递进网格算法在虚拟内窥镜中的应用*

曹 勇, 田 捷, 张晓鹏, 邱 峰

(中国科学院 自动化研究所 人工智能实验室,北京 100080)
E-mail: tian@dr.com; zxp@dr.com
http://www.3dmed.net

摘要: 采用最新提出的递进网格算法来解决虚拟内窥镜系统的关键问题.讨论了递进网格算法在数据结构、数据简化和漫游上的优点,接着分析了其在虚拟内窥镜系统中应用的特殊性,也提出了一些局限性.提出了改进的递进网格算法,并给出算法的实现和结果.改进的算法较好地解决了数据简化、基于 LoD 的漫游等核心问题.

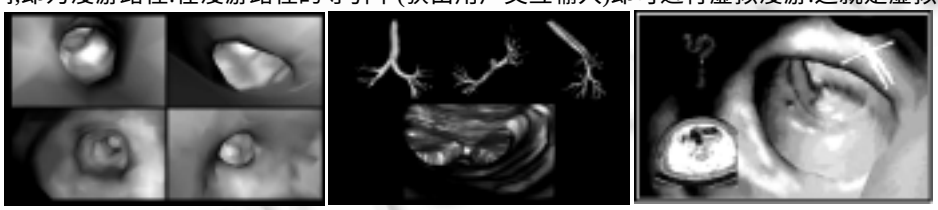
关 键 词: 医学图像;虚拟内窥镜;渐进网格;漫游
中图法分类号: TP391 文献标识码: A

1 绪 论

内窥镜技术在现代医学检查中有着重要的地位,但内窥镜会给病人带来许多痛苦.虚拟内窥镜技术的开展为克服这一问题带来了曙光.

1.1 虚拟内窥镜原理概述

虚拟内窥镜技术是随着计算机技术、计算机图形学、计算机图像处理尤其是虚拟现实等学科的发展而逐步形成的一种独特的技术[1~3].虚拟内窥镜包括如下主要内容:医疗切片图像数据获取、图像分割、面数据提取、虚拟漫游.技术路线可以描述如下:当从 CT 等医疗扫描设备获取数据后,对所得到的图像进行分割,把所要观察的器官(组织)与背景分离开来;在此基础之上进行分割,得到器官(组织)的空间数据.分析该空间数据,找出可以行进的空间,即为漫游路径.在漫游路径的导引下(获由用户交互输入)即可进行虚拟漫游.这就是虚拟内窥镜.



(a) Colon (b) Bronchus (c) Colon
(a) 结肠 (b) 支气管 (c) 结肠

Fig.1 Typical pictures of virtual endoscopy navigation
图 1 虚拟内窥镜进行过程中的典型画面

整个检查过程没有与病人身体有任何接触,避免了不必要的痛苦.另外,虚拟内窥镜可以对常规内窥镜无法到达的区域,如椎管、膝盖等进行检查,还可以越过肿瘤肿块的阻碍.

* 收稿日期: 2000-07-18; 修改日期: 2001-01-02
基金项目: 国家自然科学基金资助项目(60071002;60072007;69931010;60172057);国家 863 高科技发展计划资助项目(863-306-ZT04-06-4)

作者简介: 曹勇(1974－),男,甘肃兰州人,硕士,主要研究领域为医学图像处理,虚拟内窥镜;田捷(1960－),男,安徽芜湖人,研究员,博士生导师,主要研究领域为医学图像处理,多媒体技术;张晓鹏(1963－),男,陕西西安人,博士,副研究员,主要研究领域为医学图像处理,虚拟内窥镜;邱峰(1975－),男,重庆人,硕士,主要研究领域为医学图像处理,PACS 系统.

1.2 虚拟内窥镜中的主要技术分析

面绘制是虚拟漫游中通常采用的数据表示和显示方式.面绘制中使用的面片数据是在体素级上生成的.典型的算法一般采用 Marching Cube[4]及其各种改进版本.同一组面片对应于包围物体的一组等值面,含有物体的几何及拓扑信息.通常在一个体素中生成 1~4 个三角形,所以生成的三角面片的数量是非常庞大的.而 CT 或 MRI 等在作扫描时并不考虑器官的表面特性,即无论表面平坦与否都将均匀扫描.这样,在平坦的表面将产生大量冗余的三角形[5],增加了图形系统的负担.因此,面绘制的一个首要任务是如何精简面片,减少数据量.而在绘制过程中则可以采用 LoD(level of detail)技术.三角面片数量减少了,绘制速度自然也就提高了.

