第 3 章　基于等值面的三维空间数据场重构与显示

3.1　基于等值面的三维空间数据场重构技术

三维空间数据场的重构与显示包括基于体绘制和基于面绘制的两类可视化算法，基于体绘制算法是不构造中间几何图元而直接在屏幕上进行三维空间数据场的显示与绘制，基于面的可视化算法在三维空间数据场中首先构造出中间几何图元如三角形等，再由传统的计算机图形学技术实现面的绘制，基于等值面的三维空间数据场重构与显示的算法和技术有许多种，这主要包括移动立方体（Marching Cubes）、移动四面体（Marching Tetrahedra）和由二维轮廓线构建外表面等，而移动立方体是较为成熟和应用广泛的一种技术。移动立方体算法在 1987 年就由美国通用电气公司的研发机构成员 William E. Lorensen，Harvey E. Cline 提出，目前该算法得到了广泛的应用和优化。

Marching Cubes 算法本质是从一个三维的数据场中提取出一个等值面，所以也被称为“等值面提取”算法。由于这一方法原理比较简单，在程序编制上易于实现，目前已经在比较广泛的领域得到了应用。这章将重点介绍 Marching Cubes 算法的原理、分析算法的思路及算法程序的设计，并且比较详细地介绍了三维重构和显示中能够达到高效显示的 OpenGL 技术，列举出了算法实现中得到的一些对比性数据，结合三维绘制软件包的开发，介绍了 Marching Cubes 算法的实际应用，对该算法的研究和详细介绍是基于 Delphi 开发环境下的，结合 OpenGL 技术显示三维物体，并给出了算法的实验比较结果，同时还将较复杂的二维和三维图像预处理对基于面的三维重构的影响和作用进行了较深入的探讨。

3.1.1　Marching Cubes 算法的基本概念

在 Marching Cubes 算法中，假定原始数据是离散的三维空间规则数据场，在医院和工业无损测量的 CT 和 MRI 等设备产生的图像就属于这一类型。例如对于一个标准的医学图像的体数据集，它往往是由一系列的二维断层切片数据构成的，而每张切片都有其固定的分辨率。假设一体数据集有 100 张切片，而每张切片的分辨率为 512×512，那么它可以被认为是一个连续函数 $f(x,y,z)$在 x,y,z 三个方向上按一定的间隔分别采样了 512，512，100 次所得到的。如果对于这个数据集，需要构造出相同属性的几何物体出来，如在 $512\times512\times100$ 的组成的头脑切片扫描的数据中需要构造出灰度值等于 200～255 范围的头骨的三维图像，就需要将灰度值在这个范围的等值面求出来。所谓等值面实际上是指空间中的一张曲面，在该曲面上函数 $f(x,y,z)$的值是恒定的或在有限变化范围的。等值面提取算法的核心就是要从给定的采样点中找出等值面来，由采样点得出连续函数 $f(x,y,z)$，然后由这个 $f(x,y,z)$函数和某一给定的域值来构造等值面，这就是通常的显式的等值面提取算法，有时也是不可能实现的，由于重构和重采样所带来的误差比较大，重构出的等值面不能精度地反映原形。而采用 Marching Cubes 算法使用隐式的等值面提取方法，不直接计算函数 $f(x,y,z)$，而是直接从体数据中获取等值面的信息。

其算法要求用户提供想提取出来的物质的密度值，然后根据物体的体数据的分布信息，提取出等值面并用三角面片表示之，用软件方法或图形硬件提供的面绘制功能绘制出等值面，如用 OpenGL 提供的三角形链表而显示出表达等值面的三角形面簇。

3.1.2 Marching Cubes 算法介绍

（1）确定包含等值面的体元。

设想被重构对象所在的三维空间是由一个个体元组成。将二维断层图像序列按图像排列的顺序分层读入内存，其中每相邻的两层数据构造多个立方体形状的体元，如图 3-1 所示。由于每个立方体有 8 个顶点，故我们对这 8 个顶点进行采样，可以得到它们在对应断层切片上的灰度值。如果定义等值面的值为一个定值 G，通过判断顶点灰度值的大小就可以得知物体表面与哪些体元相交，或者说等值面穿过了哪些体元。

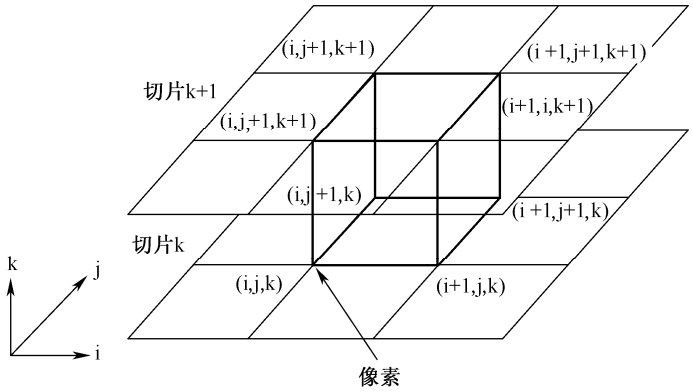

图 3-1 一个立方体元的构造与标注

显然，由于每个体元八个顶点，每个顶点有两种状态（要么在表面内，要么在表面外），这样就有 $2^8 = 256$ 种情况。通过列举这 256 种情况，建立一张索引表，表中记录了各种情况中相交边的所在，通过一个字节的索引值就可以在该索引表中查询。立方体是由八个顶点构成的，于是就有 256 种可能的顶点位置状态。考虑到通过一定条件可以对单元结构进行合并，这 256 种复杂的情况可以被大大地简化。这些简化的条件包括：

