

文章编号: 1000-582X(2012)12-068-10

# 融合构型查找表与邻接查找子表的改进 MC 方法

王旭初<sup>1a,1b</sup>, 王 赞<sup>1a,1b</sup>, 牛彦敏<sup>2</sup>, 张绍祥<sup>3a</sup>, 谭立文<sup>3a</sup>, 晋 军<sup>3b</sup>

(1. 重庆大学 a. 光电工程学院; b. 光电技术及系统教育部重点实验室, 重庆 400044; 2. 重庆师范大学 计算机与信息科学学院, 重庆 400050; 3. 第三军医大学 a. 基础部解剖学教研室; b. 新桥医院  
全军心血管内科中心, 重庆 400038)

**摘 要:** 针对医学图像三维可视化中移动立方体面绘制算法(marching cubes, MC)执行速度慢、效率不高的问题, 提出了融合构型查找表与邻接查找子表的改进 MC 方法。该方法通过显性构建邻接查找子表约束体元搜索路径, 使面绘制时只处理有效体元, 根据邻接查找子表特点设计堆栈结构实现搜索算法, 不仅提高了算法访问效率, 而且减少了临时存储空间。在可视化工具包(VTK)下用改进 MC 方法对人体脚、胸腔、头部的 CT 数据集进行三维重建实验, 结果表明在不损失重建质量的前提下, 重建过程中遍历立方体数目缩短 95% 左右, 重建时间缩短 20% 左右, 提高了 MC 方法的执行速度和重建效率。

**关键词:** 三维可视化; 面绘制; 移动立方体; 查找表; 医学图像重建  
**中图分类号:** TP391.4 **文献标志码:** A

## Improved marching cubes by combining case lookup table and adjacency lookup sub-table

WANG Xuchu<sup>1a,1b</sup>, WANG Zan<sup>1a,1b</sup>, NIU Yanmin<sup>2</sup>, ZHANG Shaoxiang<sup>3a</sup>,  
TAN Liwen<sup>3a</sup>, JIN Jun<sup>3b</sup>

(1. a. College of Optoelectronic Engineering; b. Key Laboratory of Optoelectronic Technology and Systems, Ministry of Education, Chongqing University, Chongqing 400044, China; 2. College of Computer and Information Science, Chongqing Normal University, Chongqing 400050, China; 3. a. Department of Anatomy, Third Military Medical University; b. Division of Cardiology, Xinqiao Hospital, Third Military Medical University, Chongqing 400038, China)

**Abstract:** The marching cubes (MC) is an effective surface rendering method in three-dimensional visualization for medical image sequence. However, most existing MC algorithms are slow and inefficient because they have to process all the cubes for isosurface extraction. An efficient MC algorithm is proposed by combing the case lookup table and a novel adjacent lookup sub-table to exclude unrelated empty cubes. By explicitly building the fix-length adjacent lookup sub-table that is independent on image sequences, the

收稿日期: 2012-07-15

基金项目: 国家自然科学基金资助项目(60903142, 61190122); 重庆市自然科学基金资助项目(CSTC2009BB3192, CSTC2011jjA40024); 重庆市科技攻关项目(CSTC2009AB5196); 中央高校基本科研业务费资助项目(CDJZR10120003, CDJXS10120010); 中国博士后基金资助项目(2012M521677)

作者简介: 王旭初(1976-), 男, 重庆大学副教授, 博士, 主要从事医学图像分析与模式识别、医学可视化的研究,  
(Tel)023-65102516; (E-mail)seadrift.wang@gmail.com.

volumes elements are constrained to those only intersect with the isosurface. Both execution time and temporary storage space are further reduced by incorporating the heap data structure in algorithmic implementation. Experimental results on human head, chest foot CT data sets by using the visualization toolkit package show that the traversal cubes decreases by 95%, and the reconstruction time decreases by 20% without any loss of reconstruction quality. Therefore, the proposed method can remarkably speed up the rendering time and be easily integrated into 3D visualization for clinical application.

**Key words:** three-dimensional visualization; surface rendering; marching cubes; lookup table; medical image reconstruction

医学图像可视化是科学计算可视化的一个重要领域,它将由 CT、MRI 等成像技术获得的二维数字化切片在计算机上直观表现为三维效果,提供传统手段无法获得的结构信息,清晰、全面地将病变体及其它感兴趣区域反映给医生,便于医生从多角度、多层次进行观察和分析,使医生有效参与数据的处理和分析过程,在辅助诊断、手术仿真、引导治疗等方面具有重要作用。