(1) 面片精简[6,7].面片精简有两个实施时机:在面片生成的同时以及在面片生成后或再作精简.

① 在生成面片的同时进行精简可以良好地保持对物体描述的拓扑一致性.这在医学图像领域是至关重要的.

② 在面片生成以后再作精简则纯粹是对面片作几何操作,物体的几何和拓扑信息会有所丢失.但它可同时为 LoD 准备数据.

(2) LoD[8~10].LoD 是为提高绘制速度而提出来的,基本思想是根据场景对画面的贡献大小以及人眼对画面的敏感程度决定绘制的精度.精度要求高的地方采用物体的精细描述进行绘制,而要求不高的地方则采用物体的粗略表示进行绘制,从而缩短绘制时间.采用面片精简技术,选取不同的精度阈值即可产生物体的不同细节层次表示.

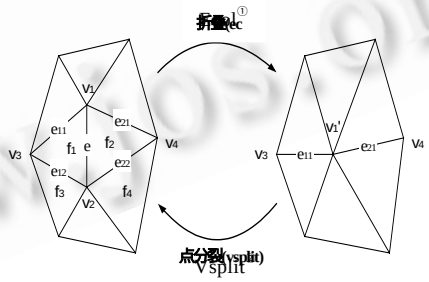
2 递进网格算法简要介绍

Hugues Hoppe 在 1993 年提出了以边折叠、点分裂为基础的三角面片简化的单位操作[11].后来,又在此基础上加以改进[12~14],并将之应用到地形的绘制上[15].这种以 LoD 为主要目的的算法在实现上比较清晰,便于进行动态的 LoD 数据简化和显示,非常适合虚拟内窥镜系统的需求.

我们选取此算法作为整个虚拟内窥镜系统的模型和基本算法,并在此基础上对该数据表示和算法核心进行一定的改进,以更适合系统的要求和数据的特点.

2.1 算法主体描述——边折叠和点分裂

边折叠和点分裂这两个互为逆操作的面片变换是实现递进网格的基础,在面片简化和压缩的算法中都采用了这一对操作作为基础.下面我们通过图 2 来说明这两个操作.



①边折叠,②点分裂.

Fig.2 Transformation of edge collapse and vertex split

图 2 边折叠和点分裂变换的说明

2.2 递进网格的三维面片数据表示和记录

我们可以通过一系列的边折叠(ecol)过程将一个初始的面片 $M^{\wedge}=M^n$ 简化成不能再简化的最粗糙的面片 M^0.如何判断一个面片不能再继续简化,将在后面的实现的过程中加以讨论.那么我们可以把这个简化的过程表述成下面的形式:

$$\left(M^{\wedge}=M^{n}\right)\xrightarrow{ecol_{n-1}}M^{n-1}\xrightarrow{ecol_{n-2}}\cdots\xrightarrow{ecol_{0}}M^{0}$$

构造序列($ecol_{n-1},ecol_{n-2},\ldots,ecol_0$)的过程也是选择性地进行边折叠的过程,选择性的原则和选择函数在这个过程中非常重要,决定了面片压缩的质量和运行速度.关于选择性的原则和选择函数,我们将在下面进行讨论.

我们认为,变换 $ecol_i$ 的逆过程是 $vsplit_i$,并且这两个过程是完全可逆的.那么又可以将上面的面片由 $M^{\wedge}=M^n$ 到 M^0 的简化过程写成下面的由 M^0 到 $M^{\wedge}=M^n$ 的恢复过程:

$$M^0 \xrightarrow{vsplit_0} \ldots \xrightarrow{vsplit_{n-2}} M^{n-1} \xrightarrow{vsplit_{n-1}} \left(M^{\wedge}=M^n\right)$$

由于这种恢复过程是完全无损的压缩恢复,那么我们可以将原来的 $M^{\wedge}$ 的传统表示改变成(M^0,{$vsplit_0,vsplit_1,\ldots,vsplit_{n-1}$})的形式,我们把这种形式称为面片的递进网格表示.我们在记录和存储的时候,首先记录 M^0 的传统表现形式,然后再记录 $vsplit$ 的序列,就可以完全地表示 $M^{\wedge}$ 了.