- 绕轴旋转一定角度。
- 图形相对轴面镜面对称。
- 顶点的位置状态取反。

考虑到这些条件，利用立方体的两种不同的对称性可以将这 256 种情况简化为 15 种，如图 3-2 所示。根据这 15 种情况，可以构造出一个查找表，记录所有情况下的等值面连接情况。这样就更加容易建立起多边形集合，用以合适地去近似表达物体表面。图 3-2 列出了这 15 种情况的对应多边形构成。球点表示在物体外面的点，箭头表示法线方向。

最简单的情况就是第 0 种情况，这种情况发生于整个立方体都被表面包围或者整体都在表面之外。情况 1 表示的是立方体中只有一个点包含在表面内，这种情况用一个三角形片就可以近似表示与它相交的曲面。通过旋转和求补可容易地得到所有的 256 种情况下三角形片的组成。空间内的每个虚拟立方体利用其 8 个顶点在表面的包含情况获得一个索引值，通过该索引

值可以在索引表中找到与其对应三角形片组成。通过对空间里每一个虚拟立方体内的三角形构造完成后，整体就将形成三维表面的近似表示。

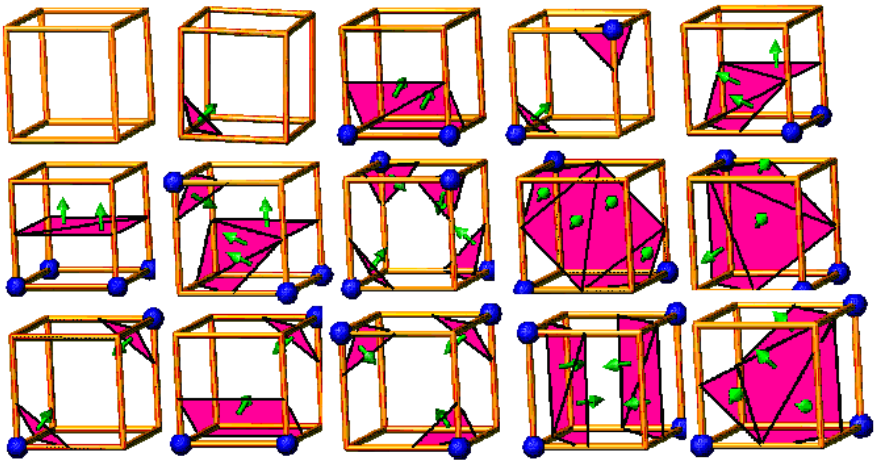

图 3-2　构成等值面的立方体内可能的三角形分布情况

（2）求等值面与体元边界的交点。

每个立方体或者网格体的八个顶点分布和索引表如图 3-3 所示，其中大的黑体数字 1，2，3…8 表示顶点（这点的灰度值），小的灰色体数字 1，2，3…8 表示各顶点的连接边，通过该索引值可以在索引表中找到等值面与体元边界的交点并找出与其对应三角形片的组成。例如，只有点 3 被表面截掉，而其他顶点都位于曲面另一端，这样我们就需要一个三角形面来近似表示与边 2、边 3 和边 11 相交的曲面。这样的三角形分布将存在 256 种的可能情况，而且必须对每一种情况都构成相应的片面从而使得相邻的网格内的平面可以正确地相连。

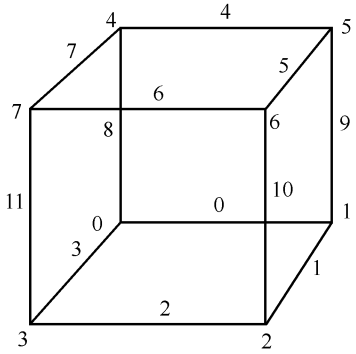

图 3-3　网格体的八个顶点分布和索引表的概念

依此推理而得到顶点表 edgeTable，这张表中每一项描述了被表面截到的边。8 比特索引值的每一位对应于网格体的八个顶点，根据网格点的值 grid.val[i]而确定网格点的索引表 cubeindex，由索引表而得到边表 edgeTable[cubeindex]进而知道等值面与几个边相交，由相交边而组成三角形，许多表达等值面的三角形簇组成整个等值曲面。

```
cubeindex = 0;
if (grid.val[0] < isolevel) cubeindex |= 1;
```