

# 海量数据的曲面分层重建算法\*

吕晟珉<sup>+</sup>, 杨勋年, 汪国昭

(浙江大学 数学系 计算机图像图形研究所, 浙江 杭州 310027)

## A Layered Surfaces Reconstruction Algorithm from Large Volumetric Datasets

LÜ Sheng-Min<sup>+</sup>, YANG Xun-Nian, WANG Guo-Zhao

(Institute of Computer Graphics and Image Processing, Department of Mathematics, Zhejiang University, Hangzhou 310027, China)

+ Corresponding author: Phn: 86-571-87951609 ext 8107, E-mail: lvsm@263.net

<http://www.zju.edu.cn>

Received 2002-06-07; Accepted 2002-10-19

Lü SM, Yang XN, Wang GZ. A layered surfaces reconstruction algorithm from large volumetric datasets. *Journal of Software*, 2003,14(8):1448~1455.

<http://www.jos.org.cn/1000-9825/14/1448.htm>

**Abstract:** Surfaces reconstruction from serial section images is a mature research in visualization of medical imaging. Traditional methods always adopt the process that starts on mesh reconstruction or isosurfaces extraction, then data simplification. Since the volumetric datasets are huge and segmented. Most of the existing algorithms that use large in-core data structures might be too large to off-load to disk. In this paper, an algorithm is presented that has been designed to reconstruct 3D surfaces from huge and segmented volumetric datasets. Approximate mesh can be reconstructed and simplified layer by layer. And user can control mesh complexity by parameters. Because memories are used rationally, some surfaces reconstruction from large datasets can work on PC using the algorithm.

**Key words:** surface reconstruction; mesh extraction; mesh fairing; mesh simplification; mesh merge

**摘要:** 从二维图像序列进行表面重建的问题由来已久,传统的重建方法通常是先重建或先等值面抽取,再简化数据量。随着处理数据量的增长,传统算法的中间过程会因为存储空间的限制不能进行下去。如何利用有限的存储空间对大数据量进行处理,从而完成曲面的重建曾是要研究的问题。针对大数据量的已分割的医学切片图像,利用逐层重建、即时简化的基本思想,给出一个易于操作实现、数据量可控制的算法。这样可以在硬件条件不太高的计算机(如内存不太大的个人微机)上实现大数据量的医学图像表面重建。

**关键词:** 曲面重建; 网格抽取; 网格光顺; 网格简化; 网格合并

中图法分类号: TP391 文献标识码: A

医学图像可视化是科学计算可视化中的一个重要分支。在医学研究中,很多数据来源于一些产生图像的设

\* Supported by the National Natural Science Foundation of China under Grant No.60073023 (国家自然科学基金); the National Grand Fundamental Research 973 Program of China under Grant No.G1998030600 (国家重点基础研究发展规划(973))

第一作者简介: 吕晟珉(1976—),男,浙江杭州人,博士生,主要研究领域为计算机图形学,医学图像处理,科学计算可视化。

备,比如 CT、MRI 或是 PET 扫描仪。由于从这些设备得到的数据均为二维断层图像,约束了人们对组织和器官的整体把握。对于如何根据连续断层图像重建物体表面的可视化技术,人们从 20 世纪 70 年代就开始进行研究。经过 20 多年的研究和开发,总结出许多效果不错的算法。但随着设备的改进,得到的二维图像越来越精细,算法要求处理的数据量也越来越大。以往的很多算法在中间过程中会因为计算机内存不够而被终止。本文主要研究的问题是如何在有限的存储空间内完成从已分割的二维序列切片图像中重建出人们感兴趣的医学组织和器官的表面。

从可视化的方法来分,连续二维图像序列重建曲面大致可以分为两大类:基于轮廓线的曲面重建和基于体素的曲面重建。基于轮廓线的曲面重建<sup>[1,2]</sup>是从二维图像中找出感兴趣部位的边界,该边界被称为轮廓线,然后建立相邻两张图像中的轮廓线的对应关系,由此构造出物体的表面;基于体素的曲面重建,首先要求将连续二维图像序列所生成的数据空间划分成一个个小的立方体,类似于二维地图的等高线的求法,在数据空间提出了等值面的抽取。Marching Cubes(MC)算法<sup>[3]</sup>就是其中最典型的等值面算法。