2.3 视点相关原则

在我们进行三维漫游时如何将面片以一种简化的形式表现在我们的面前?要解决这样的问题,关键在于将递进网格的简化、恢复与视点的相关性体现出来.这就是 3 个视点相关的原则,它们是:

① 视锥原则——在视锥之外的边一定要进行折叠,视锥内的边使用后两个原则.

② 面的方向性原则——Normal 方向和视线方向的夹角是否大于 90°,是则折叠.

③ 屏幕空间几何误差原则——三角形投影到像平面后小于分辨率的图元就简化掉.

3 虚拟内窥镜应用中的递进网格算法分析

递进网格算法提出时是没有针对特定的应用场合的,它对于面片的拓扑结构几乎没有特殊的要求,只是要求一点:面片中的一条边不能是 3 个或 3 个以上面的公共边.这样其边折叠和点分裂才能正确地进行.

但是对于一个算法和数据结构的通用性要求很强,由于不能很好地应用特殊面片的先验知识和特殊的应用要求,一般都有一定的代价.

由于是首次将递进网格算法在虚拟内窥镜中应用,我们应该注意以下几个问题:

① 我们处理的面片一般都是管状的,要求尽量保留管口面片的几何结构.

② 除了管口外,管体的内部是没有空洞的,尤其是没有三角型的空洞.

③ 大量面片在视锥之外或不可见.

④ 事先计算出漫游的路径,并且让视点沿着漫游的路径前进.

⑤ 视点的移动不能穿过漫游物体的管道壁.

4 改进的递进网格算法

尽管已经开始有人将 LoD 的想法加入到虚拟内窥镜应用中来进行漫游加速,但将递进网格的思想和算法融入其中还是没有先例的.我们首次应用递进网格到虚拟内窥镜中,希望按照应用的特殊性并保证一定的通用性,对递进网格的算法作一些改进,更适合于在这个应用场合中采用.

4.1 应用递进网格算法进行虚拟内窥镜漫游的总体过程

整个过程分成两个主要阶段:面片预处理简化过程和面片实时 LoD 变化漫游显示过程.

4.2 面片预处理时针对应用的改进

在虚拟内窥镜的应用中,处理的数据中一般是在 100 万个面左右,很小的管体也是有 10 万个面的.Hoppe 的基于能量的边折叠选择算法,虽然压缩简化的效果较好,但是时间太长了.对于有 100 万个面的物体,我们一般只需要 3~5 万个面就能很好地显示了.那么我们要把 100 万个面简化到 5 万个面的话,要做 47.5 万次边折叠(每次折叠减少两个面).Hoppe 在文章中说,其算法可以每秒简化 30 个面,就是说每秒做 15 次边折叠.这样要做 47.5 万次边折叠的话,要花费 8.8 个小时的时间.这对于医学诊断来说是不可忍受的.

在预处理时耗时的工作主要有两个:一个是求解很多次折叠时的折叠值,另一个是按照代价值进行排序.在

Hoppe 的处理中,为了得到最佳的边折叠代价,其代价函数的计算非常复杂.而且由于其每次折叠后都要重新计算所有图元到原始面片的代价,然后进行排序.而对于简化的开始阶段,大部分的图元的简化代价都是很相似的.比如将 100 万个面片进行简化,当简化到 20~30 万个面片之前时,其边折叠的代价差别都是非常小的.这些边折叠的次序对简化的效果几乎没有影响,但却消耗了大量的时间.

对于这种虚拟内窥镜的大数据(50~100 万个面),我们采用了下面的改进方式:

① 为了避免代价值的重复计算,我们在第 1 次计算完所有图元的代价值之后,只是比较本次折叠前后面片属性改变的代价.

② 我们把初步简化分成 3 个阶段:

第 1 个阶段,是将面片从 100 万简化到 20 万左右的过程.在这个过程中,由于边折叠的代价值很接近.我们就不用计算代价值,也不用进行排序.而是设定一个长度的阈值.

第 2 个阶段,实现面片简化到用于漫游的面片数目:代价函数为边的长度与边的两个端点的外法向量的差别的加权和.然后使用改进的堆排序的算法进行排序.这个阶段每秒能简化掉 200 个面左右.在第 2 个阶段结束后,就将所有简化掉的图元信息从机器中删除掉.认为这个时候的面片才是我们要进行虚拟漫游所要的信息.

