

面向网格简化的数据结构快速生成方法

陶立欣, 杨建生

(上海大学理学院, 上海 200444)

摘要: 针对网格简化算法需要的显式和完整拓扑关系, 筛选出有效的数据结构, 结合 MC(MT)算法计算过程的特点, 设计六体素法算法。该算法能在 MC(MT)算法抽取等值面的同时完成简化算法所需的数据结构, 减少了简化算法前数据结构预处理所耗费的时间和空间。
关键词: 数据结构生成; MC(MT)算法; 网格简化

Fast Generation of Data Structure for Mesh Simplification

TAO Li-xin, YANG Jian-sheng

(Department of Sciences, Shanghai University, Shanghai 200444)

【Abstract】 This paper propose a novel method based on the course of MC(MT) algorithm computation to rebuild an apparent and complete topological structure necessary for mesh simplification to reduce the time and memory of the data structure pretreatment. The method called as six-cube method, rebuilds the appropriate data structure while MC(MT) algorithm extracts the iso-surface.

【Key words】 data structure building; MC(MT) algorithm; mesh simplification

1 概述

医学上利用物体的二维CT切片图像数据重建其三维结构。在医学三维图像重建的方法中, 移动立方体(Marching Cube, MC)算法^[1]及其基础上发展起来的移动四面体(Marching Tetrahedra, MT)算法^[1]是表面绘制算法中的经典算法之一。但其缺点在于计算较大的三维表面时产生的三角面片过多(MT算法产生的三角面片较MC算法更多), 即使在高端硬件配置的计算机中绘制三维表面也会耗费大量时间, 无法做到人机实时交互。因此, 对MC(MT)算法产生的三角面片集进行网格简化^[2]是必须的工作。

本文所提到的网格是指三角面片。三角面片集在简化过程中不断调整, 能对其拓扑关系进行实时更新是进一步简化的必要条件。因此, 满足网格简化所要求拓扑关系的数据结构必须是显式和完整的^[3]。而MC(MT)算法直接产生的三角面片集合不能达到这一要求, 要经过中间步骤处理后才能为网格简化算法所用。

本文提出的算法对计算链进行了改进, 在 MC(MT)算法计算的同时完成中间步骤要完成的功能, 减少占用的时间和空间。

2 数据结构的选取

MC(MT)算法的基本思想是计算等值面与体素集合的交点, 并生成基于这交点集合的三角面片集合来拟合等值面。如果 MC(MT)算法产生的三角面片有 n 个, 每个三角面片由 3 个不同三维空间点组成, 则假设这 n 个三角面片集合对应的 $3n$ 个点集合为

$$P_n = \{P_i \mid i = 0, 1, \dots, 3n-1, P_i \in R^3\}$$

其中, P_i 不一定两两互不相同。若 P_n 中所有不同的元素为 $P_{i_1}, P_{i_2}, \dots, P_{i_k}$, 则定义:

$$\tilde{P}_n = \{P_{i_1}, P_{i_2}, \dots, P_{i_k}\}$$

由此由 MC(MT)算法产生的三角面片集可以定义为

$$\Delta(P_n) = \{(P_{3i}, P_{3i+1}, P_{3i+2}) \mid i = 0, 1, \dots, n-1\}$$

假设 f 是 $I \rightarrow \tilde{I}$ 的映射, 其中, $I = \{0, 1, 2, \dots, 3n-1\}$, $\tilde{I} = \{i_1, i_2, \dots, i_k\}$, 由 f 可以产生新的三角面片集合, 即

$$\Delta_f(P_n) = \{(P_{f(3i)}, P_{f(3i+1)}, P_{f(3i+2)}) \mid i = 0, 1, \dots, n-1\}$$

下面将构造适当的 f , 以便产生新的三角面片集合 $\Delta_f(P_n)$, 由此代替原有的三角面片集合 $\Delta(P_n)$, 从而优化 MC(MT)算法产生的数据, 使之适合于网格简化算法。

