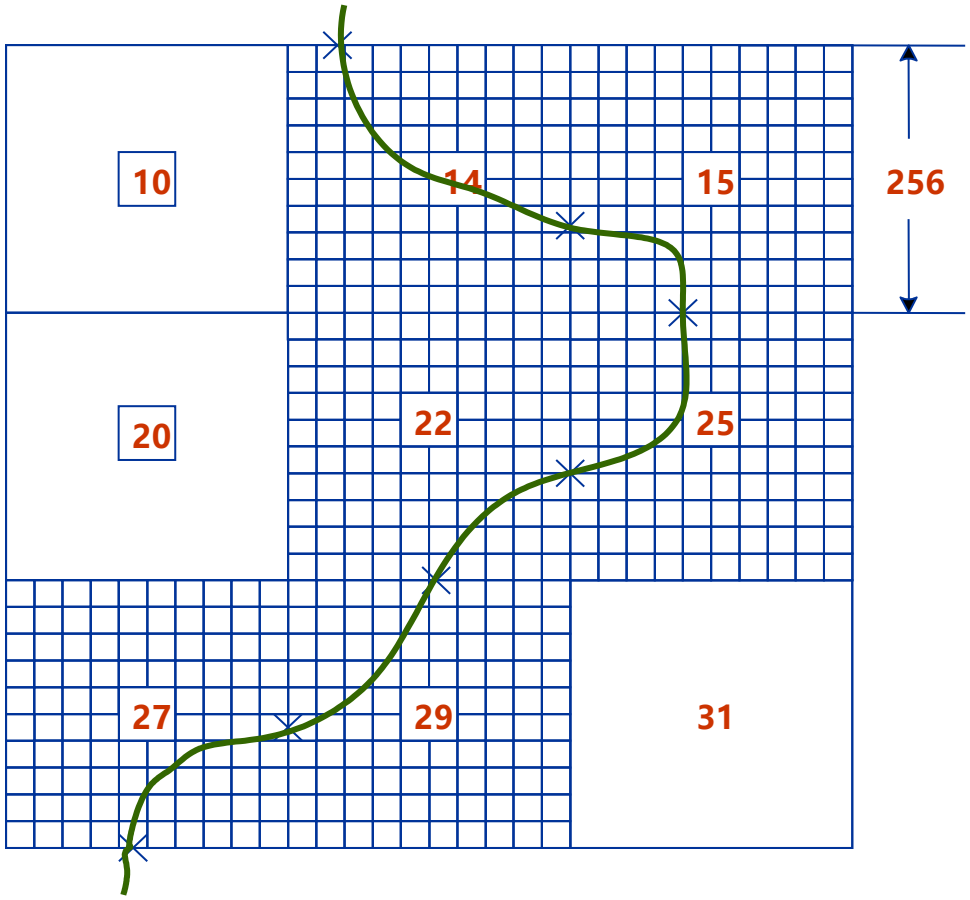
4.3 矢量与栅格数据的融合与转换

4.3.2 矢栅一体化数据结构

将地理区域划分成多级格网：

- ✓ 粗格网
- ✓ 基本格网
- ✓ 细分格网

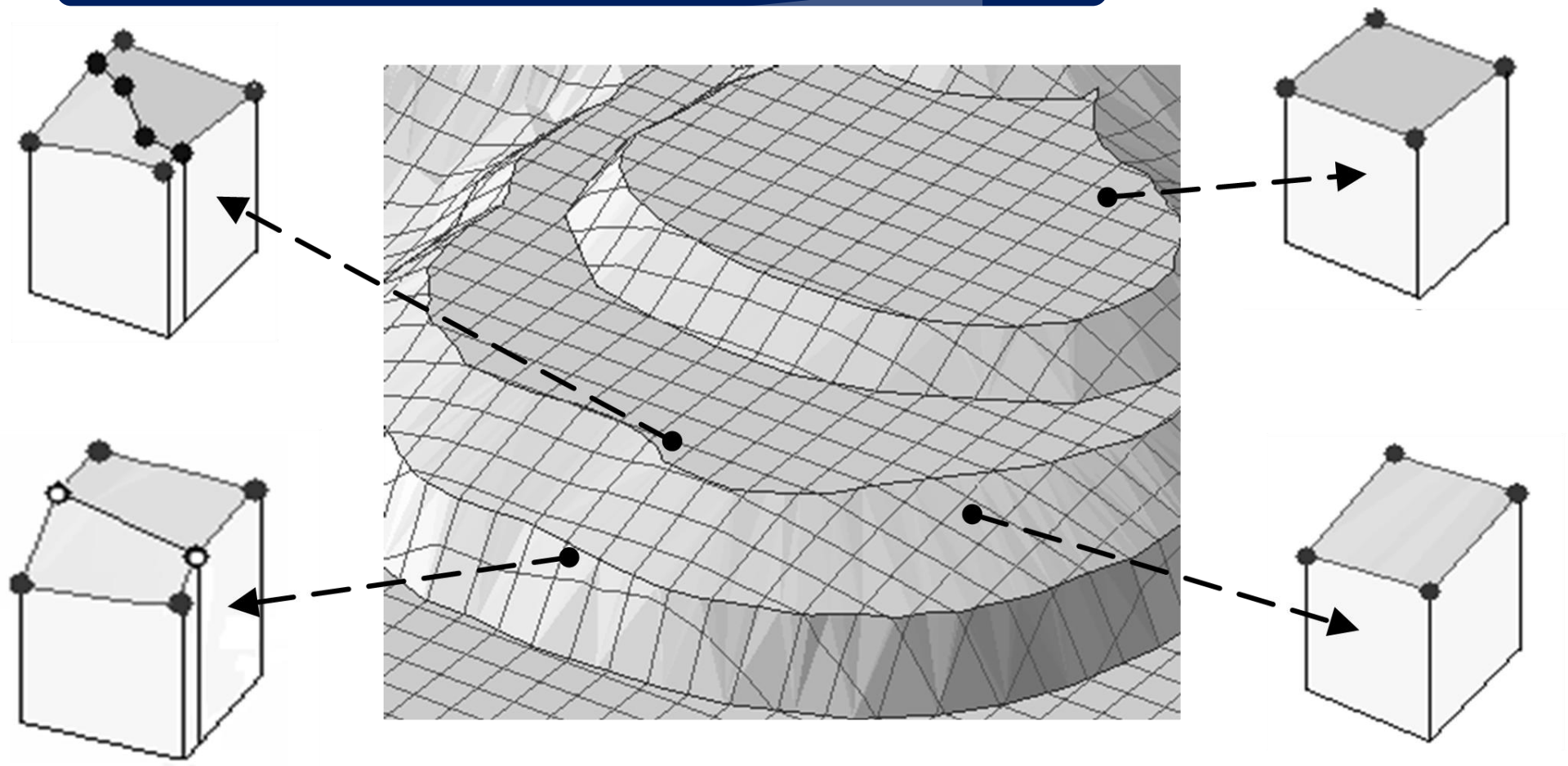
三级格网都采用线性四叉树编码用（**M0、M1、M2**）表示，其中**M0**表示点或线所通过的粗格网的MD码，是研究区的整体编码；**M1**表示点或线所通过的基本格网的Morton码，也是研究区的整体编码；**M2**表示点或线所通过的细分格网的Morton码，是基本栅格内的局部编码



