医学图像三维重建及实时性研究

杜俊俐[1,2],黄心汉[1],郭清宇[2]
DU Jun-li[1,2],HUANG Xin-han[1],GUO Qing-yu[2]

1.华中科技大学 控制科学与工程系,武汉 430074
2.中原工学院 计算机科学系,郑州 450007
1.Department of Automatic Control,Huazhong University of Science and Technology,Wuhan 430074,China
2.Department of Computer Science,Zhongyuan Institute of Technology,Zhengzhou 450007,China

DU Jun-li,HUANG Xin-han,GUO Qing-yu.Study on real-time reconstruction of medical images.Computer Engineering and Applications,2007,43(19):206-209.

Abstract: A 3D reconstruction system is developed with VTK and MC.Since rendering or storage cost of object is proportional to the number of triangles in mesh,too complex and detailed model will cause system overload in storage,transmission,computing and real-time rendering of mesh.Merging vertexes can reduce the number of triangles in mesh and hash mapping of the mesh can reduce over-duplicated storage of vertexes.

Key words: 3D reconstruction;real time;mesh simplification;vertexes merging;hash mapping

摘 要:使用 VTK 工具包和 MC 重建算法开发了一套医学影像三维重建系统,并着重对实时性进行研究。由于重建后图像数据的处理时间和存储代价与三角形网格中三角形数量成正比,过于复杂和细节化的网格会给图像数据的存储、传输、计算和实时绘制等带来负担,故采用了顶点合并的三角形网格简化方法来减少三角形数量。另外,网格存储中存在公共顶点的大量重复存储,故提出了三角形网格的哈希映射存储方法,消除了顶点的重复存储。

关键词:三维重建;实时性;网格简化;顶点合并;哈希映射
文章编号:1002-8331(2007)19-0206-04 **文献标识码**:A **中图分类号**:TP391.41

1 引言

三维重建是计算机可视化的重要研究领域。医学图像的三维重建在医学界有着广泛的应用。国外学者在这方面已做了许多工作,Lorensen 等[1]于 1987 年提出 Marching Cube(MC)算法;Nielson 等[2]提出 Marching Tetrahedra(MT)算法;另外还有 Delaunay 四面体化[3]。国内也有许多学者作了很多有益的研究。一般将三维重建方法划分为面绘制(Surface Rendering)和体绘制(Volume Rendering)两大类。面绘制因其绘制速度优于体绘制而在实际系统中采用较多。

在面绘制方法中用三角形网格来重建三维实体表面的 MC 算法是典型的代表。VTK(Visualization Toolkit)是可视化领域最负盛名的软件开发包, 可读取 DICOM 文件、显示 3D 图像、还封装了 MC 算法以及常用的图像处理算法等,是医学图像三维重建系统理想的开发工具。故本文采用 VTK 和 MC 算法构建了一个跨平台可扩展的医学图像三维重建平台。

三维重建系统的突出问题是时空问题。一方面,MC 算法产生的三角形网格中三角形数量巨大,而计算机存储和处理网格模型的代价和模型中三角形的数量成正比,庞大的网格数据对计算机的容量和处理能力等提出较高要求, 给模型的存储、传输、计算、绘制等带来了困难。而网格简化可以在基于视觉误差度的前提下减少三角形数量。另一方面,三角形网格中广泛存在的三角形顶点共享造成了顶点的大量重复存储,也消耗了大量的内存空间。空间问题是重建后图像的存储、传输、计算和实时绘制等的障碍,因此本文从三角形网格简化和改善存储结构两方面进行 3D 模型的空间优化。

2 医学图像的三维重建

MC 算法实现三维重建的基本原理是:在三维数据场中构造出等值面并找出经过该等值面的体元(Cubes)后,通过线性插值的方法确定三角片顶点的位置,用大量的三角片表示等值面,然后用三角形网格来重建三维表面。三维重建功能的处理流程如图 1 所示。

图 1 三维重建流程图

首先读取 DICOM 格式的序列图像文件, 然后切割图片为小立方体表达的体元集合,再用 MC 算法获取体元中所包含的等值三角形面并构造出三角形网格,最后对三角形网格表达的

基金项目:河南省科技公关计划(the Key Technologies R&D Program of Henan Province of China under Grant No.0624220100)。
作者简介:杜俊俐(1964-),女,副教授,博士生,研究方向:图形图像处理;黄心汉(1946-),男,教授,博士生导师,研究方向:控制理论及机器人等;郭清宇(1959-),男,教授,研究方向:图像处理与模式识别。