MC(MT)算法对一组CT二维图像集合作用, 两张二维图像可以组成几何意义上的一层体素, 多层体素可以组成几何意义上的一个大立方体数据, 此大立方体数据可以记为 $C\{(i, j, k) \mid i, j, k \text{ 为大于或等于 } 0 \text{ 的整数, 且 } i < n, j < m, k < l\}$, 第 i 层记为 $C(i) = \{(x, y, i) \mid (x, y, i) \in C\}$, 记 $C(i, j, k)$ 为一个具有 x 列、 y 行、 z 层的大立方体数据。下文中用 $C_{i,j,k}$ 表示 C 中体素 (i, j, k) 。图 1 深色部分为体素 $C_{5,0,0}$ 。

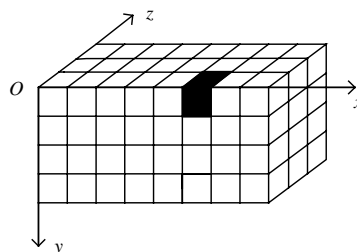


图 1 所有体素组成的大立方体数据

用三角面片集合来拟合一张曲面, 这些三角面片在几何拓扑结构上往往两两相连, 即在组成三角面片 A 的 3 个三维

作者简介: 陶立欣(1981 -), 男, 硕士, 主研方向: 医学图像处理, 3D 模型重建, 3D 网格简化; 杨建生, 副教授、博士
收稿日期: 2007-05-28 **E-mail:** taolix@sohu.com

点中存在 2 个点与组成三角面片 B 的 3 个三维点中 2 个点相同。例如 $(P_{3k}, P_{3k+1}, P_{3k+2}) \in \Delta(P_n)$ 和 $(P_{3k+3}, P_{3k+4}, P_{3k+5}) \in \Delta(P_n)$ ，有 $P_{3k} = P_{3k+3}$, $P_{3k+1} = P_{3k+4}$ ，如图 2 所示。可称 $P_{3k} = P_{3k+3}$ 和 $P_{3k+1} = P_{3k+4}$ 为重复点。若在 P_n 中减少 a 个相同的重复点 $P_{i_1} = P_{i_2} = \dots = P_{i_a}$ 的个数，最终保留 P_{i_1} ，则这个过程称为删除重复点。那么可以定义 $f(j)$ 来表示删除重复点，即

$$f(j) = \begin{cases} j & j \neq i_t, t=1,2,\dots,a \\ i_1 & j = i_t, t=1,2,\dots,a \end{cases}$$

若将 P_n 中所有的重复点删除，则 P_n 退化为 \tilde{P}_n 。

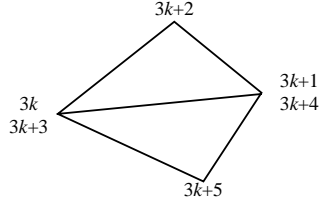


图 2 两个相邻三角面片中的重复点

本文采用的数据结构必须包含集合 P_n 和 (P_n) 的信息。最简单的一种结构就是使用 P_n 和 (P_n) 。记 P_n 和 (P_n) 组成的数据结构为 A 。此结构类似于 STL 文件格式^[4]，缺乏网格简化算法所需的显式和完整的拓扑关系。因此，该类结构不适合简化算法的计算机实现。

修改 $(P_{3k}, P_{3k+1}, P_{3k+2}) \in \Delta(P_n)$ 的充要条件是修改任意一个组成三角面片 $(P_{3k}, P_{3k+1}, P_{3k+2}) \in \Delta(P_n)$ 的三维点 P' 。若 P' 是重复点，则存在 $P'' = P'$ ， P'' 是组成三角面片 $(P_{3s}, P_{3s+1}, P_{3s+2}) \in \Delta(P_n)$ 的三维点。那么 $(P_{3s}, P_{3s+1}, P_{3s+2}) \in \Delta(P_n)$ 同样需要修改。实际上，在类似结构 A 的数据结构中，对任意 $P \in P_n$ 都不知道点 P 是否为重复点，因此，修改包含点 P 的三角面片时必须搜索整个集合 P_n 来确定点 P 是否为重复点。