本文例子中所处理的数据是已分割的医学序列切片图像数据,即切片图像中的每个像素点给出所属的医学组织的编号值。序列切片每片均视为有一定的厚度,可以根据采样步长的选择,将数据空间划分为体素的集合。由于医学组织编号值(segment 值)是一些离散的整数值,所以在这里不能像 MC 算法一样将 segment 值认为是线形分布的,而只能认为整个体素仅拥有一个 segment 值,即将该体素认为是在该 segment 值对应的医学组织内。对于用户指定的任意一个医学组织(输入 segment 值),该组织的表面可以看作是由具有该 segment 值的体素的可见面的集合。如果数据量较小,计算机内存可以容纳,就可以直接得到其表面。由于处理数据的不同,决定了本文处理的问题与传统的“轮廓线重建”、“等值面抽取”等方法的不同。

在三维曲面重建中,常用的表示方法是多边形网格。由于 OPENGL 强大的功能及三角片在 OPENGL 中的高效性,通常采用三角网格来表示三维表面。本文中采用的也是三角网格的表示法。由于本文中的序列切片分辨率较高,直接由可见面抽取得到的网格具有面片小、不光滑、数据量特别巨大等缺点。所以,网格简化和网格光顺也是本文提出的算法中不可缺少的部分。

## 1 相关工作

对于大数据量的网格数据处理人们从近两年才开始关注。目前对于“海量”数据的处理大致采取两种策略:(1) 将数据分片,每片可以用已有的方法处理。(2) 将数据有限制地降低采样密度(down-sampling),使之能用已有的方法进行处理。本文的算法将会采用上述策略,为了合理使用有限的存储空间,尽可能地使数据分层处理,将简化后的数据再合并得到所求网格。由于总体采用分层处理的思想,所以网格简化和光顺将会比较侧重利用局部信息进行处理的算法。本文的网格光顺算法采用类似 Taubin<sup>[4]</sup>的光顺算法,将直接抽取得到的有棱角的网格光顺出其大致的形状。

以往算法所处理的切片图像的分辨率一般都小于  $512 \times 512$ ,本文的切片图像是  $688 \times 720$ ,即使只有一层数据也有近 500K 个体素。且原始的切片就有 235 张,如果精度要求高,将插值产生更多的中间切片(对已分割的切片图像插值算法可见文献[18])。可以想象,对于一些比较复杂的医学组织,如大脑等,直接从体素的可见面抽取三角形网格,数量极多,内存需要大,不便后期的各类操作。所以网格简化对大数据量的医学图像可视化来说是一个很重要的关键步骤。对网格的简化策略,可以大致分为:(1) 共面的小面片合并<sup>[5~7]</sup>;(2) 网格的控制点/边/面的减少<sup>[8~11]</sup>;(3) 利用能量方程来优化<sup>[12,13]</sup>;(4) 顶点簇替代<sup>[14,15,19]</sup>;(5) 基于小波的逼近方法<sup>[16,17]</sup>。类似能量方程,小波的方法对全局优化效果较好,对大数据量的网格来说,由于采用分层处理的思想,故不方便使用。所以对大数据量网格采用上述中(1)(2)(4)的方法较多。在本文中,从数据结构和算法的策略出发,第(2)种方法是最合适的。

传统的三维曲面重建技术,不论是基于轮廓线重建还是基于等值面抽取,都是先重建或先等值面抽取,再简化结果网格的数据量。这些方法对于较小的二维图像数据效果比较好。特别是加入种子生长等表面跟踪技术后的改良的 MC 算法<sup>[20,21]</sup>,能快速地重建出曲面,并且算法稳定。但当数据量较大时,采用先重建(或是抽取)再简化的方法会因为直接重建(或抽取)出来的总网格数据量太大,不能在内存中存放而不能进行以后的简化步骤,因

而不能重建出最终的结果曲面.所以本文考虑分层处理的方法,采用逐层重建,即时简化,根据硬件的条件直接重建出简化网格模型.这样做的优点和目的是能够充分利用有限的内存和硬件条件来处理海量数据,如医学图像三维重建等.此外,本文方法的思想还可以应用在其他大数据量网格模型的简化、传输等方面.