现有医学图像可视化方法主要分为 2 类:体绘制和面绘制<sup>[1]</sup>。体绘制方法根据光线传输方程直接将三维数据场映射到二维图像平面上,直接对体素进行操作得到绘制结果。优点是重建模型信息完整,图像质量较高;缺点是计算量较大,图像生成速度慢,且结构复杂,很难得到准确的网格剖分。面绘制方法通过阈值或极值方法由三维数据场构造中间几何图元并生成最终图像。典型代表算法包括基于切片级的轮廓线法与基于体元级的等值面抽取算法(如移动立方体算法)2 种。这类方法计算量小,重建速度较快,便于人机交互;缺点是只包含表面信息,重建结果信息不完整,容易产生伪像,重建结果的保真性没有体绘制好。针对上述 2 类算法的优缺点,在医学图像三维可视化中,常对切片序列采用面绘制方法进行三维重建来快速获得某些器官或病灶等目标的整体形貌。

移动立方体(marching cubes, MC)算法是三维数据场等值面构建的主流算法<sup>[2]</sup>,较好解决了任意不规则、非线性体数据场中进行等值面重建的问题。该算法在医学切片三维重建中它可通过对二维切片图像进行一系列的预处理后,还原出被检物体的三维形貌。标准 MC 算法进行三维重建的主要不足是需要访问所有立方体体元,重建效率较低,生成三角面片数量过多,包含了大量冗余信息,从而使重建图像占用存储空间过大,显示时间过长,重建后图像难以实现实时交互显示<sup>[3]</sup>。对生成的三角形网格进行化简可大量减少 MC 算法产生的三角片数量;通过

改善三角形网格的存储结构,可进一步减少其存储量<sup>[4]</sup>。但许多改进算法都是以增加内存、降低重建质量为代价,且对立方体邻域信息未显性表达<sup>[5-6]</sup>。

文中结合医学图像可视化的特点对标准 MC 算法进行了改进,通过显性构建查找子表,在遍历体元过程中只需判断那些有等值面通过的立方体,从而大大减少等值面绘制时判断的立方体数量,缩短绘制时间,使重建后图像能够有效存储及实时显示,提高了交互能力。

## 1 MC 算法原理

### 1.1 MC 算法及构型查找表

MC 算法作为体元单元内等值面抽取技术的代表,其基本原理是将数据场中上下 2 层相邻的 8 个数据点视为立方体体元顶点,逐层、逐行、逐列计算数据场中立方体的等值面来连接成目标的等值面。算法假定原始数据是离散的三维空间规则数据场,体数据分布于数据场的网格节点且局部线性连续,首先利用查找表分离出与等值面相交的体元,然后采用插值方法计算出等值面与立方体边的交点,接着根据体元各个顶点与等值面的相对位置将交点连接成等值面并视为其在该体元内的一种逼近表示,最后集合所有体元中的等值面构成整个数据场的等值面。

MC 算法采用构型查找表确定等值面构型,对每一立方体体元,比较给定阈值和体元中 8 个顶点值,判断顶点与等值面的位置关系,根据比较结果对体元进行分类。每一体元中每个顶点共有 2 种状态,所有顶点共有 256 种组合状态,根据互补对称性和旋转对称性,可简化成 15 种基本构型<sup>[2]</sup>,并以此建立一个构型查找表:

```
int C_LUTable[256][16]=
{
{-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1}
```





像分割方法<sup>[17]</sup>的快速实现。

## 2.2 算法实现

标准 MC 算法在寻找与等值面相交的体元立方体时,需要逐步移动和访问立方体体元,且每个立方体体元只能访问 1 次,由于该算法需要访问每一体元,算法实现中数据结构设计并不起关键作用,只需遍历即可。而文中算法是自适应确定待访问体元,因此数据结构设计对算法实现具有重要作用,合理的数据结构设计能够避免立方体访问混乱,降低中间存储量,提高算法执行效率。

由于算法具有明显的递归思想,在进行等值面绘制的时候可采用膨胀立方体的方式,即通过 1 个种子立方体根据其邻接立方体是否有等值面通过,若有则进行膨胀,否则终止,通过此方法一直循环直至获得等值面。如表 1 所示,算法的具体流程为:首先根据 C\_LUTable 找 1 个与等值面相交的体元作为种子,求出其等值面并把访问标志状态设为 0;接着由其 8 个顶点的状态值查找 A\_LUTable 表,若访问标志状态不为 0,就把邻接立方体的 1 个顶点的序列号压入堆栈并将其标志状态设为 0,用此序列号来表示邻接立方体;若栈不为空则弹出栈顶元素(即 1 个顶点的序列号),通过该序列号构建立方体体元,并求出其等值面,据此循环直至栈中为空。