可以采用如下 2 个集合来组成数据结构：

$$\tilde{P}_n = \{P_{i_1}, P_{i_2}, \dots, P_{i_k}\}$$

$$\Delta_f(P_n) = \{(P_{f(3i)}, P_{f(3i+1)}, P_{f(3i+2)}) \mid i=0,1,\dots,n-1\}$$

记 \tilde{P}_n 和 $\Delta_f(P_n)$ 组成的数据结构为 B 。采用非重复点集合 \tilde{P}_n 和在其基础上构建的三角面片集合 $\Delta_f(P_n)$ ，这样做的优点在于修改 $\Delta_f(P_n)$ 时，只需要修改 \tilde{P}_n 中的点 P 一次后再修改 $f(P)$ 即可，不仅提高了效率，而且给简化算法的计算机实现带来了方便。此类结构可以满足网格简化过程中的拓扑关系实时更新所需的复杂索引和面片、点信息的动态删除。这是研究和实现简化算法时常用的数据结构。

3 方法设计与分析

下面假设由 MC(MT) 算法产生的体素 $C_{i,j,k}$ 与其等值面的交集集合记为 $P_{i,j,k}$ ，根据 MC(MT) 算法有

$$\bigcup_{k=0}^l \bigcup_{j=0}^m \bigcup_{i=0}^n P_{i,j,k} = P_n$$

当 MC(MT) 算法计算完成 $P_{i,j,k}$ 后，根据六体素法得到映射 f ，从而得到非重复点集合 $\tilde{P}_{i,j,k}$ ， $\tilde{P}_{i,j,k} = f(P_{i,j,k})$ 。则有

$$\bigcup_{k=0}^l \bigcup_{j=0}^m \bigcup_{i=0}^n \tilde{P}_{i,j,k} = \tilde{P}_n$$

六体素法的作用是在 MC(MT) 算法计算 P_n 和 (P_n) 的同时，根据 (P_n) 和 (P_n) 构造 \tilde{P}_n 和 $\Delta_f(P_n)$ ，不必等 P_n 和 (P_n) 计算完毕再构造简化算法需要的数据结构。

在六体素法中，构造非重复点集合 \tilde{P}_n 和对应的三角面片集合 $\Delta_f(P_n)$ 是从空集开始的。

3.1 方法设计

理论 1 设 $C_{i,j,k}$ 为当前体素，则 $P_{i,j,k}$ 只须比对 $P_{i-1,j,k}$ ， $P_{i,j-1,k}$ ， $P_{i,j,k-1}$ 就可以确定 $P_{i,j,k}$ 中的重复点。

根据 MC(MT) 算法的假设， $P_{i,j,k}$ 只可能在 $C_{i,j,k}$ 的 12 条棱上(在 MT 算法中还可能在 $C_{i,j,k}$ 的 6 个面上的分割线上) $P_{i,j,k}$ 都在 $C_{i,j,k}$ 的 6 个面上出现。因此，只须将准备压入 \tilde{P}_n 的 $P_{i,j,k}$ 与 $A = \{P_{i-1,j,k}, P_{i,j-1,k}, P_{i,j,k-1}, P_{i+1,j,k}, P_{i,j+1,k}, P_{i,j,k+1}\}$ 比较即可去除 $P_{i,j,k}$ 中的重复点。