第 3 个阶段,Vsplit 序列.采用复杂的代价函数——主要通过折叠后对新点坐标值的精确确定来保证每次的折叠尽可能少地改变面片的几何属性.在这个过程中,应用医学诊断的要求和医学物体的特点对边折叠进行优化,加速折叠过程.这个阶段的化简速度也可以达到每秒 50~100 个面.

通过这种改进,一般一个 100 万个面的面片简化只需要 50 分钟就可以了,在速度上的改进是很大的.已经可以满足医学诊断的要求了.

4.3 边折叠和点分裂时的改进

在选择边进行折叠的时候我们应用医学面片物体的特点,不处理管道入口处的图元,避免了对于洞孔的处理(这个问题我们将在下面一节中详细加以讨论).这样在边折叠时,我们可以保证该局部的图元都是存在的.而在 Hoppe 的文章中没有提到对这种情况的处理.比如在图 2 中,如果 v_2 是边界上的点,面 f_3 不存在的话,我们所要做的处理将会复杂很多.因为在删掉和改变关联关系的 4 个面 f_1,f_2,f_3,f_4 中任何一个都可能不存在,而是三角型的空洞,在我们要处理这种情况时,要做很多的判断.边折叠作为简化的单位操作,如果在处理上很复杂将会大大降低简化的效率.

我们作的另一个改进是为了保持边界上的点的位置不变.我们认为边界上的几何信息非常重要,所以要维护边界或管口上的点位置不能改变.如图 2 所示,我们在作边折叠时,折叠后新点的位置与 v_1 点和 v_2 的位置都是不一样的.然而,当 v_1 或 v_2 点是边界上的点时,我们就不能再这样处理了.如果 v_1 是边界点,则新点 v'_1 的位置就与 v_1 的位置相同.v_2 是边界点的处理也是相同的.在下面的说明中,我们知道,边选择时我们保证两个点不会同时是边界点,所以这样的处理可以保证边界点的位置,就能看到管口的位置不会改变.

4.4 边选择时策略的改进

我们选择一条边进行折叠,是因为它满足两个条件:一个是目前它的折叠代价最小,另一个是折叠该边具有合法性.对于第 1 个条件是处理代价函数和进行排序的问题.这里,我们提出的对选择策略的改进是针对第 2 个条件的.

① 该边本身就是处于边界上的边.

一个边处于边界,也就是说,该边所关联的两个面中有一个不存在.这种处于边界上的边对于虚拟内窥镜的应用就是管状物体的管口位置.为了保证不发生管口效应,我们要求管口的边不能进行折叠.

② 在折叠后会产生折叠效应的边.

这种情况是在作边折叠中常常遇到的情况,但是我们没有在 Hoppe 的文章中见到详细的说明.下面我们先对这种折叠效应进行分析.

如果我们现在正好选择了边 e 进行边折叠的话,我们看看这时会出现什么样的问题呢?显然,面 f_3 和 f_6 以

及 f_4 和 f_5 将发生重面的情况.这两组面的 3 个顶点都是相同的.而且在折叠后我们看到边 e_{31} 将重叠到 e_{61} 中.这时,我们发现面 f_3, f_6, f_7, f_9 这 4 个面都共一条边 e_{61} 了.我们称面对 f_3 和 f_6 发生了折叠效应,且面对 f_4 和 f_5 发生了折叠效应(如图 3 所示).前面提到过,我们对面片拓扑结构的要求有一条是:一条边最多只能和两个面相邻.所以这样进行边折叠是要出错的.

这种情况该如何处理呢?我们曾经想过将面 f_3, f_6, f_4, f_5 都删掉.因为面 f_3 和 f_6 重叠后形成的面是一个孤立面,这个面和整个面片只有边 e_{61} 是相连的.可是这样做的话,处理的过程就会非常复杂,我们必须知道有多少重折叠效应要发生,而且必须找到会发生折叠效应的所有的面对,都做相同的处理.折叠情况及折叠处理的复杂将使简化效率降低.