从表 1 可看出,算法流程中的堆栈数据结构、标志数组、立方体存储形式这三者的设计是保证算法

鲁棒性的关键因素。其中堆栈数据结构要求从任一目标表面立方体体元出发均可返回该体元,保证了无需遍历每一立方体。标志数组确保了立方体访问的唯一性。该数组记录了立方体的访问状态,若立方体体元已被访问则加入该体元索引值。在初始状态下根据立方体 8 个顶点的状态值来设置标志数组,若状态值为 0 与 255 之间,则 Flag 数组非零。待绘制目标,明确结构,Flag 数组可改为队列、简化初始化流程来进一步加快访问速度。立方体存储形式设计精简了算法中的临时存储量。由于三维数据场中每个数据点都有且只有唯一的序列号(在 VTK 平台下序列号为 id 号),在实施面绘制时若把每一个立方体的所有信息都保存起来需占用很大存储空间,因为每一个立方体所要存储的信息包括 8 个角点的坐标值和 8 个角点的序列号,这样 1 个立方体要占用的存储空间为 48 个字节,某数据场中的立方体个数为 40 000 000 个则需要 1 920 000 000 个字节的内存,在 16 位计算机上则会造成内存溢出。因此文中算法实现时只存储立方体的一个角点,来代替体元立方体,这样可大大减少存储空间。在求取等值面时,可根据这一角点信息,结合 C\_LUTable 与 A\_LUTable 构建它所在的立方体体元。利用该角点信息还可直接判断 Flag[q] 的值,避免了步骤(3)中从 Flag 队列查找值为 q 的元素时的循环操作。

表 1 文中算法主要流程

|                                                                                                                   |
|-------------------------------------------------------------------------------------------------------------------|
| 输入:图像切片序列;等值面阈值                                                                                                   |
| 输出:目标三角面片结构 T                                                                                                     |
| 全局变量:查找表 C_LUTable;邻接立方体查找子表 A_LUTable                                                                            |
| 辅助的数据结构:标志数组 Flag;堆栈 Stack                                                                                        |
| BEGIN                                                                                                             |
| 1)初始化 Flag,根据 C_LUTable 确定立方体构型,若为 0 或 255,则 Flag[p]=0,否则 Flag[p]=1,p 为立方体左下角点所对应的序列号;                            |
| 2)从左下角起,找一个构型不为 0 或 255 的立方体作为种子立方体,其左下角点对应的序列号为 q,把 q 压入堆栈 Stack;                                                |
| WHILE Stack 非空                                                                                                    |
| 3)从 Stack 中弹出元素 q,在 Flag 队列中查找值为 q 的元素,若为空,根据 q 构造立方体体元 S;                                                        |
| 4)对 S 求得等值点,计算法向量,记录等值面三角面片结构 T,且置 Flag[q]=0;                                                                     |
| 5)根据 S 的 8 个角点情况确定 C_LUTable 索引值,进而定位 A_LUTable 值,得到有等值面通过的邻接立方体序列(设其中某一立方体左下角点的序列号为 m),若 Flag[m]=1,将 m 压入 Stack; |
| END WHILE                                                                                                         |
| 6)根据 T 绘制等值面。                                                                                                     |
| END                                                                                                               |

### 2.3 与相关方法比较

文中算法中采用了结合当前立方体体元信息推进到相邻立方体体元,从而避免全局搜索的策略,类似思想在部分 MC 改进算法<sup>[5-6, 9, 17]</sup>中亦有所体现,其中 Lee 等<sup>[9]</sup>提出的增长立方体(Growing-cube)等值面抽取算法可视为其中的代表性方法,其后的诸多改进算法与这一方法思想极为相似。该方法把一个有等值面通过的立方体作为种子立方体,根据种子立方体邻接体元与等值面相交的状况逐渐进行增长,从而获得整个等值面。在进行增长的过程中,待增长的立方体通过建立基于边的 12 位表长的查找表获得,也避免了逐个遍历立方体体元,同时通过三角面片的合并减少了三角面片个数来提高重建效率。与这种方法相比,文中提出的基于邻接查找子表的改进 MC 算法存在以下不同:

1)需要构造等值面的邻接立方体的判断方式不同。Growing-cube 算法中判断方式是基于边的,它的判断方式与 256 种构型无关,不用考虑 15 种情况,只是根据两插值点(即三角面片一边的 2 个端点)所在的两边判断相邻立方体的情况。这种方法依赖于相交边索引及相交边的数目,尽管该方法未明确给出相交边索引的计算方法,算法实现中对于每一存在等值面的立方体,应采用实时计算相交边索引或引入已知的相交边查找表来确定相交边索引。由于实时计算相交边索引速度较慢,可利用一张  $256 \times 12$  大小的边查找表一次性获得当前立方体全体相交边索引值序列。但即使获得了这一索引值序列,为了判断某一面是否存在相邻立方体,该方法还需要进行  $12 \times 12$  次循环操作。由于该循环操作处于最内层循环体,会明显影响算法耗时。而文中算法与立方体体元中边的信息无关,也无需引入相交边查找表,针对 256 种构型分别建立其对应的相邻立方体情况,形成邻接查找表。对于每一存在等值面的立方体,文中算法利用构型索引,只需访问一次邻接查找表,即可获得所有的邻接立方体。邻接查找表固定且独立于具体数据,无需实时计算,使得文中算法在计算效率,临时存储量方面明显小于 Growing-cube 算法。

2)立方体体元信息的保存和遍历相邻立方体的内存开销上的不同。Growing-cube 算法中未说明立方体体元信息的保存方式,亦未说明利用堆栈方式建立立方体体元列表,对于标志变量的建立与初始化也未明确,增加了算法的不确定性。例如:如果一个立方体体元有 3 个相邻的体元立方体需要抽取