一体化数据结构细分格网

4.3 矢量与栅格数据的融合与转换

4.3.2 矢栅一体化数据结构



4.3 矢量与栅格数据的融合与转换

4.3.2 矢栅一体化数据结构

点状地物一体化数据结构

点标识号	M_1	M_2	属性值

线状地物一体化数据结构

弧ID	起点ID	终点ID	左域ID	右域ID	中间点坐标（M_1，M_2）序列	……

面状地物一体化数据结构

面域ID	边界弧ID序列	面域内点指针	……

面域内点指针	面域内点坐标（M_1，M_2）序列

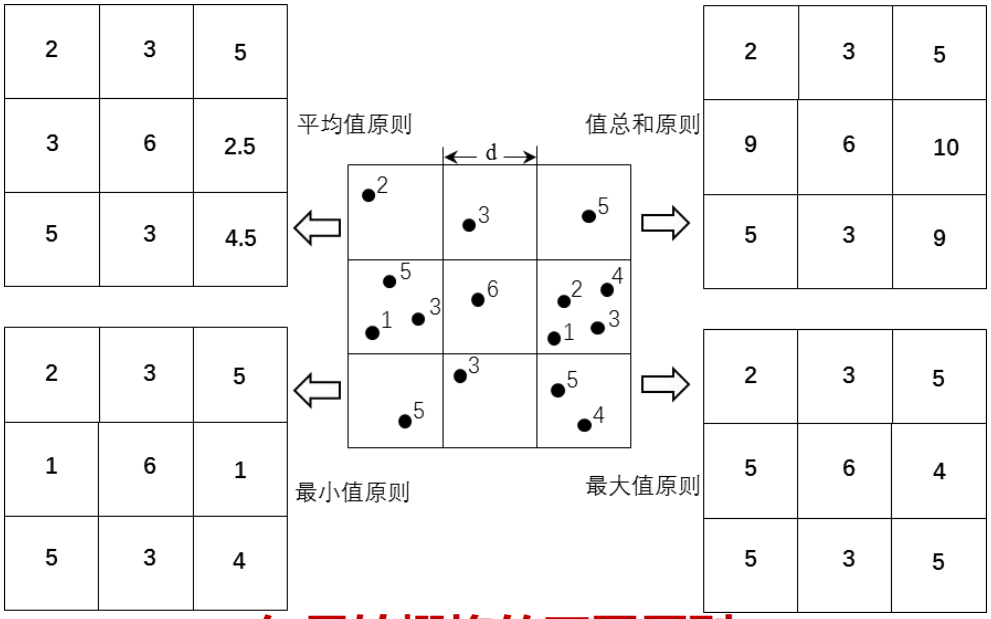
4.3 矢量与栅格数据的融合与转换

4.3.3 矢量数据与栅格数据结构的转换

1.矢量数据到栅格数据的结构转换

矢量数据转换为栅格数据之前，需要先确定栅格像元的大小、像元值分配类型及分配原则。

栅格像元的大小决定了数据的输出精度。像元值的分配类型和分配原则主要用于确定多个对象落入同一个像元中时像元值的计算方式。



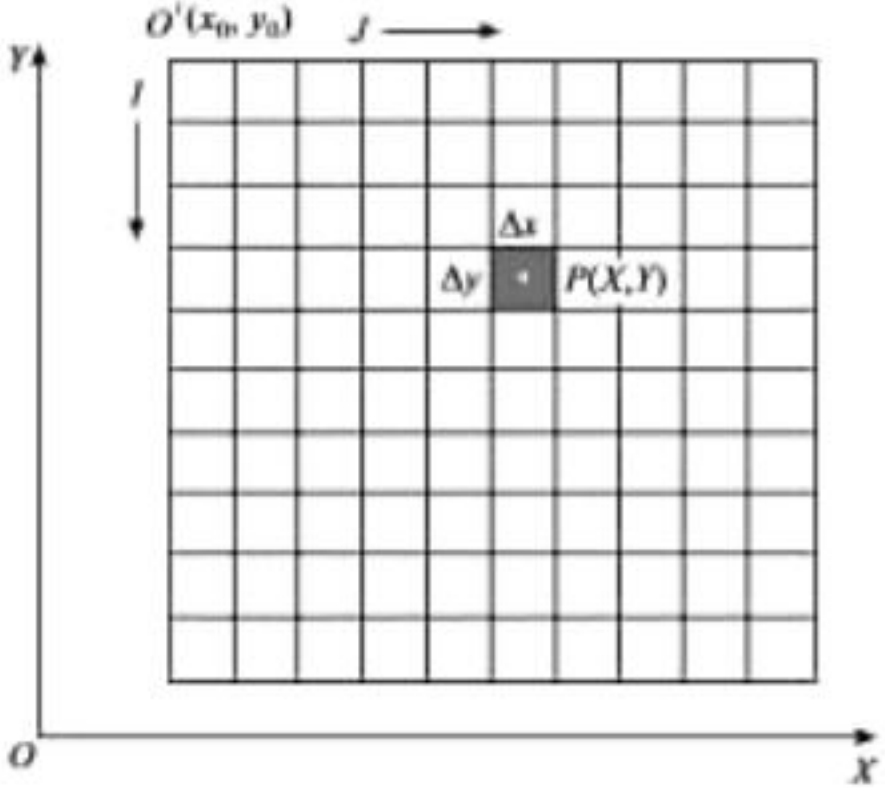
矢量转栅格的不同原则

4.3 矢量与栅格数据的融合与转换

4.3.3 矢量数据与栅格数据结构的转换

1.矢量数据到栅格数据的结构转换

（1）点的转换



$$\Delta X = (X_{\max} - X_{\min}) / N$$
$$\Delta Y = (Y_{\max} - Y_{\min}) / M$$

$$\begin{cases} I = 1 + Int((Y_0 - Y) / \Delta Y) \\ J = 1 + Int((X - X_0 / \Delta X) \end{cases}$$

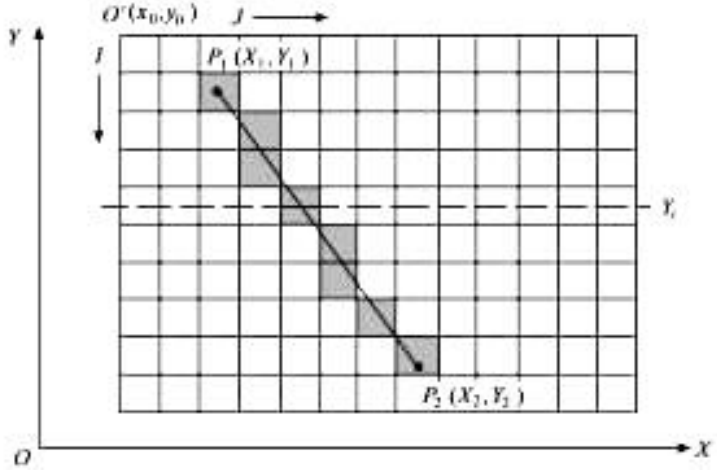
4.3 矢量与栅格数据的融合与转换

4.3.3 矢量数据与栅格数据结构的转换

1.矢量数据到栅格数据的结构转换

（2）线的转换

八方向栅格化法：核心是对任意直线段如何转换成栅格数据



为避免转换后栅格像元线段可能出现的间断现象，分两种情况判断

若行数差大于列数差

$$\begin{cases} Y = Y_i \\ X = (Y - Y_1) \times (X_2 - X_1) / (Y_2 - Y_1) + X_1 \end{cases}$$

