关系的快速处理也可用于不规则四面体结构。

4) 不规则四面体适用于快速显示，每个四面体相对观察点可看作是相对独立的，可以使用计算机图形学中的方法分别对每个四面体进行处理，通过隐藏面消除、对象前后关系排序等手段方便地绘制三维物体。

2.2.3 混合三维数据结构

由于三维地质体的复杂性，常用两种以上的数据结构综合表达，如可以用矢量表示和栅格表示方法相结合的方式形成混合数据结构以表达地质体，常见的混合数据结构有：

(1) 三维规则格网和不规则三角网模型混合数据结构

首先建立三维空间的三维规则格网，然后将不规则三角网转换到三维规则格网上，即在不规则三角网和三维规则格网之间建立联系；显示地质体表面时常使用不规则三角网，而在剖面、等值线等计算时多使用三维规则格网（李青元等，2001）。

(2) 八叉树/四面体格网混合数据结构

八叉树结构简单、操作方便，不规则四面体格网能够保存原始数据、能精确表示目标和复杂的空间拓扑关系，八叉树/四面体格网混合数据结构同时利用八叉树和四面体格网的优点，一方面，通过八叉树结点可以直接引导与之相关的四面体；另一方面，通过八叉树的八个顶点与内部的特征点相连接来形成局部四面体格网。在混合数据结构中，八叉树作为整体描述，四面体作局部表达（李德仁，1997）。

(3) 矢量与栅格集成的面向对象混合数据模型

根据空间对象的不同几何特征（点、线、面、体）和不同的描述方法（矢量、栅格），将空间对象分解为不同的类，并以这些类作为设计数据模型的基础。地学空间数据抽象为13类空间对象，包括结点、弧段、线状地物、面状地物、数字表面模型、断面、影像、体状地物、数字立体模型、体元、柱状地物、复杂地物、空间地物，采用面向对象方法进行数据模型建立（龚健雅，1999）。

2.2.4 三维滑坡模型的数据结构分析

矢量、栅格以及混合数据结构三种数据表达中，矢量结构有更高的数据表达效率和精度，但对物体内部特性的描述比较弱；栅格数据结构能够表达对象的内部结构，可以表示出物体内部的不均匀性，几何计算方便，但数据压缩率比较低，数量大，对物体表达精度较低。矢/栅混合数据结构，采用了矢量和栅格数据结构的优点，同时考虑了对象的表面和内部结构的描述，而在数据量和表达精度上都可满足要求，容易形成对各类地质体的准确描述。

滑坡研究中既要描述滑坡地形特征，又要确定滑坡的滑动面形状位置、地下水位、地层岩性和地层分布、地质构造，这些因素可抽象为空间分布的不连续面，这些不连续面把斜坡切割为不规则块体，斜坡可视为这些不规则块体的组合；每个不规则块体的物理-力学性质可看作为相对均匀的，可通过块体样品测试结果的统计平均得到。如果能够确定滑坡体内部的各种不连续面，就可以

根据它们之间的交切、组合关系，确定出滑坡的空间结构，进而形成滑坡三维地质模型。因此可用两种方式表达滑坡，一是三维结构模型，即通过不连续面的空间组合表达；二是三维实体模型，利用体元形式描述不规则块体。这两种形式各有优缺点，三维结构模型具有较好的可视化，能够清楚表达各种地质界面之间的相互关系，尤其是对地质界面之间相互交切关系更能清楚表达，缺点在于它本质上是一个框架模型，所有数据均依附于不连续面（地质界面）上，空间分析不便；三维实体模型则具有较好的空间分析能力，但在块体之间关系的可视化方面不如结构模型，现在各种三维实体模型在表达内部结构时，只能采用剖面表示（*R. Marschallinger* , 2001, *R.Moore*, 2001）或剥层方法显示（李青元，1997，王东锐等，2001, 朱大培等，2001, 李培军等，2000）。三维滑坡模型最终要包括三维结构模型和三维实体模型，采用矢量/栅格混合的数据结构，用格网或不规则三角形网结构建立三维地形模型；用八叉树和四面体格网混合的数据结构形成三维地质模型。