等值面,若不立即抽取这 3 个相邻的立方体的等值面,则必须保存它们的体元信息,以方便其后的等值面抽取处理。若把整个立方体的 8 个角点的所有信息都保存起来必然会造成很大的内存耗费。而文中算法根据数据特点,只将立方体体元中一个角点索引号压入栈中,从而节省了存储空间,也增加了算法清晰度与简洁度。

## 3 实验结果及分析

为验证文中方法的有效性,对人体脚部(Foot)、胸腔部位(Chest)、头部(Head)的 CT 数据集进行试验,其数据集分辨率分别为  $256 \times 256 \times 250$ ,  $512 \times 512 \times 313$ ,  $512 \times 512 \times 335$ 。对于不同数据集设置 5 个等值面阈值分别求取重建结果。为便于比较文中算法也实现了标准 MC 算法和改进的 Growing-cube 算法<sup>[9]</sup>,对于后者,考虑其合并三角面片的算法模块与文中讨论重点不一致,受参数影响较大,在比较中不予考虑。这 3 种算法均在 Visual studio 2010 平台下的 VTK (Visualization toolkit)与 MFC 集成环境中编程实现,计算机配置为 Intel(R) Pentium(R) Processor Duo 2.4 GHz, 1 G 内存;操作系统 Windows XP SP3。

### 3.1 在 VTK 下用文中算法进行三维重建

文中利用 VTK 平台下的显示功能对所提出的改进 MC 算法进行实验。VTK 是由 Kitware 公司开发的一套进行数据可视化的通用开发工具包,它在医学影像领域有着广泛应用,主要包括图像处理、计算机图形学、科学计算可视化,尤其是面绘制和体绘制等三维重建功能。VTK 中将各种各样的功能以类库形式存放起来,在具体应用中采用流水线(Pipeline)机制,几乎可以对任何类型的数据进行处理,并提供了许多相应的类对各种类型的数据进行转换或处理。

实验主要流程是首先利用类 vtkDICOMImageReader 将 DICOM 格式的 CT 断层图片读入,然后利用文中算法对给定的等值面阈值提取等值面,再利用 vtkPolyDataNormals 对得到的等值面进行处理,过滤掉过大的棱角以及不符合实际情况的三角面片,接着由 vtkPolyDataMapper 将其映射为几何数据,最后通过 vtkRenderer 和 vtkRenderWindow 将几何数据以一定的角度和色彩光照将等值面图像显示出来。若需要进行交互则通过 vtkRenderWindowInteractor 对用户的输入进行交互。不同算法中均设定相同的阈值与等值面插值

方式。

### 3.2 实验结果分析

图 4~图 6 为文中改进算法在 3 个数据集上根据不同等值面阈值抽取的等值面三维重建结果,可以看出算法不仅对于皮肤、骨骼等 CT 值较均衡的成分具有清晰完整重建效果,对于受造影剂影响的血管、心腔等软组织也能取得较好的视觉效果。由于文中算法和标准 MC 算法、Growing-cube 算法得到的三角面片集合相同,因此未列出另外 2 种算法的重建结果。

图 7 列出了文中算法在 3 个 CT 数据集上各有效立方体的平均邻接数目的出现频率,从中可以看出,无论是对表面比较简单的 Foot 数据集,还是对零散碎片较多的 Chest 数据集,以及对内部连通成分复杂的 Head 数据集,各有效立方体的相邻有效立方体个数的出现频率非常稳定,最可能的邻接立方体数目为 4,概率为 45%~50%,其次为 3 和 5,概率为 20%~30%以及 15%~25%,而需要判断 6 个邻接面的概率为 5%左右,远小于其他 3 种情形,这说明采用文中算法并不需要如标准 MC 算法那样判断所有 6 个邻接立方体,从而验证了文中算法通过约束搜索路径提高效率的可行性。

表 2 列出了不同等值面阈值参数下 3 个 CT 数据集中标准 MC 算法、Growing-cube 算法、改进算法访问的有效立方体数目,三角面片数目以及算法耗时。从中可以看出,对人体不同部位的 CT 数据集,尽管等值面阈值发生变化,但有效立方体数均远小于总体立方体数,不超过 5%,从统计意义上表明了文中算法避免访问空立方体的必要性。从访问立方体数目来看,文中算法与 Growing-cube 算法相同。表 3 对 3 种算法在 3 个 CT 数据集上的平均有效立方体数目、平均三角面片数目、平均耗时进行了比较。耗时部分括号内数据表示该算法相对于标准 MC 算法耗时的百分比。改进算法在每个数据集上都取得了最少的耗时,在非空立方体数目较多的 Chest 数据集上优势最为明显。总体而言,文中算法约为标准 MC 算法的 79.03%,为 Growing-cube 算法的 90.88%。文中算法及 Growing-cube 算法耗时小于标准 MC 算法的根本原因在于算法将搜索路径限制在有效立方体中,避免了对大量非空立方体的无效处理;而文中算法耗时少于同是基于局部邻接信息的 Growing-cube 算法的原因在于定位邻接