## 2 曲面分层重建算法

### 2.1 算法概述

在讲述本文的算法之前,先说明文中所用网格的数据结构:

```
typedef struct TypeVertex
{
    MCVector3f v; //三维向量,记录顶点的空间位置
    MCVector3f vn; //三维向量,记录顶点的法向
    short type; //分类标识,在简化、合并中将被使用
} VERTEX;
typedef struct TypeTriangle
{
    int v0,v1,v2; //三角形 3 个顶点在顶点数组中的索引(index)
} TRIANGLE;
typedef struct TypeMesh
{
    long max_vertex; //顶点的总个数
    VERTEX *vertex; //顶点数组
    Long max_triangle; //三角形的总个数
    TRIANGLE *triangle; //三角形数组
} MESH.
```

由于处理的数据量比较大,在本地计算机上,算法必须能够运用有限的存储量和运算量来建立拓扑关系正确、几何准确的网格曲面表示.本文方法的主要思想是:逐层重建,及时简化,由各切片数据依次重建组织的表面表示,并建立其简化表示.而在实际操作中容易发现,如果将整个数据空间分割成若干个互不相交的数据层来分别处理的话,网格光滑和网格合并有着一定的不可调和性:网格合并要求相邻两层在接口处的网格顶点有相同的空间位置和法向,否则合并后会产生缝隙或皱褶;而在网格光滑中,每层抽取的原始网格中的层与层连接的顶点若不调整,对它们的一重邻点和二重邻点影响很大,即使加大光滑次数,光滑的效果也不好;若调整,网格合并则会产生问题.所以本文算法中对数据场的分割得到的数据层是有一定的相交部分的,这样可以很好地解决上述问题.具体细节见第 2.5 节.

为了便于说明,本文约定 `active_mesh` 来存储每个数据层需处理的网格.而 `mesh` 是存储经过光滑简化后最终的简化网格,也就是最终所求的网格.其中 `active_mesh` 和 `mesh` 都是 MESH 型的数据.在这里可以将本文的算法简单地描述如下:

对每层切片,进行以下几个步骤的操作:

网格抽取(mesh extraction).对于新加入的切片,找出用户指定的 `segment` 值的体素的可见面,将其三角化从而形成三角网格,添加进入 `active_mesh` 中.

网格光滑(mesh smoothing).对 `active_mesh` 中的三角网格光滑,调整网格顶点的空间坐标,并使其法向变化连续,减少原始网格的棱角.

网格简化(mesh simplification).对 `active_mesh` 中经过光滑的三角网格进行简化,能在较好保持网格形状的前提下减少数据的存储量.

网格合并(mesh merge).将 active\_mesh 中的网格分裂成上、下两部分,上部分添加进 mesh 中,下部分依然在 active\_mesh 中,作为两个数据层的交的部分,进入下一个数据层的处理.

对所有的切片处理完毕,mesh 中就得到了整个曲面重建后的简化网格模型.

## 2.2 网格抽取

在以往的等值面抽取的算法中,有不少基于面跟踪的算法,对提高算法效率有很大帮助.这些算法往往是由一个种子点出发利用空间的邻域来进行跟踪,从而减少对“空”的小立方体的检索,提高搜索的效率.由于医学组织的表面几何复杂(有可能结果是由离散的曲面集合组成),不可能交互选择种子点,所以不能利用图像的连贯性,本文的网格抽取算法只能采用对采样点逐点判别的方法,这样,对任何复杂的曲面都能很好地表示出来.对于大小不同的医学组织,可以采取不同的采样步长.由于切片被认为是有一定厚度的,不同的平面上的采样步长就决定了对数据空间的不同划分.大的采样步长意味着降低了采样密度,从而减少了小立方体的个数.在这里,网格抽取首先要从所有 segment 值为用户指定的值的小立方体中找出边界.虽然切片数据中每个点都给出 segment 值,但要找出边界小立方体,还是要对该层中的每个 segment 值为用户指定值的小立方体进行判断:对它的每个面进行判断是否为可见面,若是,则将该面划分为两个三角形,将三角形的顶点的信息添加进 active\_mesh 中,其中三角形的顶点法向取为相邻可见面的法向加权之和,对于层与层之间的边界点,在顶点的分类标识(具体可参见上述的数据结构)上注明.