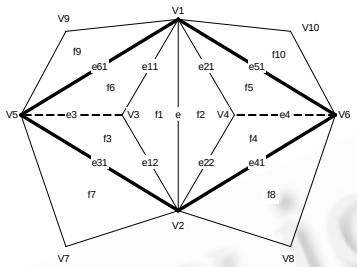


Fig.3 Edge collapse which may cause collapse irregularity

图 3 会发生折叠效应的边折叠的情况

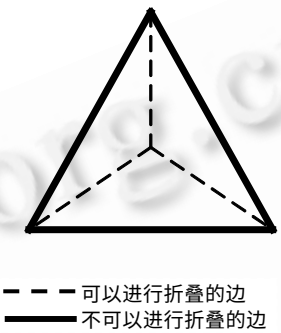


Fig.4 Dealing with collapse irregularity

图 4 折叠效应的处理说明

现在我们采用了一种更科学的方法处理这种折叠效应.那就是若将图 4 中虚线边先进行边折叠的话,就不会发生这种情况了.作为统一的考虑,对于所有的情况,当我们遇到上面这种面片局部情况时,如果折叠粗线的边就会发生折叠效应,这个时候我们只要先将虚线的边进行折叠就可以阻止这种情况的出现了.

如何判断这种情况的出现呢?让我们再看图 2.我们将点 v_1 的边的所有的顶点看成集合 Set_1,将点 v_2 的边的所有的顶点看成集合 Set_2,然后再看集合 $\mathrm{Set}_1 \cap \mathrm{Set}_2$ 的结果,如果结果集合中有 3 个或 3 个以上的元素,那么我们认为折叠边 e 将会发生折叠效应.这种做法很直观和简单,证明可以留给读者了.

我们在计算边的折叠代价值的时候保证它一定小于一个阈值 C,那么一旦发现该边的折叠会发生折叠效应,我们就将该边的代价值加上 C 重新加入到活动的边集合中进行排序.这样就可以保证绿色的边一定比红色的边先折叠(因为代价值小).

③ 邻接边界的边.

我们先定义如果一个顶点的一条边在边界上,那么我们说这个点是边界点.

那么有两种边属于邻接边界的边.一是该边的两个端点都是边界点.这样会产生很大的拓扑结构改变,所以这种情况要避免.

第 2 种情况比较复杂.我们来看图 2,我们定义 e 为邻接边界的边,当且仅当 v_1, v_3, v_4 同为边界点或 v_2, v_3, v_4 同为边界点时.

这种情况的发生是因为该边的折叠将一条原来只有一个边界顶点的边变成了两个顶点都是边界顶点的边.在有一种情况下,这种边也是可以折叠的,但是要判断出这种情况非常复杂,会极大地增加时间复杂度,而且这种情况不做折叠在视觉上可以接受,也很少发生.

4.5 基于视点相关的碰撞检测

由于碰撞检测的计算量巨大,所以我们必须在进行其他运算的时候得到碰撞的计算结果.我们现在来看在视点相关原则中的屏幕空间几何误差原则.在应用这个原则的时候不是把面片当前的图元都投影到屏幕上,再计算屏幕误差忍受范围,而是计算某类图元(如边图元)到视点的距离,再将该图元的大小进行考虑,如果这个图元的距离很远,而且图元的长度或面积很小,我们就可以将这种图元进行简化了.我们看到,在这个应用中,已经

计算了视点到图元的距离,那么如果我们应用这个计算结果就可以很好地进行碰撞检测了.如果我们在一个方向上距离一个图元太近了,就禁止视点向着这个方向移动.这样就可以很好地进行碰撞检测.

5 改进的递进网格算法的实现

由于整个虚拟内窥镜系统是为 Windows NT 系统环境设计的.我们的改进的基于递进网格算法的简化和漫游系统也是在 Windows NT 系统中实现的.我们使用 C++为程序开发语言,在 Visual C++和 C++ Builder 的集成环境中进行相应的开发.

5.1 递进网格格式文件处理与数据结构

递进网格文件存储的就是 M^0 的传统面片格式,再加上 Vsplit 序列.上面,我们讨论了 Vsplit 序列是存储在数据结构 CollapsedEdgeArray 中的.我们通过图 2 来看看到底要存储哪些信息才能够进行点的分裂?