立方体过程中避免了新的循环查找,这从实验上验证了前文 2.3 节的分析。

改进算法中遍历的立方体数减少 95%,而算法的运行时间却没有相应缩短 95%。发生这种情况的主要原因有两方面,一方面是虽然避免了立方体的遍历操作,但在查询查找表和堆栈建立、压入和释放的过程中会占用额外时间,在邻接立方体的联合判断过程会增加耗时;另一方面是内存申请和其它处理也会占用一定时间,两者综合起来使得实际运行效率的提升在 20%左右。为了进一步明确文中算法中不同部分的耗时比例,将文中算法耗时大致分为三角面片生成、堆栈处理及邻接立方体判断、初始化及其他 3 个部分,对应地也将标准 MC 算法耗时分为三角面片生成、遍历及非空立方体判断、初始化及其它等 3 个部分,然后在实验中记录各部分的耗时,结果如表 4 所示。从中可以看出,2 种算法各部分的耗时比重相差很大。对于 MC 算法,遍历及非空立方体判断部分的平均耗时最多,占总体耗时的 50.88%,三角面片耗时占总体耗时的 45.77%;对于文中算法初始化处理单一,耗时占比最少,虽然避免了遍历及非空立方体判断的耗时,但增加了堆栈处理及邻接立方体查找的耗时,这部分耗时占总体耗时的 27.85%,在三角面片生成的耗时上,2 种算法的平均耗时极为相似,均占据了较大的比例,故不能只以遍历数目的比例来看待文中算法的效率提升,而应综合各部分耗时来进行评判;此外文中算法的初始化及其他部分的耗时从绝对时间和所占比例方面也显著大于标准 MC 算法,这主要是因为文中算法需要对标志数组及其他成分进行初始化,而标准 MC 算法没有这类初始化处理。因此,文中算法总体上未随着非空立方体比例而同比例缩短耗时。需要说明的是,由于 MC 算法自洽性强,结构紧凑,文中对算法耗时的分类侧重于大体,未严格按逐条语句的功能分类。此外,文中算法实现中仅集中于实现算法思想,未对算法代码进行优化。由于算法实现软硬件平台原因,算法重建时间总体偏长,而所有算法在同一平台下进行开发与比较,这不影响实验的主要结论。综上所述,文中改进算法在空间和时间上都明显减小了开销,更便于 MC 算法面绘制时的实时人机交互。并且由于邻接查找子表固定且独立于图像数据,可方便地与其他基于 MC 的算法进行集成。

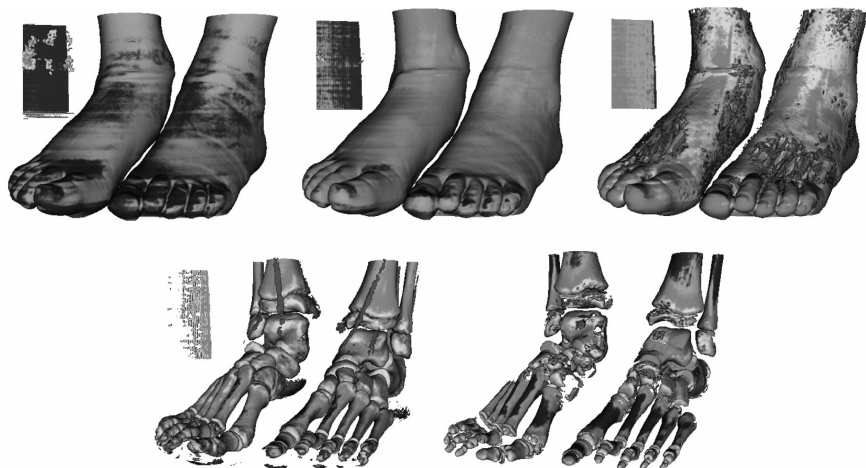


图 4 Foot CT 数据集三维重建结果(等值面阈值  $k = -275, -125, 25, 175, 325$ )