## 2.3 网格光顺

对 active\_mesh 中顶点数组中的每个点  $V$  (设法向为  $\Delta V$ ),可以从三角形数组中找出它的重邻点,设其重邻点为  $\{V_1, V_2, \dots, V_b\}$ ,对应的法向分别是  $\{\Delta V_1, \Delta V_2, \dots, \Delta V_b\}$ ,则  $V$  调整后的新坐标  $V^*$  及新法向  $\Delta V^*$  可用下述公式求得:

$$\begin{cases} V^* = V + \lambda(\bar{V} - V), \\ \Delta V^* = \Delta V + \mu(\Delta \bar{V} - \Delta V), \end{cases}$$

其中  $\bar{V} = \frac{1}{b} \sum_{i=1}^b V_i$ ,  $\Delta \bar{V} = \frac{1}{b} \sum_{i=1}^b \Delta V_i$ ,  $\lambda, \mu$  是值在 0,1 之间的权因子( $\lambda, \mu$  的取值有一定经验性,从测试结果来看,选择  $\lambda = 0.4, \mu = 0.1$ ).

算法中对顶点数组中的每个点要计算多遍:

```
for (iteration=0; iteration<max_interation; iteration++)
    for (every vertex in active_mesh)
    {
        compute new position;
        compute new normal;
    }
```

由于该光顺方法会产生一定的收缩现象,特别是在 max\_interation 较大时比较明显.在实际运算中,max\_interation 取 5,6 时效果就可以了.需要说明的是,在光顺前,需记录 active\_mesh 中与上下层连接的边界点的位置信息和法向信息,在光顺后,再将些点赋回原值,防止网格合并时缝隙或皱褶情况的产生.

## 2.4 网格简化

active\_mesh 中记录的三角网格由于是直接由小立方体表面划分得到,所以网格很细且多.可以在保持网格形状的前提下,对网格进行简化,减少顶点的数目,以减少所需的内存.在本文中采用的是边合并的方法,如图 1 所示.

当然,可合并的顶点是有条件的,首先要对 active\_mesh 的顶点进行分类,分为可去点和不可去点.设顶点  $V$ , 其一重邻点为

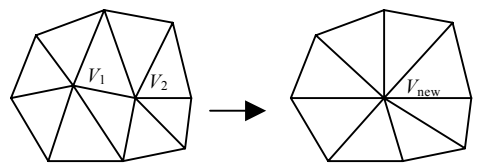


Fig.1 Edge collapse  
图 1 边合并操作

$\{V_1, V_2, \dots, V_b\}$ , 对应的法向分别是  $\{\Delta V_1, \Delta V_2, \dots, \Delta V_b\}$ , 则定义

$$d(V, \{V_1, V_2, \dots, V_b\}) = |(V - \bar{V}) \cdot \Delta \bar{V}|,$$

其中  $\bar{V} = \frac{1}{b} \sum_{i=1}^b V_i, \Delta \bar{V} = \frac{1}{b} \sum_{i=1}^b \Delta V_i$ . 这里, 定义距离的几何意义是求顶点  $V$  对其一重邻点所拟合的平面距离的近似值. 当顶点  $V$  不是在 active\_mesh 中的网格边界点, 且  $d(V, \{V_1, V_2, \dots, V_b\}) < \epsilon_{\text{vertex}}$ , 则顶点  $V$  为一个可去点. 反之, 若不满足条件, 该点就是不可去点. 这里的  $\epsilon_{\text{vertex}}$  是用户事先设置的参数, 可以控制可去点的个数, 从而控制网格的顶点数目. active\_mesh 中的网格边界点一定是不可去点.

在对 active\_mesh 中的顶点数组进行分类后, 再检索 active\_mesh 的三角形数组, 若一个三角形中有两个可去点, 则将该两点所在的边蜕化为一个点, 蜕化出来的新点的属性也是可去点. 在 active\_mesh 的顶点数组中去掉原来的可去点, 添加新点, 并在三角形数组中将原来与老的可去点相邻的点与新点相邻.

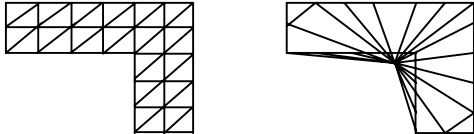


Fig.2 New vertex outside without limit

图2 不设边长限制出现的新点“越界”情况

制, 尽可能地避免新点越界的发生.