首先记录对哪个点进行分裂,记录 v_1 值;要知道生成的两个新点 v_1 和 v_2 的坐标位置,记录 v_1 和 v_2 的坐标;记录 e_{11} 和 e_{21} 的边的索引值和面 f_3 的索引值.这样就够了.我们首先可以通过 e_{11} 和 e_{21} 找到 v_3 和 v_4.当我们知道 e_{11},e_{21} 及 f_3 以后,由 e_{11} 沿 f_3 出发,通过分裂前的 v_1 的边就可以找到和其相邻的 f_4,同时可以找到那些属于新点 v_2 的边了.那么分裂后的 v_1 的边我们也知道了.通过这些信息我们就可以进行点分裂了.

将这些信息按照点分裂的顺序(也就是边分裂的逆顺序)把 Vsplit 序列存储到文件里,然后再按照这个顺序读入.每读入一个 Vsplit 元素我们就进行一次点分裂,这样就可以完全恢复出原来的面片了.

5.2 虚拟漫游时的数据结构

在进行虚拟漫游的时候,我们主要进行控制的数据结构是与 Vsplit 序列相关的数据,也就是 CollapsedEdgeArray.还有就是所有活动的图元,特别是活动的边图元数据,也就是 ActiveEdgeArray.由于我们在漫游时使用的是上面说过的第 1 种处理方式,即每次对所有的能折叠的边进行计算,选出要进行折叠的边,从 CollapsedEdgeArray 中对所有的元素应用视点相关原则,从这些可以进行边折叠的信息中选择要进行的边折叠.在选中的边折叠中有些可能已经折叠过了,有些可能要进行折叠，还有些不要折叠的边也进行了折叠,因此,要对这些折叠应用点分裂恢复.这些操作会在下面详细地进行讲解.

5.3 漫游的路径控制

我们要求自动漫游和人为控制漫游能够随时交互进行.在漫游前先选定了一条自动漫游的路径,就是从一个管口进去,从另一个出来.在进行自动漫游时,我们可以随时终止自动漫游,转换成人为控制的模式,进行进一步的详细的观察,然后再回到自动漫游的路径上进行下面的自动漫游.这样可以让用户很快地找到想观察的病患处,然后由医生控制进行详细的观察.

5.4 视点相关原则和视点函数

我们使用的视点相关原则是视锥原则、面的方向性原则和屏幕空间几何误差原则.

对于屏幕空间几何误差原则,我们是看图元(主要是边图元)到视点的距离以及图元的大小,距离视点近且较大的图元要更精细一些,而远而较小的则可粗糙一些.所以我们将边折叠的视点代价函数定义如下:

首先根据物体面片大小的平均大小和显示区域的大小计算出屏幕误差忍受的阈值 C.由于我们可以根据图元的大小(S_{object})和图元到视平面的距离(D)计算出该图元投影到视平面上的大小(S_{project}):

$$S_{\text{project}}=aS_{\text{object}}+bD,$$

其中 a 和 b 是权值系数,所以我们对于 S_{project} 大于 C 的图元就不用选择出来了,否则就选择出来作为待简化的图元.在上面的图元函数中其实只有 D 是自变量,而图元本身的大小是不会因为视点的变化而改变的.所以我们把视点函数写成 $S(D)$的形式,表示视点函数只与图元和视点的距离有关.

在我们的系统中,选择的图元是边,而图元的大小就是边的长度.实际上,边在视平面上表示和边与视线的方向也是相关的,而且边的长度在边折叠时发生了改变,这里,我们为了加快显示速度进行了一定的简化处理.

6 有待进一步改进的几个方面

下面这些问题是几个有待解决和改进的地方.

■ 视点路径信息在促使面片简化时的应用

假定我们已经知道漫游路径,那么我们就可以在促使的简化中应用这些信息,效果就会好很多.但是这种应用面临的问题较多,我们在征求 Hoppe 本人的意见的情况下,都没有得到很好的答案.

这个问题以后是必须进行处理的,希望在代价函数的选择以及排序的问题上能有所突破,以提高简化速度,使得实时的初试简化成为可能.

■ 被遮挡的正向面的视点相关简化

对被遮挡的面的处理一直是计算机图形学中的一个难题.当我们用 OpenGL 方法处理的时候是用 Z-Buffer 处理的,而不是利用简化面片的方法进行处理.对这个问题的处理与硬件直接相关.希望能够有个很好的结果出来.