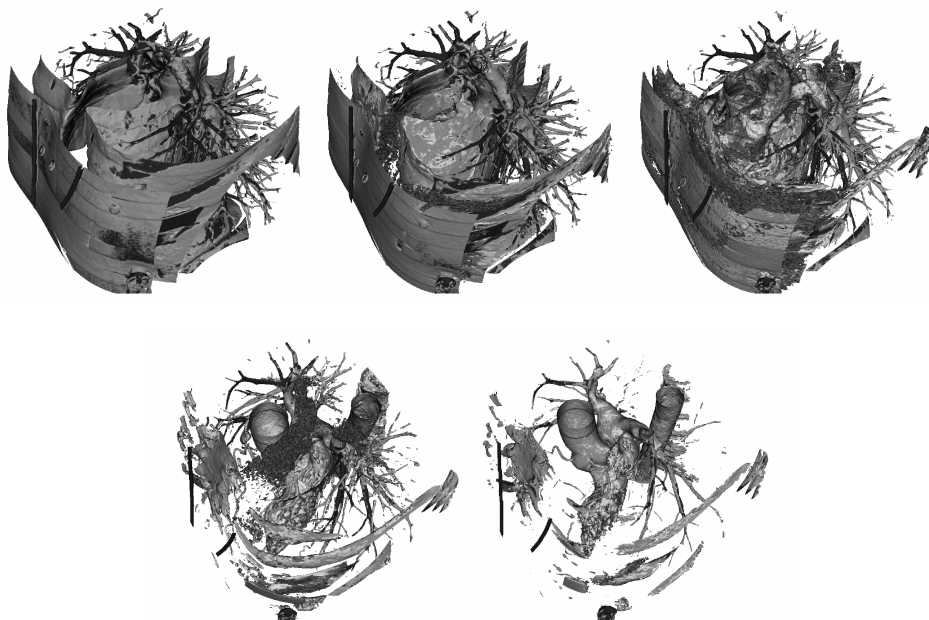


图 5 Chest CT 数据集三维重建结果(等值面阈值  $k = -275, -125, 25, 175, 325$ )

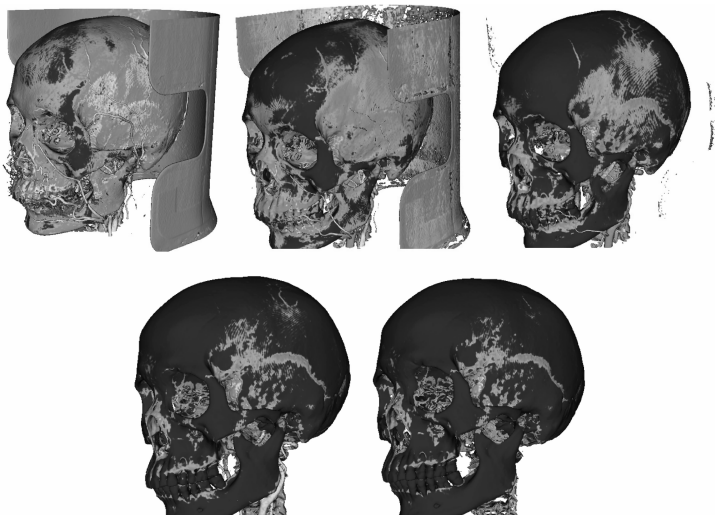


图 6 Head CT 数据集三维重建结果(等值面阈值  $k = 125, 225, 325, 425, 525$ )



表 2 CT 数据集上不同算法的重建耗时对比

| CT 数据集                 | 等值面 $k$ | 非空立方体数    | 三角面片数     | 重建时间/s |              |       |
|------------------------|---------|-----------|-----------|--------|--------------|-------|
|                        |         |           |           | MC     | Growing-cube | 文中算法  |
| Foot<br>(512×512×250)  | -275    | 601 303   | 1 197 266 | 5.13   | 4.07         | 3.47  |
|                        | -125    | 596 600   | 1 184 806 | 5.11   | 4.04         | 3.45  |
|                        | 25      | 1 936 240 | 3 745 315 | 21.07  | 19.13        | 18.92 |
|                        | 175     | 921 483   | 1 805 179 | 11.71  | 10.1         | 8.97  |
|                        | 325     | 662 547   | 1 312 586 | 6.16   | 4.65         | 4.02  |
| Chest<br>(512×512×313) | -275    | 1 989 726 | 3 949 684 | 22.38  | 19.16        | 17.36 |
|                        | -125    | 2 814 752 | 5 470 079 | 32.53  | 29.17        | 25.92 |
|                        | 25      | 4 344 726 | 8 548 422 | 50.85  | 43.41        | 37.79 |
|                        | 175     | 1 963 570 | 3 807 947 | 23.02  | 20.08        | 18.42 |
| Head<br>(512×512×335)  | 125     | 2 699 219 | 5 307 355 | 30.31  | 27.32        | 24.90 |
|                        | 225     | 2 338 731 | 4 647 127 | 26.21  | 22.75        | 20.84 |
|                        | 325     | 1 664 911 | 3 315 476 | 19.28  | 17.09        | 15.54 |
|                        | 425     | 1 657 860 | 3 315 559 | 19.22  | 17.02        | 15.37 |
|                        | 525     | 1 656 095 | 3 319 086 | 19.08  | 16.87        | 15.35 |