而在边合并的过程中, 需要用户输入的边长限制  $d_{\text{edge}}$ , 否则产生的新点会因为一系列的边合并而到实体的外部 (如图 2 所示). 可以根据数据场的不同, 设置不同的边长限制

### 2.5 网格合并

这一步是要将 active\_mesh 中的网格信息添加进总的网格数据 mesh 中. 需要详细说明的是, 在网格处理中如何做到逐层处理. 在这里, 算法考虑的是“浮动”的数据层. 在对 active\_mesh 初始化之前, 预先给定一个  $m$  值 ( $m \geq 2$ ), 找到含有用户指定的 segment 值的最顶切片, 读入  $m$  片连续切片数据作为一个数据层, 假设为第  $a \sim$  第  $a+m-1$  片切片. 此时, 这个数据层产生的网格的边界点由两部分组成:  $A = \{\text{第 } a \text{ 片切片分割产生的小立方体的上表面上的网格点}\}$  和  $B = \{\text{第 } a+m-1 \text{ 片切片产生的小立方体的下表面上的网格点}\}$ . 从第 2.3 节的说明中可以知道, 在 active\_mesh 中, 上下层相连的网格边界点通过光顺这一步是没有发生变化的, 并且第 2.4 节也保证了这些边界点不会被删除. 该数据层的数据在 active\_mesh 中经过光顺、简化, 最终可以认为是一个简化的网格. 在网格合并这一步中 (如图 3 所示), 将与集合  $A$  中的点有连接关系的点, 添加入 mesh 的顶点数组内, 将对这些点在 active\_mesh 中的 index 和在 mesh 中的 index 作映射, 就可以将 active\_mesh 中三角形的连接关系添加入

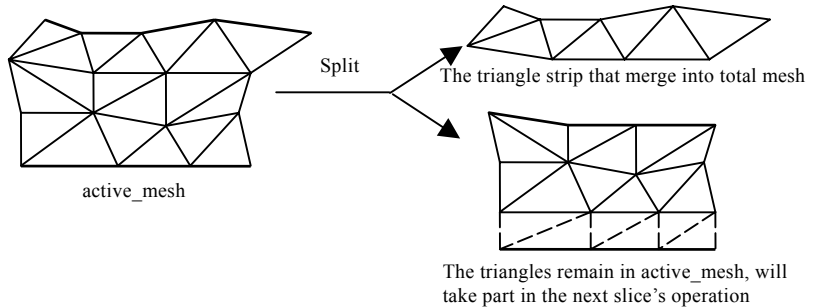


Fig.3 Illustration of mesh merger (the bold line is the boundaries)

图3 网格合并的演示图(粗实线是网格的上下边界线)

mesh 的三角形数组中表现出来. 在 active\_mesh 中的顶点数组内删除  $A$  中的点, 并在三角形数组中删除这些添加入 mesh 的三角形, 由于顶点数组变动过, 将保留下来的三角形对应的顶点在顶点数组的 index 作相应的修改. 当对下一片切片进行操作时, 读入第  $a+m$  片的数据, 对新的数据片进行网格抽取, 此时可以认为现在考虑的数据层是第  $a+1 \sim$  第  $a+m$  片切片的数据层. 而现在的边界点由上边界点, 即与集合  $A$  中的点有连接关系但本身不是在  $A$  中的点, 下边界点则是第  $a+m$  片切片产生的小立方体的下表面上的网格点. 原来集合  $B$  中的点将不再是边界点, 将会参与本次的光顺及简化操作. 这样可以使除该组织表面曲面的上边界和下边界以外的所有层网格边界点都有得到光顺的机会.

### 3 例子及讨论

#### 3.1 实例

本文使用的数据是向科罗拉多州 golden standard multimedia 购买的,为以层为单位的 PHOTO 图像数据和 SEGMENT 图像数据,经过无用信息的剔除,最终的切片大小为  $688 \times 720$ ,用立方体的形式保存在数据库中,便于网络的共享访问.为了说明本算法能在较小的内存中对大数据量的切片进行表面重建.本文计算所用的微机仅使用 64MB 内存,操作系统为 Windows NT 4.0 workstation,同时微机上开启 SQL Server,通过数据库读取原始切片的数据.由于内存较小,以下图中的  $m=2$ .图 4 是通过本文算法重建的人体小脑模型.