若列数差大于行数差

$$\begin{cases} X = X_i \\ Y = (X - X_1) \times (Y_2 - Y_1) / (X_2 - X_1) + Y_1 \end{cases}$$

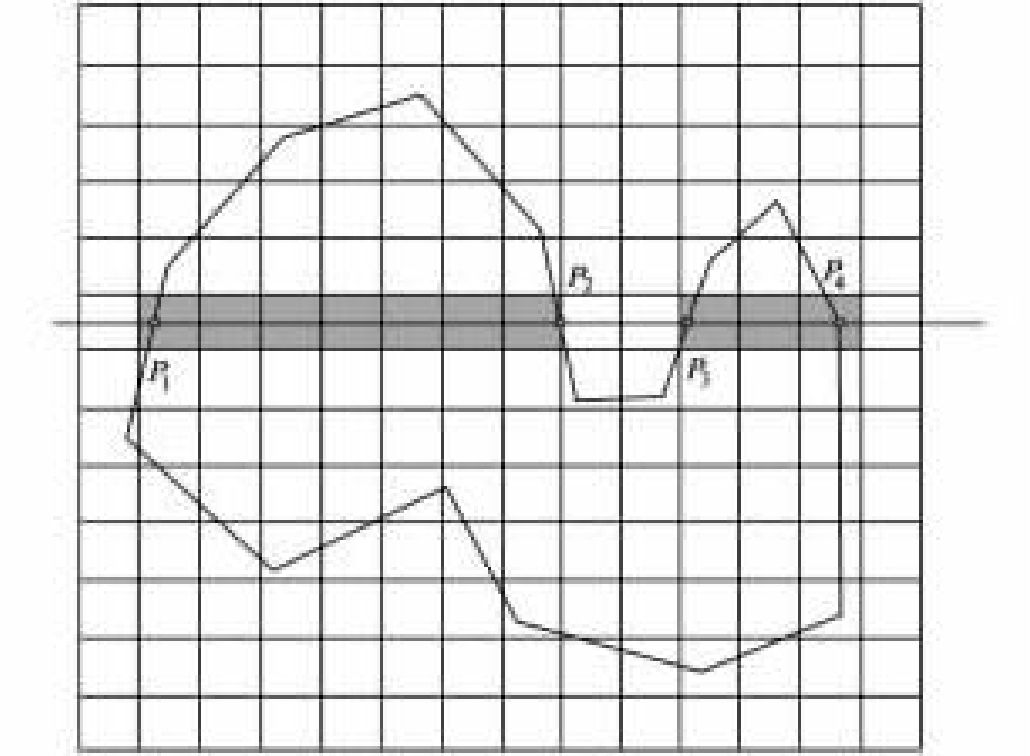
4.3 矢量与栅格数据的融合与转换

4.3.3 矢量数据与栅格数据结构的转换

1.矢量数据到栅格数据的结构转换

（3）面的转换

核心是将多边形矢量区域及内部转换成栅格数据



行填充法

计算栅格图形每行中心线与面边界（多边形）的交点，对交点进行排序、配对

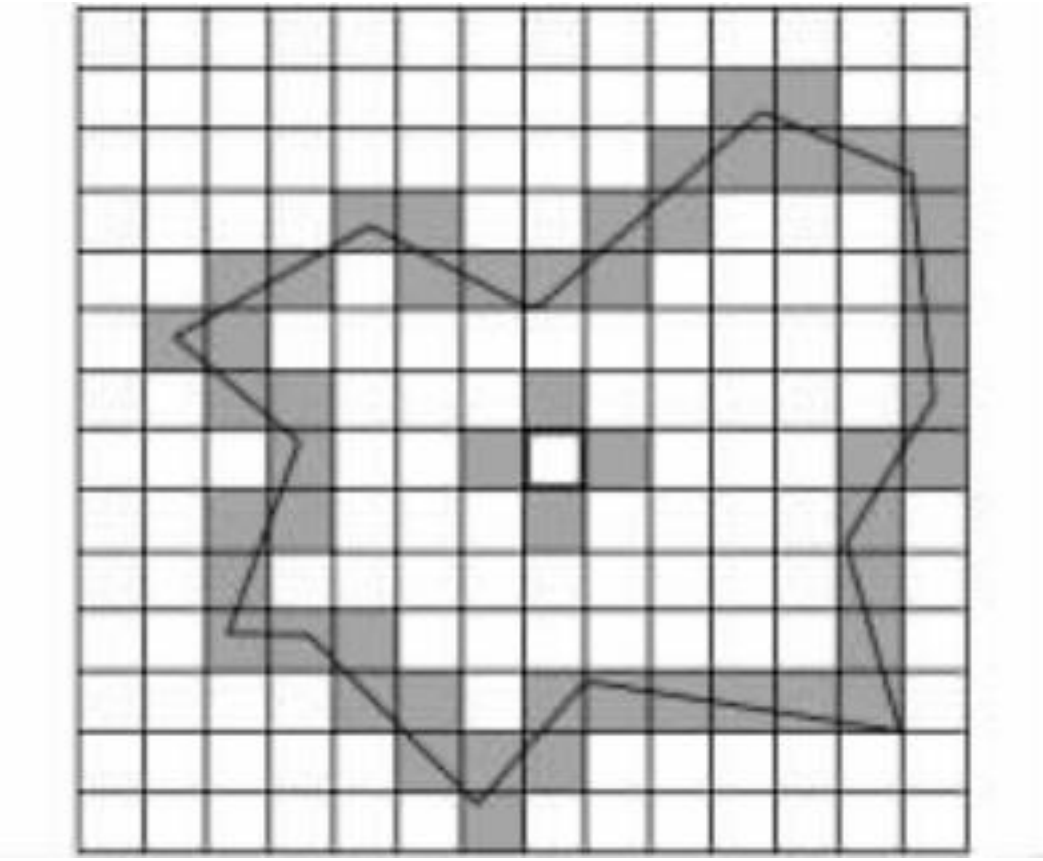
并用本面域的栅格属性值去填充每对交点（含）之间的栅格

4.3 矢量与栅格数据的融合与转换

4.3.3 矢量数据与栅格数据结构的转换

1.矢量数据到栅格数据的结构转换

点填充法



- 先将多边形边界转换为栅格数据
- 由多边形内部的点（种子）向其4个相邻方向点扩散
- 判断新加入的点是否在多边形边界上

4.3 矢量与栅格数据的融合与转换

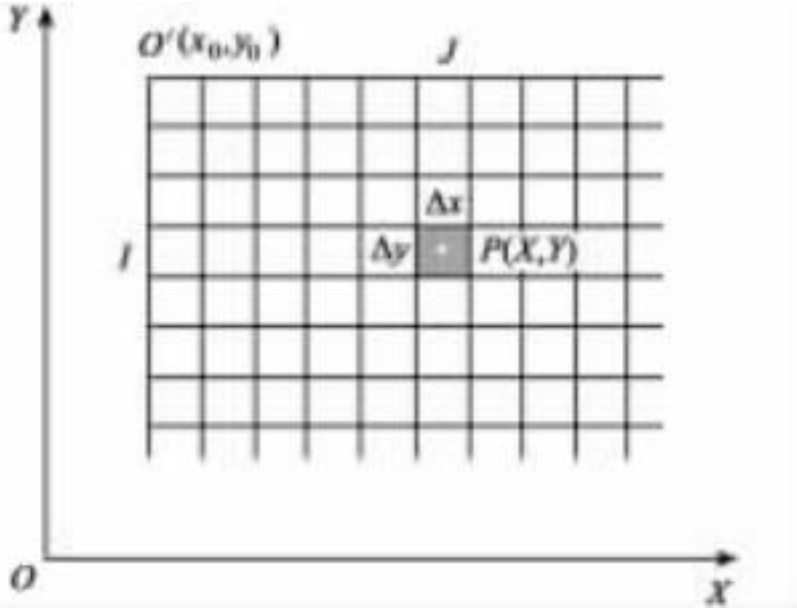
4.3.3 矢量数据与栅格数据结构的转换

2.栅格数据到矢量数据的结构转换

二值化处理： 在栅格图像灰度级的最大值和最小值之间选取一个阈值，当灰度级小于阈值时，取值为0；当灰度级大于阈值时，取值为1。

(1) 点的转换

设栅格点行，列号为*I,J*，相应栅格中心点的矢量坐标为*P (X,Y)*



$$\begin{cases} X = X_0 + (J - 0.5) \times \Delta X \\ Y = Y_0 - (I - 0.5) \times \Delta Y \end{cases}$$