■ 复杂面片拓扑结构的处理

我们处理的面片要求是一条边最多有两个面与其相邻接.但实际上确有多个面与一条边相邻的情况.对于边界上的边的处理也存在很多的问题.这些问题大都与面片的几何拓扑结构相关.

7 算法结果分析

下面给出一些静态简化的结果:

■ 头骨骼的简化(预处理的静态简化)

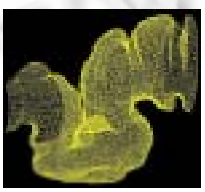
1 040K faces

200K faces

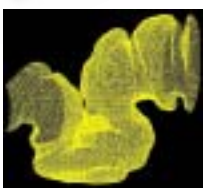
100K faces

面片数目变化	面片简化数	简化时间(s)	简化速度(Faces/s)
1 040 140→200 000	840 140	114.49	7 338.17
200 000→100 000	100 000	8.34	11 990.41
100 000→50 000	50 000	4.17	11 990.41

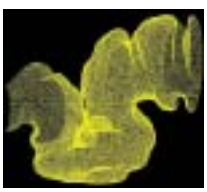
■ Avc_colon 肠子的简化



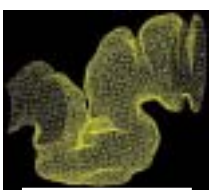
30 422 faces



15 422 faces



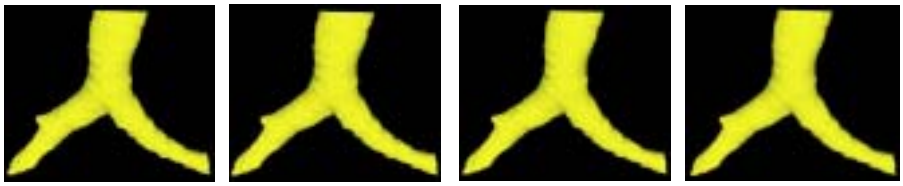
10 422 faces



5 422 faces

面片数目变化	面片简化数	简化时间(s)	简化速度(Faces/s)
30 422→15 422	15 000	103.77	144.55
15 422→10 422	5 000	16.98	294.46
10 422→5 422	5 000	8.90	561.80

■ Airways 气管的简化



26 979 faces 16 979 faces 11 979 faces 6 979 faces

面片数目变化	面片简化数	简化时间(s)	简化速度(Faces/s)
26 979→16 979	10 000	66.27	150.90
16 979→11 979	5 000	11.22	445.63
11 979→6 979	5 000	3.74	1 336.90

References:

[1] Pascal, H., Gérard, L.B., Coatrieux, J.L. 3D navigation in medicine. IEEE Engineering in Medicine and Biology, 1996,15(2):70~78.

[2] Hong, L., Kaufman, A., Wei, Y-C., *et al*. 3D virtual colonoscopy. In: Loew, M., Gershon, N., ed. IEEE Symposium on Frontiers in Biomedical Visualization. Los Alamitors, CA: IEEE Computer Society Press, 1995. 26~32.

[3] Yamashita, J., Yamauchi, Y., Mochimaru, M., *et al*. Real-Time 3-D model-based navigation system for endoscopic paranasal sinus surgery. IEEE Transactions on Biomedical Engineering, 1999,46(1):107~116.

[4] Lorensen, W.E., Cline, H.E. Marching cubes: a high resolution 3D surface construction algorithm. Computer Graphics (SIGGRAPH'87 Proceedings), 1987,21(4):163~169.

[5] Shekhar, R., Fayyad, E., Yagel, R., *et al*. Octree-Based decimation of marching cubes surfaces. In: Nielson, G., Silver, D., ed. Proceedings of the Visualization'96. Los Alamitors, CA: IEEE Computer Society Press, 1996. 335~342.

[6] Algorri, M-E., Schmitt, F. Mesh simplification. Computer Graphics Forum, 1996,15(3):77~86.

[7] Cohen, J., Varshney, A., Manocha, D., *et al*. Simplification envelopes. In: Rushmeier, H., ed. Proceedings of the SIGGRAPH'96 Conference (Annual Conference Series). ACM SIGGRAPH, Addison Wesley, 1996. 119~128.

[8] Astheimer, P., Poche, M. Level-of-Detail generation and its application in virtual reality. In: Singh, G., Feiner, S.K., Thalmann, D., eds. Virtual Reality Software and Technology: Proceedings of the VRST'94 Conference. World Scientific, 1994. 299~309.