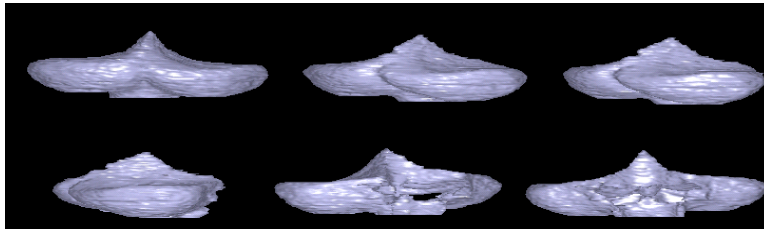


Fig.4 Images of cerebellum mesh with different viewpoint

图 4 小脑的多方位视图

比较一下相同模型(大脑模型),不同参数下传统算法(算法 I)和本文分层重建算法(算法 II)所需的数据存储量,重建的效果如图 5 所示,具体数据见表 1.

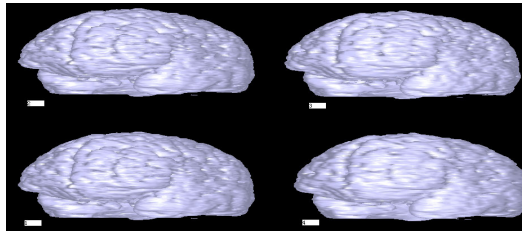


Fig.5 Brain meshes reconstruction with different parameters (parameters detail in table 1)

图 5 不同参数下得到的大脑模型(具体参数见表 1)

Table 1 Mesh datasets comparison with different parameters on brain model

表 1 不同控制参数下大脑模型数据量的比较

Sampling step	$\epsilon_{\text{vertex}}$	$d_{\text{edge}}$	Vertices number		Triangles number		
			Algorithm I	Algorithm II	Algorithm I	Algorithm II	
1	4	1.5	8	164 241	26 688	333 012	54 650
2	4	2	8	164 241	26 576	333 012	54 422
3	6	2	8	89 180	22 200	179 828	45 240
4	8	2	10	58 367	15 575	117 216	31 698

不同模型相近参数下,传统的算法(算法 I)和本文分层重建算法(算法 II)所需的数据存储量.重建的效果如图 6 所示,具体数据见表 2.



Fig.6 Other models' meshes (Hyoid Bone, Axis, Cerebellum)

图 6 其他模型(舌骨,脊椎骨,小脑)

Table 2 Mesh datasets comparison between algorithm I and II on other models

表 2 算法 I, II 在其他模型数据量的比较

Sampling step	$\varepsilon_{\text{vertex}}$	$d_{\text{edge}}$	Vertices number		Triangles number		
			Algorithm I	Algorithm II	Algorithm I	Algorithm II	
Hyoid Bone	1	1.5	4	13 787	929	27 660	1 880
Axis (C-2)	2	1.5	5	20 506	2 926	40 900	5 934
Cerebellum 1	2	1.5	5	79 065	9 938	159 460	20 366
Cerebellum 2	4	1.5	5	27 321	7 825	55 292	15 972

### 3.2 讨论及总结

在本文的算法中,  $m(\geq 2)$  是用户可以根据运行计算的内存大小及医学组织的形状大小, 设置 active\_mesh 处理数据层的切片数目, 一般  $m$  大, 中间层切片产生的网格相对更光滑. 采样步长对数据量的控制影响很大, 可以估计网格大致的数据量,  $\varepsilon_{\text{vertex}}$  可以控制网格顶点中近似平面的顶点个数, 对网格平坦之处的处理影响很大, 影响着网格质量的好坏.  $d_{\text{edge}}$  控制着网格简化时产生的新点尽可能地不越界, 对网格数据量的影响较小. 用户可以交互地调节  $m$ , 采样步长,  $\varepsilon_{\text{vertex}}$ ,  $d_{\text{edge}}$ , 使之能在内存较小的微机上较好地实现大数据量的曲面重建.

从例子中的表上数据可以看到, 直接表面抽取的数据量很大, 那些先重建(抽取)再简化的算法内存需要也很大, 一般微机难以应付. 本文的算法是基于已分割的医学序列切片图像数据, 逐层重建出网格, 并即时对网格进行简化. 这样能够较大地减少内存的使用量, 使得该算法能在普通微机上较好地实行. 本文算法在对大数据场分层处理的同时能保证结果网格不会出现皱褶和裂缝. 网格简化和光滑可以采取其他不同的方法. 从上述来看, 本文算法也存在着一一定的问题, 主要是网格抽取中由于是对采样点逐点检测的, 所以检索效率不高.

本算法是专门为大数据量的曲面表示而设计的. 它具有以下几个特点, 可以较好地解决数据量大、计算量大、曲面几何复杂的困难.