三维实体进行存储和绘制。

2.1 DICOM 格式文件的读取

医学图像的存储和交流格式标准为 DICOM，故系统支持该标准,以支持对重建前图像的浏览,综合重建前的二维图像和重建后的三维图像进行医学分析。DICOM 文件扩展名为 DCM,VTK 可直接读取该格式文件。实现读取一个文件夹内的 DICOM 文件的代码如下:

```
vtkDICOMImageReader * m_dcmreader;
m_dcmreader->SetDirectoryName(sdir);
m_dcmreader->Update();
```

该部分被封装为 Reconstruction 类的一个成员函数 SetDCMDirectory(char* sdir),其中参数 sdir 为 DICOM 文件的文件夹路径。

另外,因许多图像处理软件不支持 DICOM 格式,故系统还实现了 DICOM 文件到 BMP 文件的转换,以便为其他图像处理软件提供医学影像资料。系统还实现了多帧 DICOM 文件拆分为多个单帧 DICOM 文件的功能，以方便对单帧图像的操作和分析。

2.2 切割图像序列为体元集合

MC 算法的输入是体元集合,所以,在 MC 算法之前需先对图像集进行体元集的分割。而在切割之前又需要先设定体元的边长和图像间距。CT 图像的采集层厚一般在 1 mm~10 mm,故将默认值设为 8。体元边长越大重建速度越快,但重建结果越粗糙,故应综合考虑重建时间和质量需求选取合适的体元边长。

体元切割过程如下:

(1)把所有图像按照已知的断层扫描间隔依次排列。假定扫描足够致密,间隔足够小,那么这个排列就可以看作是一个三维实体。

(2)根据速度与精度的需要,合理选择立方体的 X 和 Y 间隔(Z 间隔是断层间隔)等分三维实体,并用 8 组坐标表示一个立方体。

(3)通过以上方式取得所有立方体的顶点坐标,存储于立方体结构数组中,这便是 MC 算法的输入数据。

(4)MC 算法需要的另一个参量是决定在面上还是面下的灰度值,这个值的输入根据图像的曝光度来确定。

具体实现:unsigned int RC::Reconstruction::Dismember();返回分割后的体元数目。

2.3 MC 重建算法的实现

MC 算法的输入是前面切割出的体元集合,输出是由三角形面片表达实体的三维网格。VTK 中封装了 MC 算法,重建时直接调用 MC 重建函数即可。

实现 MC 算法的函数原型是:

```
int Polygonise(GRIDCELL& grid,double isolevel,TRIANGLE*triangles)
```

其中，参数 grid 为输入的体元,isolevel 是等值面的值,triangles 是用来存放生成的三角形面片的缓冲区。

单值三维重建能重点考察某组织构成的器官形态,比如大脑的骨骼;而多值三维重建能表现实体的更多细节,比如大脑的皮肤、骨骼、脑组织等。多值三维重建是根据所给的多个等值面的值进行多次调用重建函数来获得。

重建结果被保存到一个 PolyDataColl 的 vector(STL 中的一个高效的智能数组)中,通过 GetOutput()即可获取重建结果。

重建的控制是在 Reconstruction::Update(double isolevel)中。该函数遍历已切割好的体元,传送给重建关键函数并对返回的三角形面片进行处理，使之成为索引表示的三角形网格;然后把等值面的值、顶点的数组和索引三角形的数组一起作为 PolyData 结构存放到由该结构组成的数组中。

2.4 重建后数据的封装与存取

对重建结果进行存储,以便以后直接读取以实现重建后图像的快速显示,并支持重建结果的共享和交流。重建结果以二进制文件形式保存，该文件由文件头和数据体两部分组成,其结构如表 1 所示。其中,前五项构成文件头,后面的项构成数据体。

重建结果文件的存取及显示由如下函数来实现:成员函数 SaveFile 实现数据的存储，成员函数 LoadFile 实现重建文件的读取,成员函数 FromReconstruction 从 Reconstruction 结构获取信息,成员函数 GetOutput 向显示设备提供显示数据。

2.5 三维图像的绘制

实现重建后三维图像的显示，首先是从 vtkPolyDataAlgorithm 派生出 MeshReader 类,每个不同的等值面会被输入到不同的 MeshReader 中进行处理,包括求法线向量、设定材质等。重载其 RequestData 方法,并用 VTK 间接调用此方法来获得三维数据。

为防止重建后图像显示时出现“鱼鳞现象”,必须加入顶点的法线向量,再采用高洛德着色(Gouraud Shading)算法,最后通过差值算法得到三角形内部各点的颜色,达到一种平滑过渡的显示效果。