4.3 矢量与栅格数据的融合与转换

4.3.3 矢量数据与栅格数据结构的转换

2.栅格数据到矢量数据的结构转换

(2) 线的转换

线状栅格图像往往具有一定的宽度，且往往宽窄不一，需要先细化处理，提取中轴线的基础上再矢量化，包括细化、跟踪和拓扑化过程。

细　化

通过3×3的像元窗口确定细化：凡是去掉后不会影响原栅格图像拓扑连通性的栅格都应该去掉。

3×3的像元阵列共有2^8即256种情况，经过旋转、去除相同情况，共有51种，其中，只有一部分是可以去掉中心点。

0	1	1
1		0
0	0	0

0	1	0
1		1
0	0	1

可去

1	0	1
0		0
0	1	0

0	1	0
0		0
0	1	0

不可去

4.3 矢量与栅格数据的融合与转换

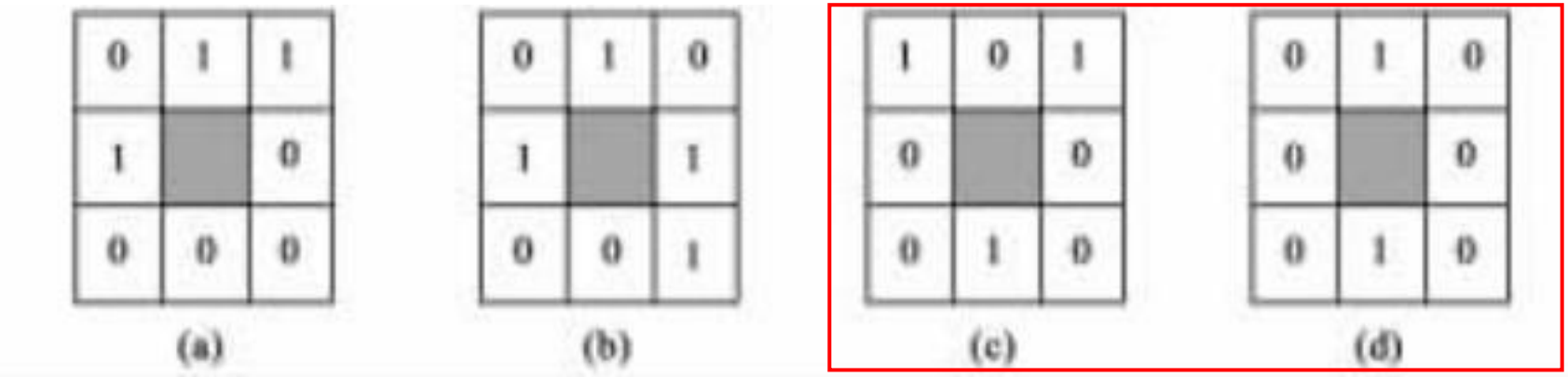
4.3.3 矢量数据与栅格数据结构的转换

0	1	1
1		0
0	0	0

0	1	0
1		1
0	0	1

1	0	1
0		0
0	1	0

0	1	0
0		0
0	1	0



可去

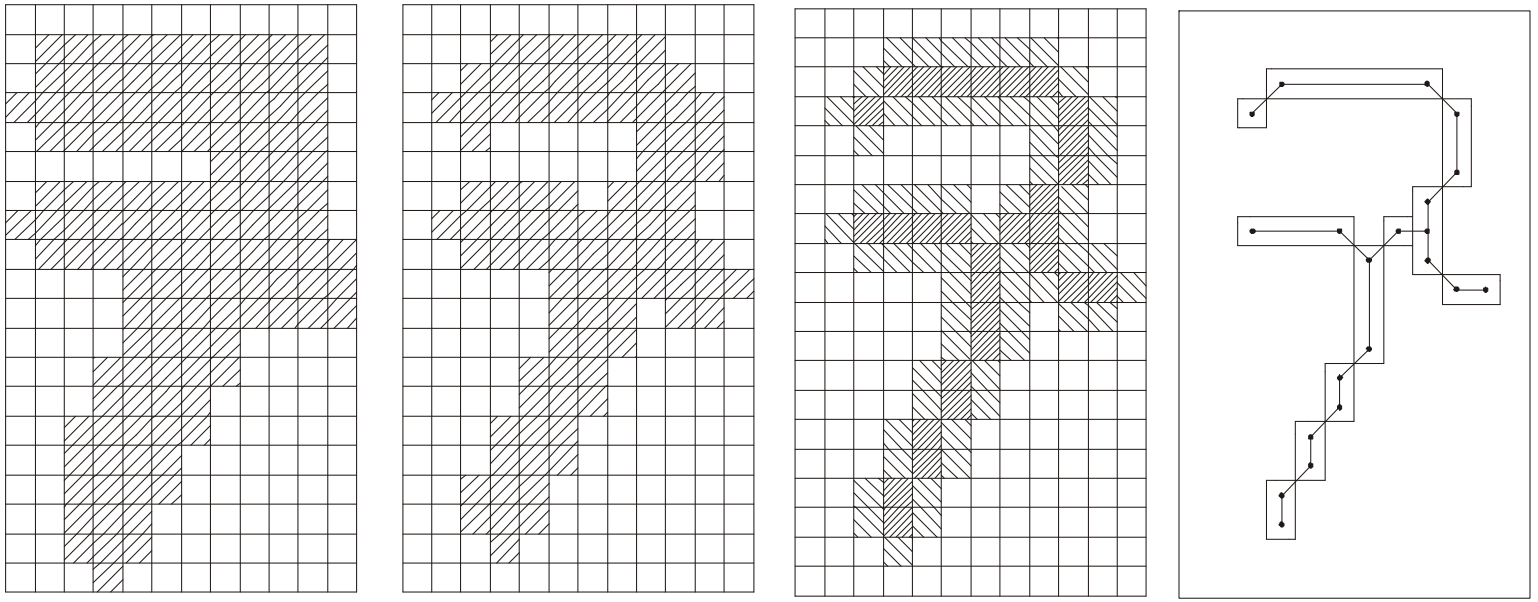
不可去

4.3 矢量与栅格数据的融合与转换

4.3.3 矢量数据与栅格数据结构的转换

2.栅格数据到矢量数据的结构转换

跟　踪



第一步：从上到下，从左到右搜索骨架线搜索起始点，并记录按点转换得到的（X,Y）坐标

第二步：按该点8邻域方向追踪下一点，若没有则本线跟踪结束；转下一条线跟踪；否则，记下该点的坐标

第三步：把搜索点移到新取的点上，转第二步

4.3 矢量与栅格数据的融合与转换

4.3.3 矢量数据与栅格数据结构的转换

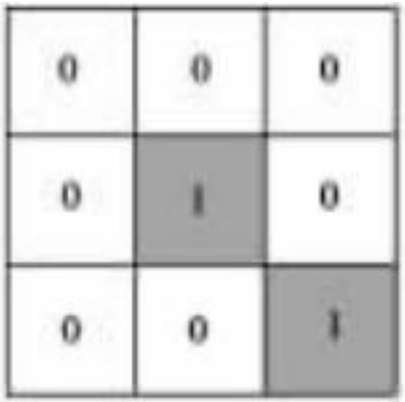
2.栅格数据到矢量数据的结构转换

拓扑化

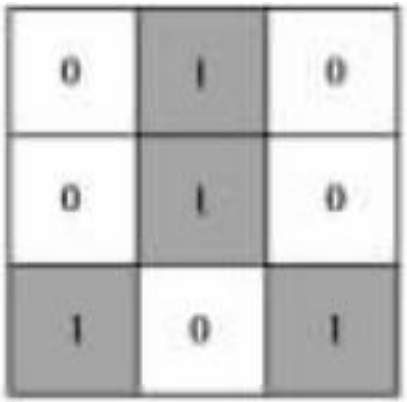
端点：8个邻域中，只有一个值为1的像元；

节点：8个领域中有3个或以上值为1的像元；