MC(MT) 算法的计算过程按一定的顺序实现，例如沿 x 方向、 y 方向、 z 方向的顺序来完成 $C(x,y,z)$ 的等值面抽取。此时与 $C_{i,j,k}$ 几何上面或棱相邻接的体素只有 $C_{i-1,j-1,k-1}$ ， $C_{i-1,j-1,k}$ ， $C_{i-1,j-1,k+1}$ ， $C_{i-1,j,k-1}$ ， $C_{i-1,j,k}$ ， $C_{i-1,j,k+1}$ ， $C_{i,j-1,k-1}$ ， $C_{i,j-1,k}$ ， $C_{i,j-1,k+1}$ ， $C_{i,j,k-1}$ ， $C_{i,j,k}$ ， $C_{i,j,k+1}$ 这 13 个体素存在。记这 13 个体素与等值面的交集集合为 B 。则有 $A \cap B = \{P_{i-1,j,k}, P_{i,j-1,k}, P_{i,j,k-1}\}$ 。所以，要确定 $P_{i,j,k}$ 中的重复点只须比对 $P_{i-1,j,k}$ ， $P_{i,j-1,k}$ ， $P_{i,j,k-1}$ 即可。

将 $P_{i,j,k}$ 压入 \tilde{P}_n 时，可以使用以下 2 种方法去除重复点：

- (1) 保留 $P \in P_{i,j,k}$ 而删除 $P' \in \tilde{P}_n$ ，其中 $P = P'$ ；
- (2) 删除 $P \in P_{i,j,k}$ 而保留 $P' \in \tilde{P}_n$ ，其中 $P = P'$ 。

若使用方法(1)，则在将点 P 和三角面片压入 \tilde{P}_n 和 $\Delta_f(P_n)$ 的同时必须调整 \tilde{P}_n 和 $\Delta_f(P_n)$ ，开销较大；而使用方法(2)，则 \tilde{P}_n 和 $\Delta_f(P_n)$ 都不需要再调整。方法(2)是较好的选择。但方法(2)也有其固有的缺点，即只根据理论 1 查找重复点有可能遗漏。

理论 2(六体素法) 设 $C_{i,j,k}$ 为当前体素，若以删除 $P \in P_{i,j,k}$ 而保留 $P' \in \tilde{P}_n$ ，其中 $P = P'$ 的方法来去除重复点，则 $P_{i,j,k}$ 须比对 $P_{i-1,j,k}$ ， $P_{i,j-1,k}$ ， $P_{i,j,k-1}$ ， $P_{i-1,j-1,k}$ ， $P_{i,j-1,k-1}$ ， $P_{i-1,j,k-1}$ 才可以确定和删除 $P_{i,j,k}$ 中的重复点。

设当前体素为 $C_{i-1,j,k}$ ，在 $P_{i-1,j,k}$ 压入 \tilde{P}_n 之前，根据理论 1， $P_{i-1,j,k}$ 须比对 $A = P_{i-2,j,k} \cup P_{i-1,j-1,k} \cup P_{i-1,j,k-1}$ ，若存在 $P' \in P_{i-1,j,k}$ ， $P'' \in A$ ， $P' = P''$ ，则删除 $P' \in P_{i-1,j,k}$ 保留 $P'' \in A$ 。根据理论 1 中假设的 MC(MT) 算法计算顺序，处理 $C_{i-1,j,k}$ 结束后当前体素移至 $C_{i,j,k}$ 。集合 $P_{i-1,j,k}$ 中有 P' 被删除，且可能存在 $P = P' = P''$ ， $P \in P_{i,j,k}$ ，而 P'' 可能在 $P_{i-2,j,k}$ ， $P_{i-1,j-1,k}$ 或 $P_{i-1,j,k-1}$ 任意一个集合中，因此，当 $P_{i,j,k}$ 仅仅比对 $P_{i-1,j,k}$ ， $P_{i,j-1,k}$ ， $P_{i,j,k-1}$ 就可能遗漏重复点。因为 $C_{i,j,k}$ 与 $C_{i-1,j-1,k}$ 和 $C_{i-1,j,k-1}$ 有面或棱相邻接，所以 $P_{i,j,k}$ 还须比对 $P_{i-1,j-1,k}$ ， $P_{i-1,j,k-1}$ 。同样的情况也可能出现在 $C_{i,j-1,k}$ ， $C_{i,j,k-1}$ 中。因此，当 $P_{i,j,k}$ 压入 \tilde{P}_n 时不仅需要比对 $P_{i-1,j,k}$ ， $P_{i,j-1,k}$ ， $P_{i,j,k-1}$ ，还须比对与 $C_{i,j,k}$ 面或棱相邻接的体素 $C_{i-1,j-1,k-1}$ ， $C_{i-1,j,k-1}$ ， $C_{i-1,j-1,k}$ 与等值面的交集集合。综上所述，当 $P_{i,j,k}$ 压入 \tilde{P}_n 时，共需要比对 $P_{i-1,j,k}$ ， $P_{i,j-1,k}$ ， $P_{i-1,j-1,k}$ ， $P_{i,j,k-1}$ ， $P_{i-1,j,k-1}$ ， $P_{i-1,j-1,k-1}$ 才能保证 $P \in P_{i,j,k}$ 不为重复点。