(1)法线向量的求取

一个顶点的法线向量是共享该点的三角形的法线向量的平均值。三角形的法向量通过将三角形的两条边的向量点叉乘然后单位化来获得。在数万个三角形中查找相同顶点的三角形,对时间的消耗是相当巨大的,使用哈希映射中的一对多映射(hash_multimap)来实现搜索可大大节省时间,其时间复杂度是常量。通过 hash_multimap 遍历一次三角形之后,所有的共点三角形就会自动分类到各自的顶点,同时生成三角形到其各自向量的映射(hash_map)。显然一个三角形会在 multimap 中出现 3 次,遍历 hash_multimap 的键(key),就可读出它所对应的多个三角形的地址，然后查询映射就可得到三角形的法向量,再对这些三角形的向量求和并单位化就得出了所需顶点的法向量。把计算出来的法线向量加入模型信息中，开启 OpenGL 的 Gouraud Shading 实现平滑的颜色效果。

(2)RequestData 方法的重载

通过继承重载 vtkPolyDataAlgorithm 提供给 VTK 的图形引擎三维数据,重载其 RequestData 方法,提供渲染器模型数据。过程是，先对输入的 PolyData 进行处理，转化为 VTK 专用 vtkPolyData,然后计算法向量并传递给 vtkPolyData。

(3)重建后图像的显示

重建后图像的显示通过函数 void ShowMesh(MeshInfo &mi)实现。它首先定义一个 MeshReader 指针的动态数组,遍历从 MeshInfo 中得到的 PolyDataColl(不同等值面的三维数据集

表 1 重建后模型存储的文件结构

宽度	高度	深度	最小像素值	最大像素值	PolyDataColl.size()	PolyData	PolyData	PolyData	……

合),对每个 PolyData 都生成一个 MeshReader,并通过 vtkActor 来呈现三维图像。同时根据 PolyData 中的等值面的值确定其材质的颜色,然后添加到渲染器 vtkRenderer 中,通过 vtkRenderWindowInteractor 来实现鼠标和键盘的控制,实现对图像进行交互方式下的旋转和缩放。

3 三角形网格的空间优化

由于先进的医学设备测量得到的数据分布十分密集,由此建立的三维网格模型通常是由几十万个、几百万个甚至上亿个三角面片组成来刻画模型细节以逼近现实的物质形状。虽然过于庞大的模型可以使实体的表达更加精确,但却很不实用,导致了实体模型过于复杂,存储时占用空间太大,绘制时耗费时间太长。解决该问题的方法之一是提高绘制机器的性能,但这在微机普及的今天并不能被广泛接受和使用,而在保持模型精度的同时合理简化模型便成为一个重要课题。

模型的简化是指采用适当的算法减少表示物体模型的面片数、边数和顶点数。其目的就是在保证对原有的模型具有良好的形状逼近的前提下,尽量减少用于表示该模型的三角形的数目。

国外对网格简化的研究已有许多成果。Schroeder 提出了基于顶点删除的网格删减方法[4],Turk 给出了基于重新划分的模型简化方法[5],Hoppe 采用了能量函数最优化的网格简化方法[6],Eck 将小波技术用于模型简化[7],Hamann 给出了一种基于三角形的曲率计算来移去三角形的模型简化方法[8],Isler 使用边折叠和三角形折叠操作来实时生成简化模型[9],Garland 利用二次误差方法来控制网格简化[10]。国内在这方面也开展了一些卓有成效的研究工作。

在众多的网格简化算法中,以二次误差测度为基础的边折叠的简化算法在时间复杂度和简化质量两方面都有较为出色的表现,得到了广泛的认同和研究,所以本文也采用以此为基础的顶点合并的网格简化方法。另外,由于网格的连续性,每个顶点均为若干三角形的公共顶点,这些公共顶点的大量重复存储也为网格存储增加了额外的负担。所以,对三角网格模型进行空间优化主要有以下两个途径:

(1)对三角网格进行简化,减少三角形数量;

(2)改善三角形网格的存储结构,消除公共顶点的重复存储。

3.1 三角形网格的简化

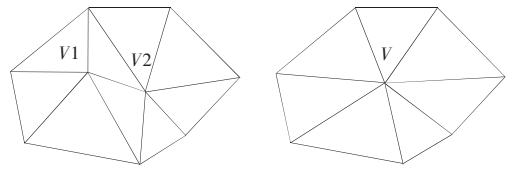
网格简化是对输入网格模型产生与其对应的、基本保持形状的、较粗糙的简化模型的过程。输出的简化模型应该满足某些预先给定的条件,如新旧模型间的某种误差阈值,以保证其逼近程度。