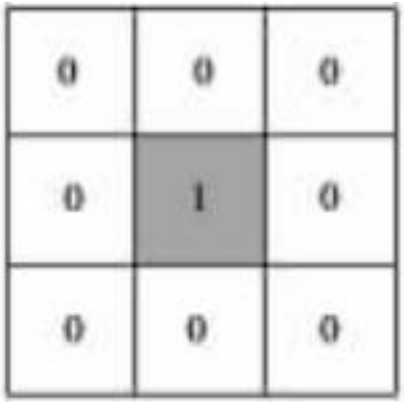
孤立点：8个邻域中没有值为1的像元



端点



节点



孤立点

4.3 矢量与栅格数据的融合与转换

4.3.3 矢量数据与栅格数据结构的转换

2.栅格数据到矢量数据的结构转换

(3)面的转换

面状栅格数据向矢量数据转换只需进行轮廓转换即可。

- 以面状栅格影像边缘（内沿或外沿）的一个像元为起点并记下按点转换方法所得的（X,Y）坐标
- 检测其8邻域，以找到（跟踪）下一个相邻的边缘像元作为新的起点，并记下其（X,Y）坐标
- 重复该跟踪过程直到结束

当前内容

4.1 矢量数据结构

4.2 栅格数据结构

4.3 矢量与栅格数据的融合与转换

4.4 镶嵌数据结构

4.5 多维数据结构

4.4 镶嵌数据结构

当前大纲

4.4.1 Voronoi数据结构

4.4.2 TIN数据结构

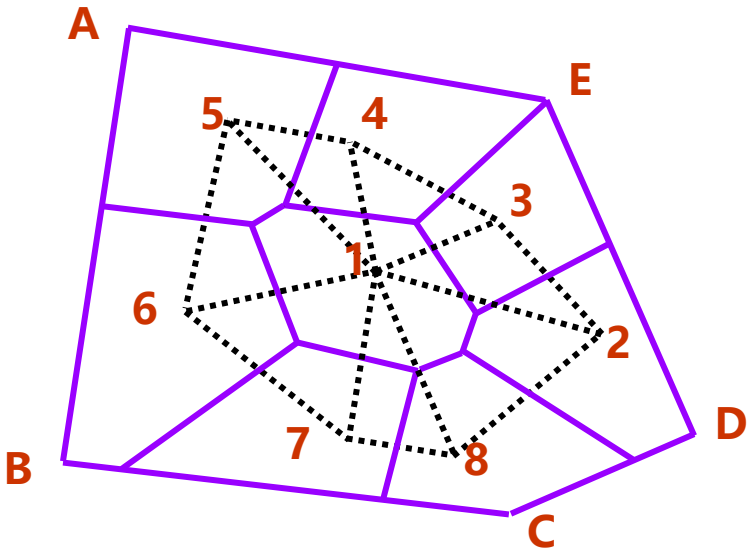
4.4 镶嵌数据结构

镶嵌数据结构概述

- 以正方形和矩形单元进行地理空间划分的规则镶嵌数据模型，采用栅格数据结构进行数据的组织；
- Voronoi多边形和TIN三角网采用专门的数据结构进行数据组织。

4.4 镶嵌数据结构

4.4.1 Voronoi数据结构



特征点数据

样点ID	样点坐标	样点属性值
1	X_1，y_1	A_1
2	X_2，y_2	A_2
……	……	……
7	X_7，y_7	A_7
8	X_8，y_8	A_8

Voronoi单元邻接关系表

Voronoi单元ID	相邻Voronoi单元号
1	2，3，4，5，6，7，8
2	3，1，8
……	……
7	8，1，6
8	2，1，7

Voronoi单元顶点组成表

Voronoi顶点 - ID	Voronoi顶点坐标	Voronoi顶点标识
v_1	X_1，y_1	
v_2	X_2，y_2	
…	……	
A	x_A，y_A	边界点
B	x_B，y_B	边界点
…	……	