表 3 不同算法的平均重建耗时对比

| 数据集   | 全体立方体数          | 平均非空立方体数(占总数百分比) | 平均三角面片数     | 平均重建时间/s |                              | 比率<br>文中方法(占 MC 方法百分比) |
|-------|-----------------|------------------|-------------|----------|------------------------------|------------------------|
|       |                 |                  |             | MC       | Growing-cube<br>(占 MC 方法百分比) |                        |
| Foot  | 65 019 129 943  | 634.6(1.45%)     | 1 849 030.4 | 9.836    | 8.398(85.38%)                | 7.766(78.95%)          |
| Chest | 81 469 7522 474 | 201.6(3.04%)     | 4 846 359.8 | 29.006   | 25.220(86.95%)               | 22.502(77.58%)         |
| Head  | 87 214 4142 003 | 363.2(2.30%)     | 3 980 920.6 | 22.842   | 20.232(88.57%)               | 18.400(80.55%)         |

表 4 不同算法的耗时成分比较

| 数据集   | MC 算法/s             |                    |                  | 文中算法/s             |                       |                   |
|-------|---------------------|--------------------|------------------|--------------------|-----------------------|-------------------|
|       | 遍历及非空立方体判断(占总耗时百分比) | 三角面片生成(占总耗时百分比)    | 初始化及其它(占总耗时百分比)  | 三角面片生成(占总耗时百分比)    | 堆栈处理及邻接立方体判断(占总耗时百分比) | 初始化及其它(占总耗时百分比)   |
| Foot  | 4.979<br>(50.62%)   | 4.487<br>(45.62%)  | 0.370<br>(3.76%) | 4.493<br>(57.85%)  | 2.148<br>(27.66%)     | 1.125<br>(14.49%) |
| Chest | 14.805<br>(51.04%)  | 13.161<br>(45.37%) | 1.04<br>(3.59%)  | 13.196<br>(58.64%) | 6.238<br>(27.72%)     | 3.068<br>(13.63%) |
| Head  | 11.647<br>(50.99%)  | 10.582<br>(46.33%) | 0.613<br>(2.68%) | 10.726<br>(58.29%) | 5.185<br>(28.18%)     | 2.089<br>(11.35%) |

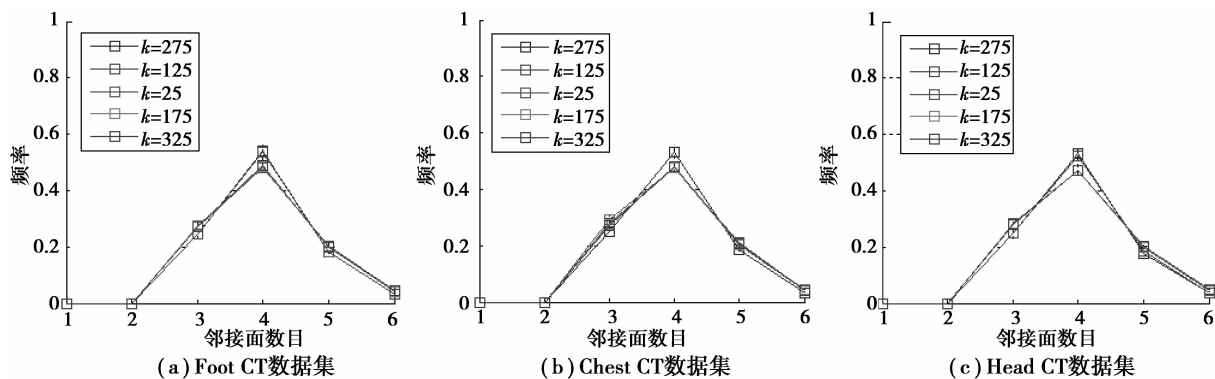


图 7 Foot, Chest, Head CT 数据集上邻接面数目的出现频率

## 5 结 语

文中提出了融合构型查找表与邻接查找子表的 MC 算法,针对传统 MC 算法较低的重建效率,对标准 MC 算法进行了改进,通过显性构建邻接查找子表来定位相邻体元,只判断有等值面通过的那些立方体体元。利用可视化工具 VTK 对人头、心脏和脚部的三维重建实验得知,文中算法在没有损失重建质量与增加临时存储量的前提下,显著减少了立方体体元访问数目,提高了重建效率。