滑坡研究中所涉及的各种数据可概括为以下对象类：点类（涉及空间位置的数据，如钻孔位置、控制点）、线类（线状地质特征，如滑坡周界等各种边界）、柱体类（如钻孔柱状图数据）、面类（各种面对象）和体类（滑坡地质体）等。三维地形模型由面对象加上遥感影像对象组成，三维地质模型由面对象或体对象组成。主要的数据类可类似下列定义得到：

点类：

```
class TPoint{
    int PointID;
    String PointAttr;
    float x;//坐标 x
    float y//坐标 y
    float z;//坐标 z
    int Color;//颜色
}
```

线类：

```
class TPolyline{
    int PolyID;//标识
    string PolyAttr;//属性描述
    float datax[];//坐标 x
    float datay[];//坐标 y
    float dataz[];//坐标 z
    int color;//颜色
 }
```

多边形类：

```
    class TPolygon{
        int PolygonID
    string attr;
    float datax;
```

```
    float datay;
    float dataz;
    int FillStyle;
    int FillColor;
 }
```

柱体类：

```
class TDrill{
    int DrillID;
    string DillAttr;
    TPoint pos;//位置
    Float Elevation;//孔口高程
    Float depth[];//各层深度
    string Stratum[] ;//各层层号
}
```

面类：

```
Class TFace{
    int FaceID;//面标识
    string FaceAttr;//面属性描述
    Uint samples;//格网行数
    Uint lines;//格网列数
    Uint xoffset, yoffset;//起始节点偏移量
    Int datatype;//数据类型
    TMapInfo mapinfo;//地理信息，用于确定地理编码
    Float data[][];//数据体
    TFace UpperFace;//顶层对象
    TFace LowerFace;//底层对象
    String upperStratum, LowerStratum;//面所分隔的地层属性
    Int FaceStyle;//样式
    Int FaceColor;//颜色
}
```

体类：

```
class TSolid{
    Int SolidID;
    string volattr;
    int NumofVolumes;//体元数
    TVolume *volume;//体元
}
```

体元：

```
    class TVolume{
    int VolID;
    string attr;
```

```
    TPoint vertex;//顶点
}
```

影像类，与面类对象结构相似，只是对多波段数据需要更多的存贮空间，并需要指定波段数据的存取方式，一般图像处理软件提供 BSQ（波段顺序）、BIL（颜色值顺序）、BIP（行顺序）三种方式：

```
class TImage:TFace{
    string InterLwave;//存贮方式
    int bands;//波段数
    float data[][][];//数据体
}
```

为了不同对象之间的准确配准，每个对象必须带有地理信息和投影信息，可用如下方式定义：

地理信息：

```
class TMapInfo{//地理信息
    string ProjectName;//投影名称，一般采用 Gauss-Kruger 投影
    Float xscl, yscl;//坐标缩放因子
    Float x_orign, y_orign;//起始点坐标
    Float x_cell_Size, y_cell_size;//行向/样向的单元尺寸
    String Uints;//坐标单位，一般采用米
    TProject project;//投影信息
}
```

投影信息：

```
Class TProject{//投影信息
    Int projectiontype;//投影类型
    Float EllipsoidA, EllipsoidB;//椭球参数
    Flaot latitudeofOrigin, LongitudeofMeridian;//起始纬度；中央经线
    Flaot FalseEasting, FalseNorthing;//坐标偏移量
    Float SacleFactor;//坐标比例因子
    String units;//坐标单位
}
```

目前已经完成的三维滑坡模型是基于栅格数据结构建立的三维结构模型。采用面向对象技术，建立各种地质界面（不连续面）的栅格模型，把每个地质界面作为一个对象，整个三维模型由这些面对象组成。

2.3 三维滑坡模型中的空间插值技术

实际滑坡调查得到的数据是空间上分布不均匀的散点数据，为得到滑坡完整的空间结构，就需要进行空间数据插值。空间数据插值就是给定一组已知的数据，从这些数据中找出一个关系式，使该关系式最佳逼近已知数据，并可根据该关系式推求区域范围内未知点的值；也就是通过已知的数据，推求未知点数据，从而在区域空间内形成一个连续特征的变化。

空间插值按数据点与插值结果之间的关系，可分为纯插值和曲面拟合两类