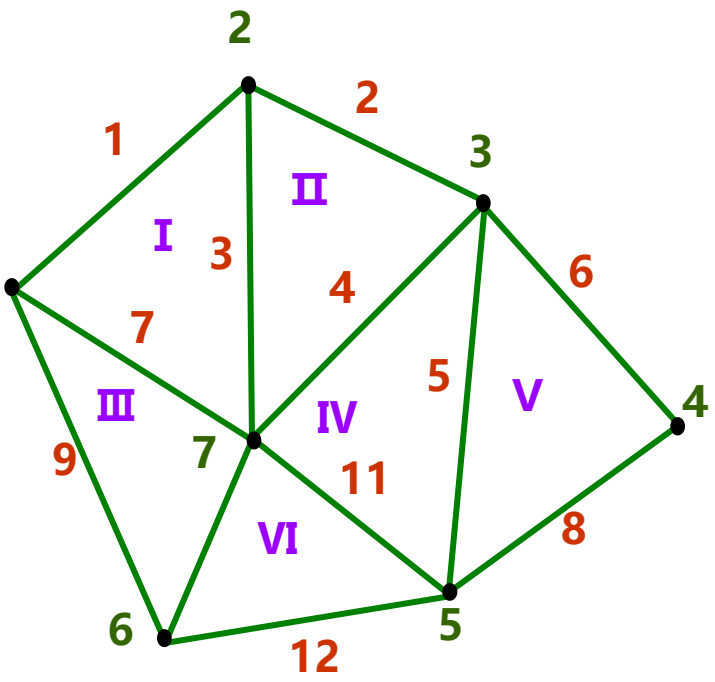
Voronoi顶点信息表

Voronoi单元ID	顶点 - ID
1	v_1,v_2,v_3,v_4,v_5，v_6,v_7
2	v_2,v_3,F,D,G
……	……

4.4 镶嵌数据结构

4.4.2 TIN数据结构

以三角形为基本对象



TIN网图

点文件

点ID	x	y	属性
1	x_1	y_1	z_1
2	x_2	y_2	z_2
…	…	…	…
7	x_7	y_7	z_7

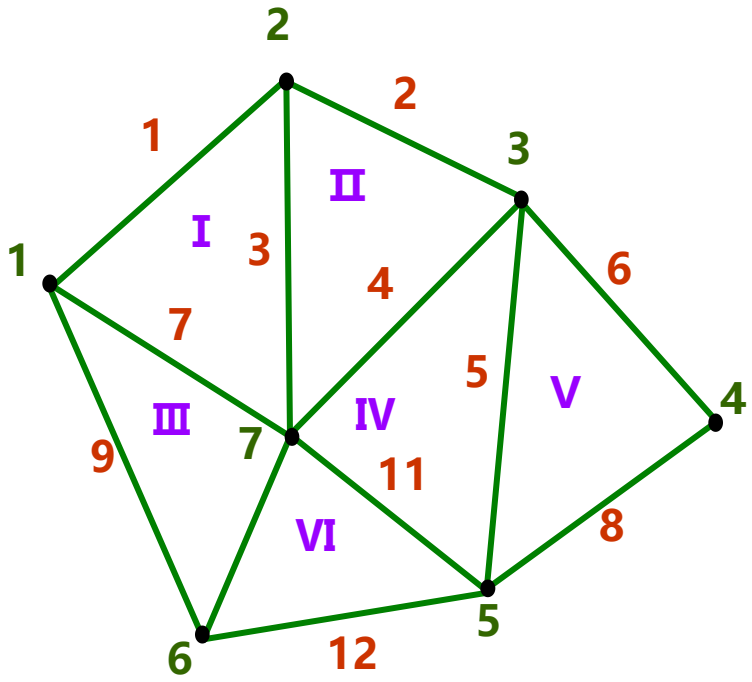
三角形拓扑文件

三角形ID	三角形顶点			邻接三角形		
	1	2	3	1	2	3
Ⅰ	1	2	7	Ⅱ	Ⅲ	×
Ⅱ	2	3	7	Ⅳ	Ⅰ	×
…	…	…	…	…	…	…
Ⅵ	6	7	5	Ⅳ	×	Ⅲ

4.4 镶嵌数据结构

4.4.2 TIN数据结构

以结点为基本对象



TIN网图

点文件

点ID	x	y	属性	指针
1	x_1	y_1	z_1	1
2	x_2	y_2	z_2	5
…	…	…	…	…
7	x_7	y_7	z_7	72

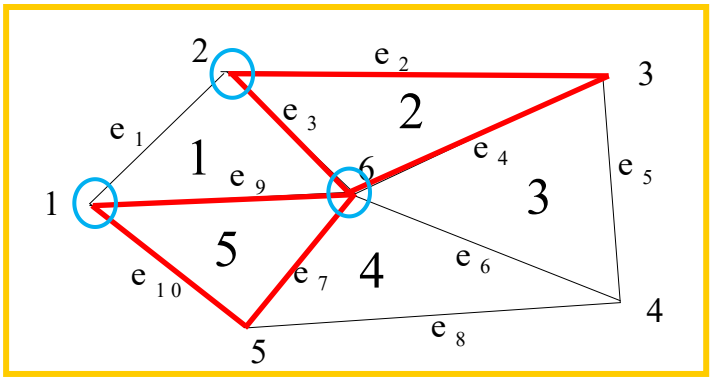
连接点文件

索引号	相连特征点
1	2
2	7
3	6
4	-
5	3
6	7
7	1
8	-
……	……

4.4 镶嵌数据结构

4.4.2 TIN数据结构

Polygon structure



Node

No.	X	Y	Z
1	.....		
2	.....		
3	.....		
4	.....		
...	.....		

Triangles

No.	V_1	V_2	V_3
1	1	2	6
2	2	3	6
3	3	4	6
4	4	5	6
...	.....		

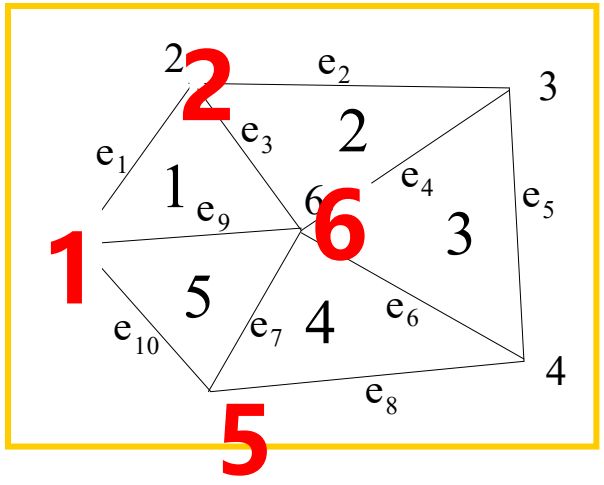
Neighbors

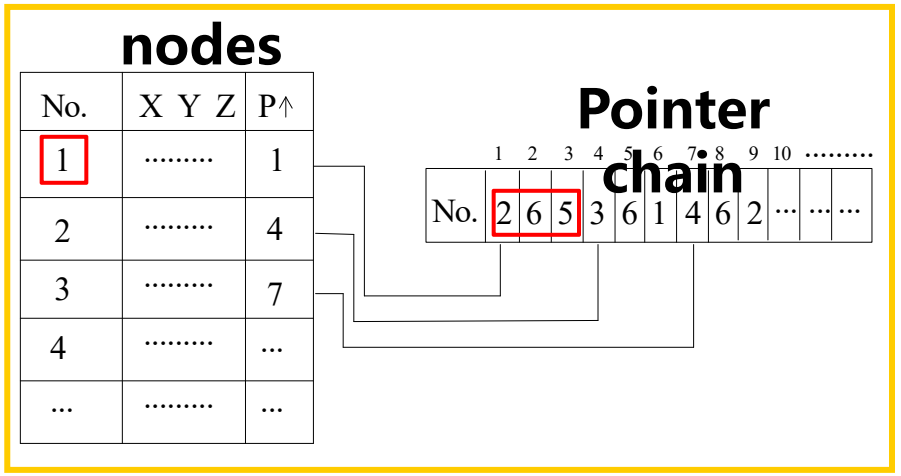
No.	T_1	T_2	T_3
1	2	5	-
2	3	1	-
3	4	2	-
4	5	3	-
...	.....		

4.4 镶嵌数据结构

4.4.2 TIN数据结构

Node structure

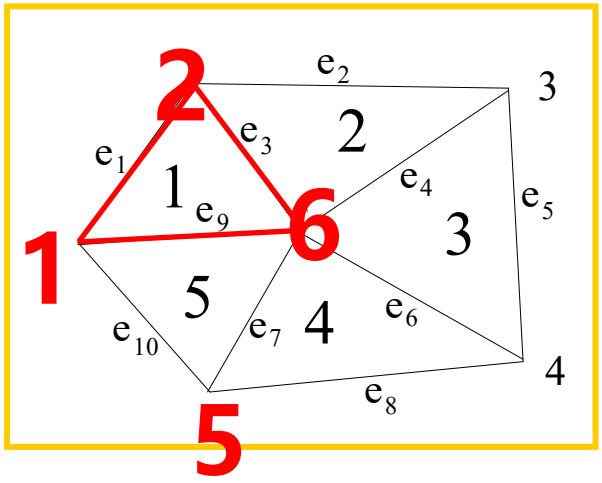




4.4 镶嵌数据结构

4.4.2 TIN数据结构

Node-Polygon structure



Node

No.	X Y Z	P↑
1	.....	1
2	.....	4
3	.....	7
4	.....	...
...	.....	...