3.2 方法的复杂度分析

将六体素法与穷举法和堆排序法的复杂度进行比较。

条件假设：设 $C(x,y,z)$ ，既此大立方体共有 xyz 个体素；定义每个体素与等值面的交点平均数为常数 n_0 ； $n_0 =$ 所有体素与等值面的交点/体素个数；设在 MC(MT) 算法进行过程中，当前体素坐标为 $C_{i,j,k}$ ， $C(k)$ 与等值面的交点集合为 $P(k)$ 。

3.2.1 时间复杂度分析

(1) 六体素法将 $P_{i,j,k}$ 压入 \tilde{P}_n ， $P_{i,j,k}$ 所需比对的平均点数为 $t_1, t_1 \leq 6n_0$ ；六体素法 $P(k)$ 压入 \tilde{P}_n ； $P(k)$ 总共需要比对的平均点数为 t_2 ，其中， $t_2 < 6xy n_0 = O(xy)$ 。

可见，六体素法在第 k 层构造 \tilde{P}_n 时时间复杂度只与 x 和 y 有关。在一个确定 x 和 y 的案例中，它需比对的点数 $6xy n_0$ 是一个常数。

(2) 穷举法将 $P_{i,j,k}$ 压入 \tilde{P}_n 时， $P_{i,j,k}$ 需要比对当前 \tilde{P}_n 中的所有点，其值为

$$g_1 = (xyk + xj + i)n_0$$

穷举法将 $P(k)$ 压入 \tilde{P}_n ， $P(k)$ 总共需要比对的平均点数为

$$g_2 = [xyk + (xyk + 1) + \dots + (xyk + xy)]n_0 =$$

$$\left[\sum_0^{xy-1} xyk + (0 + 1 + \dots + xy) \right] n_0 =$$

$$\left[\sum_1^{xy} xyk + \left(\frac{1}{2} xy(xy-1) \right) \right] n_0 =$$

$$\left(x^2 y^2 k + \frac{1}{2} x^2 y^2 - \frac{1}{2} xy \right) n_0 = O(x^2 y^2 k)$$

可见，穷举法的时间复杂度不仅与 x 和 y 有关，且与当前层位置 k 有关，随着计算过的体素增多，它需比对的点数成平方级增加。

(3) 堆排序先将所有体素与等值面的交点压入 P_n 再来构造 \tilde{P}_n ，完成 \tilde{P}_n 的时间复杂度为

$$d = O(xy k \lg(xy k))$$

比较 t_2, g_2 和 d ，可以发现六体素法时间复杂度与堆排序法相同，都优于穷举法。

3.2.2 点列空间复杂度分析

(1) 六体素法在构造 \tilde{P}_n 空间时已经剔除了重复点，所以需要的 \tilde{P}_n 空间与 $f(P_n)$ 所需要的空间大小近似。实验中 \tilde{P}_n 空间一般仅为 $f(P_n)$ 所需空间的 $1/2 \sim 2/3$ 。