顶点合并(Vertex Merging)是通过合并三角形顶点来减少网格中三角片数量,提高计算机处理复杂模型速度的有效方法。顶点合并算法的基本原理是投影到图像空间中足够小的区域内的一组顶点可以用一个代表性的顶点来取代,合并形成的新顶点又可参与下一轮合并。如果每次合并一个三角形的 3 个顶点,称为三角形收缩;如果每次只合并一条边的两个顶点,则叫边收缩(如图 2)。

本文采用边收缩的方法进行顶点合并。算法的主要思想是把两个三角形的公共边收缩到一个点上,同时删除共享该边的三角形。这里的两个关键问题是:

(1)满足什么条件的边才能把两个端点合并;

(2)合并生成一个新点的位置如何确定。



(a)$V1V2$ 边收缩前 (b)$V1V2$ 边收缩到 V 点后

图 2 边收缩示意图

3.1.1 顶点合并的约束条件

网格模型简化应优先删除一些不重要的特征。大多数光照模型的真实感显示都要涉及物体表面的法向量,表面法向量会在模型简化后发生变化,使模型的三维立体视觉效果受到影响。因此,物体的顶点和面的法向量的变化比物体顶点和面的位置变化对视觉的影响还要大。所以,一般认为短小的边、平坦的区域应该优先简化合并。故采用法向约束和距离约束两个约束条件来确定三角形中需要合并的顶点[11]。其中,法向约束确定平坦的区域,距离约束确定短小的边。

具体约束条件定义如下:

(1)法向约束:若将三角形两顶点 Vi、Vj 的法向量夹角记为 $\alpha(Vi,Vj)$,设定一个最小角度约束 MAC,则法向约束公式为:$\alpha(Vi,Vj)\geqslant MAC$。其中,

$$\alpha(Vi,Vj)=\arccos\left[\frac{\boldsymbol{Ni}\cdot\boldsymbol{Nj}}{|\boldsymbol{Ni}||\boldsymbol{Nj}|}\right]$$

$\boldsymbol{Ni}$、$\boldsymbol{Nj}$ 分别指顶点 Vi、Vj 的法向量。

(2)距离约束:若将三角形顶点 Vi、Vj 之间的几何距离记为 $d(Vi,Vj)$,设定一个最大距离约束 MDC,则距离约束公式为:$d(Vi,Vj)\leqslant MDC$。

对三角形顶点进行判断,当两个顶点满足上述两个约束条件之一时,则将两个顶点合并。两个顶点组成的边收缩成一点,包含该边的所有三角形被删除。若条件不满足,则保留三角形,继续判断下一个三角形顶点情况。

3.1.2 合并顶点位置的确定

在采用边收缩操作的简化算法中,选取收缩点坐标的原则是使简化后的网格形状尽可能接近原始网格。一般选择收缩点坐标的策略主要有如下两个。

(1)子集位移法

子集位移法是把收缩边的一个端点收缩到另一个端点的方法。一种最简单的实现方法是:不做任何计算判断简单地指定将 $V1$ 收缩到 $V2$ 或将 $V2$ 收缩到 $V1$。该法处理时间最短但结果可能不是最优。另一种确定收缩点位置的方法是:根据两顶点的二次误差测度 $Q(V1)$和 $Q(V2)$的大小在 $V1$、$V2$ 之间进行选择。其中 $Q(V)$被定义为点到集合中所有平面距离的平方和。子集位移法所得到的任何收缩点均属于原始模型的顶点集。

(2)优化选择法

收缩点更好的选择并非局限于边 $V1V2$,甚至该局部网格的表面。可求取使 $Q(V)$最小的 V 作为收缩点的坐标。最小值应位于 $\frac{\partial Q}{\partial x}=\frac{\partial Q}{\partial y}=\frac{\partial Q}{\partial z}=0$ 处,求解该等式,即得最佳收缩点

位置。

本系统中,在兼顾处理时间复杂度和简化质量两个指标的前提下,设计了一种折中的确定收缩点位置的方法,即在两端点及其中点三者之间进行收缩点的选择。

3.1.3 实验结果

图 3 给出了一组系统的输出实例。从视觉上看,模型在简化 50%后与原模型还很接近。

(a)初始模型 (b)50%的简化模型

(顶点数 9 999,面片数 19 994) (顶点数 4 999,面片数 9 994)

图 3 头颅网格模型的简化

3.2 顶点的索引表示和哈希映射

由于等值面的连续性,每个顶点都被若干个三角形所共享,如图 4 中顶点 3 被 5 个三角形所共享,顶点 1、2、4、5、6 分别被 2 个三角形所共享。这些顶点的大量重复为实体的存储和绘制增加了额外的负担。顶点的索引表示法可以解决该问题。