Pointer chain

	1	2	3	4	5	6	7	8	9	10	.....
No.	2	6	5	3	6	1	4	6	2	...	... ...

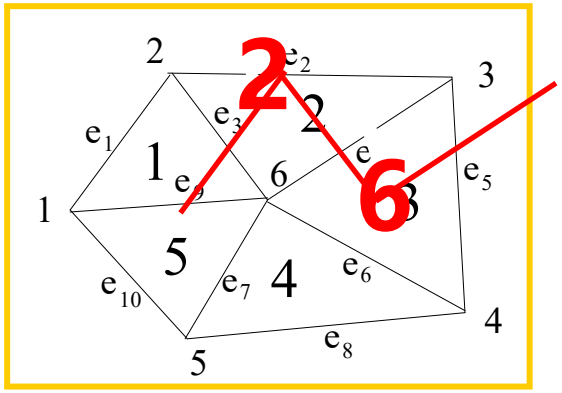
Triangles

No.	V_1	V_2	V_3
1	1	2	6
2	2	3	6
3	3	4	6
4	4	5	6
...	.....		

4.4 镶嵌数据结构

4.4.2 TIN数据结构

Arc structure



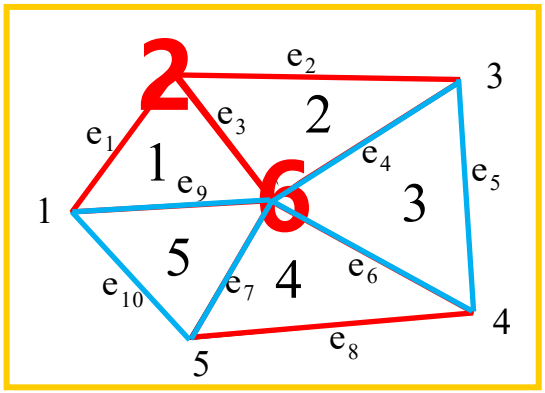
Arcs

No.	V_1	V_2	E_1	E_2
e_1	1	2	-	e_2
e_2	2	3	-	e_4
e_3	2	6	e_1	e_4
e_4	3	6	e_2	e_6
…	…	…	…	…

4.4 镶嵌数据结构

4.4.2 TIN数据结构

Arc-Polygon structure



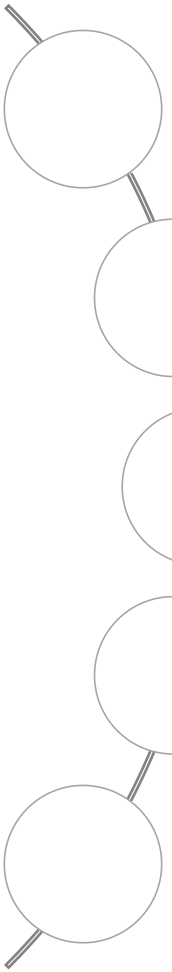
Arcs

No.	左△	右△	V_1	V_2
e_1	-	1	1	2
e_2	-	2	2	3
e_3	1	2	2	6
…	…	…	…	…

Triangles

No.	T_1	T_2	T_3
1	-	5	2
2	-	3	1
3	-	4	2
4	…	3	5

当前内容



4.1 矢量数据结构

4.2 栅格数据结构

4.3 矢量与栅格数据的融合与转换

4.4 镶嵌数据结构

4.5 多维数据结构

4.5 多维数据结构

当前大纲

4.5.1 多维数据的特征

4.5.2 多维数据结构

4.5 多维数据结构

4.5.1 多维数据的特征

现实世界呈现给我们的是一个三维甚至是多维的地理场景。

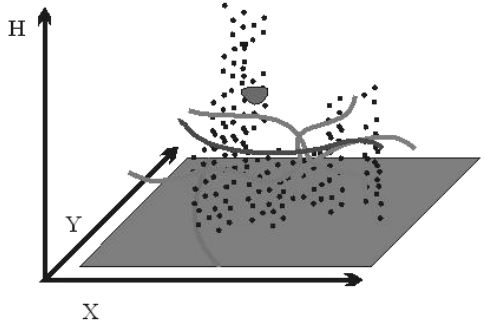
例如，严格意义上讲，温度不仅在二维空间分布上存在差异，随着海拔的升高，气温也会逐渐下降。这里就会涉及三个维度。但是，三个维度只能表示某个时刻任意位置的气温，而无法表示随时间的动态变化特征，这就需要添加第四个维度——时间维，从而形成了一个四维数据模型。

通常将两个维度以上的数据称之为多维数据，因此，三维及更多维度的数据均可以被看作是多维数据。

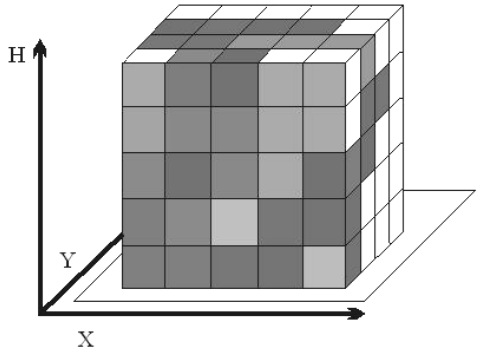
4.5 多维数据结构

4.5.1多维数据的特征

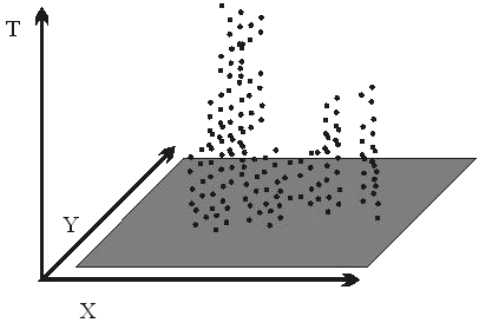
如果以Z轴分别表示高程(H)和时间(T)，则时空间可以用离散和连续两种方式表达。



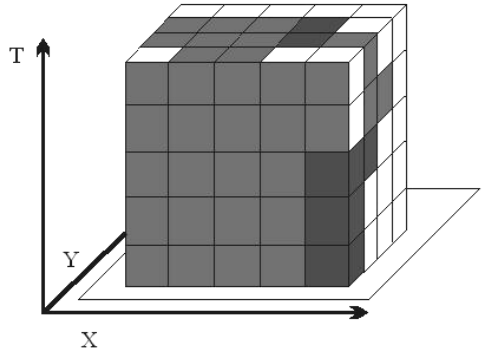
（a）三维离散数据



（b）三维连续数据



（c）时空离散数据



（d）时空连续数据

三维数据表达与时空维数据表达

4.5.2 多维数据结构

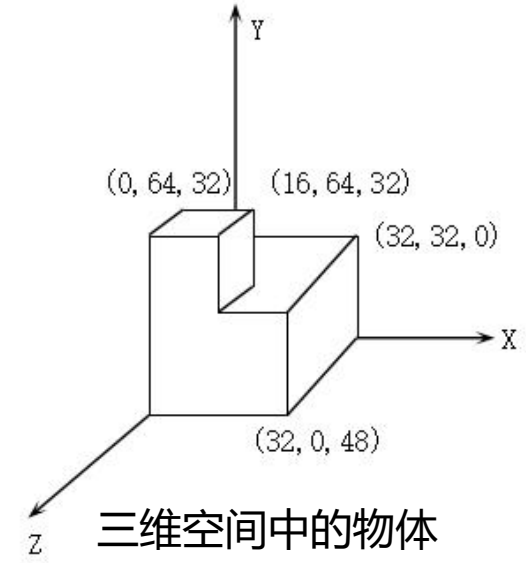
1.八叉树数据结构

八叉树数据结构可以看成是二维栅格数据中的四叉树在三维空间的推广。该数据结构是将所要表示的三维空间V按X、Y、Z三个方向从中间进行分割，把V分割成八个立方体；然后根据每个立方体中所含的目标来决定是否对各立方体继续进行八等分的划分，一直划分到每个立方体被一个目标所充满，或没有目标，或其大小已成为预先定义的不可再分的体素为止。