[9] Certain, A., Popovic, J., DeRose, T., *et al*. Interactive multiresolution surface viewing. In: Rushmeier, H., ed. Proceedings of the SIGGRAPH'96 Conference (Annual Conference Series). ACM SIGGRAPH, Addison Wesley, 1996. 91~98.

[10] Gross, M.H., Gatti, R., Staadt, O. Fast multiresolution surface meshing. In: Nielson, G., Silver, D., eds. Proceedings of the Visualization'95. Los Alamitors, CA: IEEE Computer Society Press, 1995. 135~142.

[11] Hoppe, H., DeRose, T., Duchamp, T., *et al*. Mesh optimization. Computer Graphics (SIGGRAPH'93 Proceedings), 1993,27:19~26.

[12] Hoppe, H. Progressive mesh. In: Rushmeier, H., ed. Proceedings of the SIGGRAPH'96 Conference (Annual Conference Series). ACM SIGGRAPH, Addison Wesley, 1996. 97~108.

Application of a Progressive Meshes Algorithm to Virtual Endoscopy*

CAO Yong, TIAN Jie, ZHANG Xiao-peng, QIU Feng

(*Laboratory of Artificial Intelligence*, *Institute of Automation*, *The Chinese Academy of Sciences*, *Beijing* 100080, *China*)

E-mail: tian@dr.com; zxp@dr.com

http://www.3dmed.net

Abstract: Progressive meshes algorithm presented recently is used to solve key problems in Virtual Endoscopy in this paper. The advantages of the data structure, data simplification and navigation of virtual endoscopy are introduced. The speciality of the application of the progressive meshes algorithm to virtual endoscopy is analyzed, and some limits are pointed out also. An improved progressive meshes algorithm is presented and realized, and

some satisfactory results are obtained. This improved algorithm can solve the data simplification and LoD based navigation better than before.

Key words: medical imaging; virtual endoscopy; progressive mesh; navigation

* Received July 18, 2000; accepted January 2, 2001

Supported by the National Natural Science Foundation of China under Grant Nos.60071002, 60072007, 69931010, 60172057; the National High Technology Development 863 Program of China under Grant No.863-306-ZT04-06-4

未来软件技术国际学术会议(ISFST 2002)

征 文 通 知

在当今信息社会中计算机软件技术占有重要地位,是发展迅速的高科技产业.由日本软件协会、联合国大学澳门软件研究中心和国内外著名高等学校、研究机构联合组织的《未来软件技术国际学术会议》(International Symposium on Future Software Technology).已成功举办了 6 届.成为世界上有一定影响的软件技术国际学术会议.第 7 届未来软件技术国际学术会议将于 2002 年 10 月 23~25 日在武汉召开.会议将广泛征集论文、充分交流信息、促进国际合作与交流,将第 7 届未来软件技术国际学术会议办成一次高水平学术盛会.主办单位为华中科技大学(中国);日本软件技术协会(日本);联合国大学澳门软件研究中心(中国澳门).支持单位为中国国家自然科学基金委员会.承办单位为华中科技大学(中国).所有录用论文会前将由国外正式出版.

一、征文范围

软件开发方法/工程技术;
软件处理(建模/管理)/能力成熟度模型(CMM);
Web 相关技术;
数据库/数据管理;
软件体系结构与构件;
测试与验证;
系统演进和维护;
计算机应用中人的方面;
紧密集成软件;
软件质量/质量保证;
基于统计的质量和过程管理;
约束/关键链路理论;
信息安全;
电子商务、电子政务与电子教育;
实时、智能与移动计算;
软件集成环境;
数字企业与企业软件.

二、重要时间和联系方式

摘要接收期限: 2002 年 3 月 20 日 第 2 轮通知: 2002 年 5 月 10 日 全文截稿日期: 2002 年 7 月 10 日

论文摘要要求(中英文各一份): 论文题目;作者(包括姓名、工作单位、电话、传真、电子邮件、地址);摘要正文(500 字);关键词(不超过 5 个).

联系人:彭世卿,李蓉蓉 武汉华中科技大学科学技术协会(430074)

E-mail: hxwu@public.wh.hb.cn Fax: 027−87547971 Tel: 027−87543051

E-mail: glorias@263.net Tel: 027−87522513