三角形顶点的索引表示是将顶点和三角形的表示分开,把所有顶点存放到一个数组中,三角形的每个顶点用它在顶点数组中的索引表示,如图 5 所示。以图 4 为例,其中包含 6 个顶点和 5 个三角形,假设索引值和坐标值均占用 2 Byte,则直接存储需空间 5*3*3*2=90 Byte,索引存储需空间 6*3*2+5*3*2=66 Byte。显然,三角形顶点的索引表示能减少存储空间,甚至比化简三角形网格还要显著,且不丢失任何网格信息。

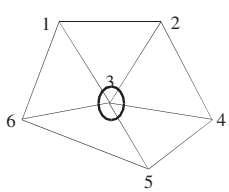


图 4 三角形的公共顶点

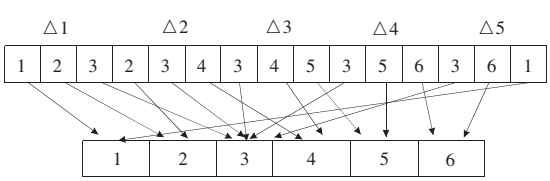


图 5 图 4 网格的索引存储结构示意图

虽然索引表示减少了存储空间,但它又引入了另一个问题,即对于新得到的顶点,检测它是否已经存在于数组,又增加了时间消耗,粗略估算为(1+n)*n/2(n 为含重复顶点的顶点个数)。所以,顶点搜索时应采取一定的搜索算法来优化点搜索速度。此处借助 STL 中的哈希映射(hash_map),使时间复杂度接近 1,达到了理想的点分析搜索速度。

具体方法如下:

(1)因普通对象不能作为 hashmap 的 key(关键码),故在数据定义时,重载了点结构的(int)强制转换方法,使 hashmap 可以用点的对象计算出散列值。

(2)定义哈希映射为点的结构(XYZ)到整型(int)的映射。

```
typedef std::hash_map<XYZ,int>PointMap;
PointMap ptm;  //点到索引的映射
```

(3)添加新点前,先使用映射的 find 方法查找顶点是否已存在。

```
PointMap::iterator pmpos=ptm.find(tg[i].p[j]);
```

(4)如果该点不存在则插入新的顶点且索引递增 1,并把索引值赋给三角形的相应顶点;如果该点已存在,则将找到的该点索引赋给三角形的相应顶点。

4 结束语

本文所实现系统能对 CT、MRI 等序列图像进行三维重建。未经化简的重建系统因处理时间过长无法实际使用。经简化和存储优化后,三维图像的存储空间和绘制时间大大减少。但因三角形网格化简会使首次重建时间比 MC 算法稍长,实验表明,增加了约 31%~37%。经化简后三角形数量相对于 MC 算法减少了 46%~66%,网格的索引存储结构使存储量减少 60%~80%,后续绘制处理时间减少了 47%~67%。

(收稿日期:2007 年 3 月)

参考文献:

[1] Lorensen W E,Cline H E.Marching cubes:a high resolution 3D surface construction algorithms[J].Computer Graphics,1987,21(4):163-169.

[2] Nielson G M,Sung J.Interval volume tetrahedronization[C]//Proceedings of the 8th IEEE Visualization'97 Conference,Phoenix,1997:221-228.

[3] Golias N A,Tsiboukis T D.An approach to refining three-dimensional tetrahedral meshes based on Delaunay transformation[J].International Journal for Numerical Methods in Engineering,1994(37):793-812.

[4] Schroeder W J,Zarge J A.Decimation of triangle meshes[J].Computer Graphics,1992,26(2):65-70.

[5] Turk G.Re-tiling polygonal surface[J].Computer Graphics,1992,26(2):55-64.

[6] Hoppe H,De Rose T.Mesh optimization[J].Computer Graphics,1993,27(1):19-26.

[7] Eck M,De Rose T.Multi resolution analysis of arbitrary meshes[J].Computer Graphics,1995,29(2):173-182.

[8] Hamann B.A data reduction scheme for triangulated surfaces[J].Computer Aided Geometric Design,1994,11(3):197-214.

[9] Islet V,Lau R W H.Green mark real-time multi-resolution modeling for complex virtual environments[C]//Proc of VRST'96,Hongkong,China,1996:11-19.

[10] Garland M,Heckbert P S.Surface simplification using quadric error metrics[J].Computer Graphics,1997,31(3):209-216.

[11] 李华,蒙培生,王乘.医学图像重建 MC 算法三角片的合并与实现[J].计算机应用,2003,23(6):104-106.