- (a) 数据量可控制. 可由曲面采样点的步长及简化时的参数控制, 控制网格点的数量.
- (b) 运算量经济. 逐层重建, 简化, 运算量相对较小.
- (c) 易于操作, 实现.

### References:

- [1] Baba N, Computer-Aided three-dimensional reconstruction from serial section images. In: Image Analysis in Biology. CRC Press, 1991. 251~270.
- [2] Boissonnat JD. Shape reconstruction from planar cross sections. In: Computer Vision, Proceedings of the Graphics and Image. 1988, 44:1~29.
- [3] Lorensen NE, Cline HE. Marching cubes: a high resolution 3D surface construction algorithm. Computer Graphics, 1998,21(4): 163~169.
- [4] Gabriel T. A signal processing approach to fair surface design. In: Proceedings of the Computer Graphics, Annual Conference Series. Los Angeles: ACM Press, 1995. 351~358.
- [5] De Hamer MJ, Zyda MJ. Simplification of objects rendered by polygonal approximations. Computer & Graphics, 1991,15(2): 175~184.
- [6] Hinker P, Hansen C. Grometric optimization. In: Proceedings of the IEEE Visualization'93. Los Alamitos: IEEE Computer Society Press, 1993. 189~195.
- [7] Kalvin AD, Taylor RH. Surfaces: Polygonal mesh simplification with bounded error. IEEE Computer Graphics & Applications, 1996,1(3):64~77.
- [8] Algorri ME, Schmitt F. Mesh simplification. Computer Graphics Forum, 1996,15(3):78~86.
- [9] Garland M, Heckbert P. Surface simplification using quadric error metrics. In: Proceedings of the Computer Graphics, Annual Conference Series. Los Angeles: ACM Press, 1997. 209~216.
- [10] Hamann B. A data reduction scheme for triangulated surfaces. Computer Aided Geometric Design, 1994,11(2):197~214.
- [11] Schroeder WJ, Zarge JA, Lorensen WE. Decimation of triangle mesh. ACM Computer Graphics, 1992,26(2):65~70.

- [12] Hoppe H, DeRose, T, Duchamp T, McDonald J, Stuetzle W, Mesh optimization. In: Proceedings of the Computer Graphics, Annual Conference Series. Anaheim: ACM Press, 1993. 19~26.
- [13] Hoppe H. Progressive meshes. In: Proceedings of the Computer Graphics, Annual Conference Series. New Orleans: ACM Press, 1996. 99~108.
- [14] Low KL, Tan TS. Model simplification using vertex clustering. In: Proceedings of the 1997 Symposium on ACM Symposium on Interactive 3D Graphics. 1997. 75~81.
- [15] Rossignac J, Borrel P. Multi-Resolution 3D approximation for rendering complex scenes. In: Falcidieno B, Kunii TL, eds. Geometric Modeling in Computer Graphics. Berlin: Springer-Verlag, 1993. 445~465.
- [16] Hebert DJ, Kim HJ. Image encoding with triangulation wavelets. Proceedings of the SPIE, 1995,2569(1):381~392.
- [17] Gross MH, Staadt OG, Gatti R. Efficient triangular surface approximations using wavelets and quadtree data structures. IEEE Transactions on Visualization and Computer Graphics, 1996,2(2):130~144.
- [18] Li JY, Yang XN, Wang GZ. A local interpolation algorithm for segmented images. Journal of Zhejiang University Science, 2002,29(1):55~62 (in Chinese with English abstract).
- [19] Lindstrom P. Out-of-Core simplification of large polygonal models. In: Proceedings of the Computer Graphics, Annual Conference Series. New Orleans: ACM Press, 2000. 259~262.
- [20] Lee TY, Lin CH. Growing-Cube isosurface extraction algorithm for medical volume data. Computerized Medical Imaging and Graphics, 2001,(25):405~415.
- [21] He HG, Tian J, Zhao MC, Yang H. A 3D medical imaging surface reconstruction scheme based on segmentation. Journal of Software, 2002,13(2):219~226 (in Chinese with English abstract).

#### 附中文参考文献:

- [18] 李将云,杨勋年,汪国昭.分割图像插值的一种局部算法.浙江大学学报(理学版),2002,29(1):55~62.
- [21] 何晖光,田捷,赵明昌,杨骅.基于分割的三维医学图像表面重建算法.软件学报,2002,13(2),219~226.