(2) 穷举法与六体素法都是在构造 \tilde{P}_n 空间时剔除重复点，所以，它们的 \tilde{P}_n 空间复杂度相同。

(3) 堆排序法先将 $P_{i,j,k}$ 压入 P_n ，再构造 \tilde{P}_n 。因为三角面片由 3 个点组成，所以 P_n 应为 $f(P_n)$ 所需空间的 3 倍。即使不加上其辅助空间，一开始堆排序法所需要的 P_n 空间也至少为 $f(P_n)$ 所需空间的 3 倍。

综上所述，六体素法的 \tilde{P}_n 空间复杂度与穷举法相同，均大大优于堆排序法。

4 六体素法

算法如下：

$\tilde{P}_n, f(P_n)$ 初始化为空集；

for ($k = 0; k < z; k++$)

for ($j = 0; j < y; j++$)

for ($i = 0; i < x; i++$)

if ($C_{i,j,k}$ 与等值面有交点)

{
MC(MT) 算法计算出 $P_{i,j,k}$ 和 $(P_{i,j,k})$ ；

$P_{i,j,k}$ 与 $P_{i-1,j,k}, P_{i,j-1,k}, P_{i-1,j-1,k}, P_{i,j,k-1}, P_{i-1,j,k-1}, P_{i-1,j,k-1}$ (如果存在) 比较，得映射 f ；

根据 f 得 $\tilde{P}_{i,j,k} = f(P_{i,j,k})$ ，更改 $(P_{i,j,k})$ 为 $(\tilde{P}_{i,j,k})$ ；

将 $\tilde{P}_{i,j,k}$ 压入 \tilde{P}_n ， $(\tilde{P}_{i,j,k})$ 压入 $f(P_n)$ ；

}

else

continue

5 构造不同数据结构所需时间的比较

在笔者开发的医学 3D 图像重建系统中，使用六体素法完成了三维重建模块，并与没有采用六体素法的三维重建模块进行比较。

试验环境：CPU 为赛扬 D320(2.4 GHz)；内存为 256 MB × 2 DDR400；编译环境：WinXP 平台下 VC6.0；使用类库为 MFC 和 OpenGL。

使用 MT 算法计算 175 层的 512×512 的体素与等值面的交点队列和其构成的三角面片队列，不做任何处理，完成数据结构 A 并显示的时间为 14.3 s。

使用 MT 算法计算相同案例，期间使用六体素法构造三维点队列和基于其的三角面片队列，完成数据结构 B 并显示的时间为 17.9 s。

可以发现，两者的时间相差为 3.6 s，即六体素法所耗时间为 MT 算法的 25.17%。但此时数据结构 B 可以直接被网格简化算法作用，而数据结构 A 不能。实践证明六体素法是高效率的。

6 结束语

在分析、比较六体素法、穷举法和堆排序法的时间、空间复杂度之后，发现在时间复杂度上，六体素法与堆排序法处于同一数量级，都大大优于穷举法；在空间复杂度上，六体素法与穷举法处于同一数量级，都大大优于堆排序法。六体素法不仅平均复杂度较好，而且在计算机程序中直接连系了 MC(MT) 算法和网格简化算法所处理的数据结构，在 MC(MT) 算法抽取等值面的同时就完成了网格简化算法所要求的拓扑结构，减少了中间步骤，从而节约了算法运行的时间和空间，提高了效率。

参考文献

- [1] 秦绪佳. 医学图像三维重建及可视化技术研究[D]. 大连: 大连理工大学, 2001.
- [2] 潘志庚, 庞明勇. 几何网格简化研究与进展[J]. 江苏大学学报: 自然科学版, 2005, 26(1): 67-71.
- [3] 张必强, 邢 渊, 阮雪榆. 面向网格简化的 STL 拓扑信息快速重建算法[J]. 上海交通大学学报, 2004, 38(1): 39-42.
- [4] 罗梅伟. STL 文件在可视化曲面拼接中的应用[J]. 山东农机, 2005, 1: 7-9.