### 参考文献:

- [1] Newman T S, Yi H. A survey of the marching cubes algorithm[J]. Computers & Graphics, 2006, 30(5): 854-879.
- [2] Lorensen W E, Cline H E. Marching cubes: a high resolution 3d surface construction algorithm[J]. ACM Computer Graphics, 1987, 21(4): 163-169.
- [3] Dietrich C A, Scheidegger C E, Comba J L D, et al. Marching cubes without skinny triangles [J]. Computing in Science and Engineering, 2009, 11(2): 82-87.
- [4] Liu S J, Li J. Preserving zeros in surface construction using marching cubes[J]. Machine Graphics & Vision International Journal, 2010, 19(1):97-123.
- [5] 刘少丽, 杨向东, 陈恩. 基于分割 MC 算法的超声影像三维重建方法[J]. 清华大学学报: 自然科学版, 2010, 50(8): 1214-1218.  
LIU Shaoli, YANG Xiangdong, CHEN Ken. Segmentation MC algorithm-based method for ultrasound image 3-D reconstruction [J]. Journal of Tsinghua University: Science & Technology, 2010, 50(8): 1214-1218.
- [6] Nielson G M, Zhang L Y, Lee K, et al. Spherical parameterization of marching cubes isosurfaces based upon nearest neighbor coordinates [J]. Journal of Computer Science and Technology, 2009, 24(1): 30-38.
- [7] Montani C, Scateni R, Scopigno R. Discretized marching cubes [C] // Proceedings of the IEEE Visualization, October 17-21, 1994, Washington DC, USA. Piscataway: IEEE Press, 1994: 281-287.
- [8] Dyken C, Ziegler G, Theobalt C, et al. High-speed marching cubes using histogram pyramids [J]. Computer Graphics Forum, 2008, 27(8): 2028-2039.
- [9] Lee T Y, Lin C H. Growing-cube isosurface extraction algorithm for medical volume data [J]. Computerized Medical Imaging Graphics, 2001, 25(5): 405-415.
- [10] Vignoles G L, Donias M, Mulat C, et al. Simplified marching cubes: an efficient discretization scheme for simulations of deposition/ ablation in complex media [J]. Computational Material Science, 2011, 50(3): 893-902.
- [11] 刘海华, 万双, 陆雪松. 基于 Marching Cubes 算法的多层医学图像重建实现[J]. 中南民族大学学报: 自然科学版, 2009, 28(3): 79-84.  
LIU Haihua, WAN Shuang, LU Xuesong. The Multi-storey medical image reconstruction based on marching cubes algorithm [J]. Journal of South-Central University for Nationalities: Natural Science Edition, 2009, 28(3): 79-84.
- [12] Nielson G M. On marching cubes [J]. IEEE Transactions on Visualization and Computer Graphics, 2003, 9(3): 283-297.
- [13] Dietrich C A, Scheidegger C E, Schreiner J M, et al. Edge transformations for improving mesh quality of marching cubes [J]. IEEE Transactions on Visualization and Computer Graphics, 2009, 15(1): 150-159.
- [14] Gelder A V, Wilhelms J. Topological considerations in isosurface generation [J]. ACM Transactions on Graphics, 1994, 13(4): 337-375.

- Communication Conference, February 2-7, 1992, San Jose, California. Piscataway: IEEE Press, 1992(5): 219
- [11] Bogatyryov V A, Dianov E M, Romyantsev S D, et al. Technical digest postconference edition: trends in optics and photonics[C] // Proceedings of the Optical Fiber Communication Conference, February 21-26, 1993, San Jose, California. Piscataway: IEEE Press, 1993(14): 78.
- [12] 李国英. 表面工程手册[M]. 北京:机械工业出版社,2001.
- [13] Li Y L, Hua Z, Yan F, et al. Metal coating of fiber Bragg grating and the temperature sensing character after metallization [J]. Optical Fiber Technology, 15(4):391-397.
- [14] Sandlin S, Kinnunen T, Rämö J, et al. A simple method for metal re-coating of optical fibre Bragg gratings[J]. Surface and Coatings Technology, 2006 (201):3061-3065.
- [15] Jiang B Q, Huang Q R, Peng J, et al. Stimulation of rare earths in quartz optical fiber surface ,metallization by electroless plating and electroplating[J]. Journal of Rare Earths, 2007(S1):426-431.
- [16] Huang L, Wang Z F, Li Z M, et al. Electroless nickel plating on optical fiber probe [J]. Chinese Optics of Letters, 2009, 7(6):472-474.
- [17] Li C A, Xu L N, Gu N. Electroless silver plating for metallization of near-field optical fiber probes [J]. Chinese Optics Letters, 2007, 5(10):594-595.
- [18] 谢剑锋, 张华, 徐建宁, 等. 智能金属结构中光纤表面金属化的研究[J]. 材料保护, 2007,40(8):40-42,68. XIE Jianfeng, ZHANG Hua, XU Jianning, et al. Optimization of the process for electroless nickel plating of optical fiber[J]. Materials Protection, 2007,40(8): 40-42,68.
- [19] 李宁. 化学镀实用技术[M]. 北京:化学工业出版社,2004.

(编辑 陈移峰)

~~~~~

(上接第 77 页)

- [15] Liang P, Dong B, Yu X, et al. Prognostic factors for percutaneous microwave coagulation therapy of hepatic metastases[J]. American Journal of Roentgenology, 2003, 181(5): 1319-1325.
- [16] Lempitsky V. Surface extraction from binary volumes with higher-order smoothness[C] // Proceedings of the 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2010), June 13-18, 2010, San Francisco, CA, USA. Piscataway: IEEE Press, 2010: 1197-1204.
- [17] Feltell D, Bai L. A new marching cubes algorithm for interactive level set with application to MR image segmentation[C] // Proceedings of the 6th International Conference on Advances in Visual Computing (ISVC'10), November 29-December 1, 2010, Las Vegas, NV, USA. Berlin: Springer, 2010: 371-380.

(编辑 陈移峰)