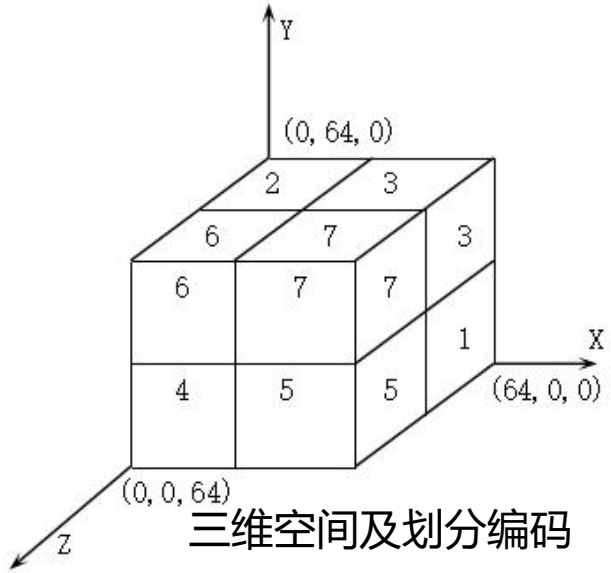
4.5 多维数据结构

4.5.2 多维数据结构

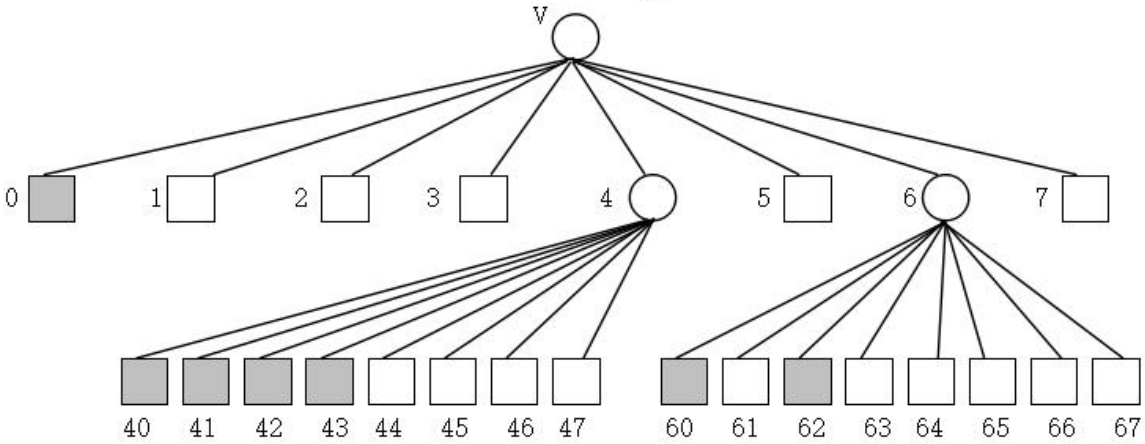
1.八叉树数据结构



三维空间中的物体



三维空间及划分编码

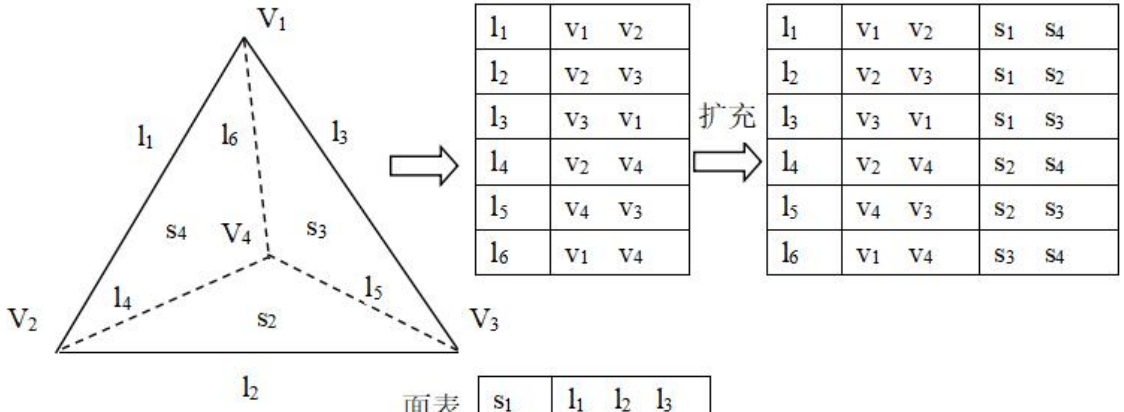


八叉树数据结构举例

4.5.2 多维数据结构

2.三维边界表示法

通过指定顶点位置、构成边的顶点以及构成面的边来表示三维物体的方法被称为三维边界表示法。



顶点表

V_1	$x_1\ y_1\ z_1$
V_2	$x_2\ y_2\ z_2$
V_3	$x_3\ y_3\ z_3$
V_4	$x_4\ y_4\ z_4$

边表

l_1	v_1	v_2
l_2	v_2	v_3
l_3	v_3	v_1
l_4	v_2	v_4
l_5	v_4	v_3
l_6	v_1	v_4

扩充

扩充后的边表

l_1	v_1	v_2	s_1	s_4
l_2	v_2	v_3	s_1	s_2
l_3	v_3	v_1	s_1	s_3
l_4	v_2	v_4	s_2	s_4
l_5	v_4	v_3	s_2	s_3
l_6	v_1	v_4	s_3	s_4

面表

s_1	l_1	l_2	l_3
s_2	l_2	l_4	l_5
s_3	l_5	l_3	l_6
s_4	l_1	l_4	l_6

专业术语与思考题

专业术语

实体数据结构、拓扑数据结构、网络数据结构、栅格数据结构、镶嵌数据结构、双重独立地图编码、DIME游程长度编码、四叉树数据结构、链码结构、影像金字塔、多维数据结构、八叉树数据结构、三维边界表示法

专业术语与思考题

复习思考题

一、思考题（基础部分）

1、总结矢量数据和栅格数据在结构表达方面的特色。

2、简述栅格数据压缩编码的几种方式和各自优缺点。

3、简述矢量数据编码的几种方式和各自优缺点。

4、常用的多维数据结构有哪些?

5、矢量和栅格数据的结构都有通用标准吗？请说明。

6、有人说矢量数据的实质还是栅格数据，你怎么理解这句话?

7、给出一张图，试写出图中的DIME数据文件和对其中多边形进行联接编辑的算法步骤，比较多边形联接编辑的异同。

专业术语与思考题

复习思考题

二、思考题（拓展部分）

1、试修改四叉树定义，使之适合存储数字化高程数据，并说明其存储结构的优缺点。

2、试述栅格数据结构和矢量数据结构在一个GIS平台中可能结合的几种方案。