

针对面片的 Reeb 图骨架抽取算法

黄坤武, 唐 杰, 武港山

(南京大学计算机软件新技术国家重点实验室, 南京 210093)



摘 要: 骨架是三维模型几何形状的表达方式之一。它保存了模型的拓扑特性, 有着广泛的应用前景。提出了一种基于面片采用 Reeb 图对多边形网格模型进行骨架抽取的算法。通过对模型进行一定的预处理保证面片的规则, 定义面片间距离计算方法, 创建模型的对偶图, 识别连通分量, 在连通分量上应用 Reeb 图的计算思想抽取原模型的骨架。试验表明, 该算法具有较好的计算效果和效率, 最终的骨架保存了模型的拓扑连通性以及姿态, 可用于基于内容的三维模型检索时的特征描述符。

关键词: 骨架; 三维模型; Reeb 图; 面片

中图分类号: TP391

文献标识码: A

文章编号: 1004-731X (2006) S1-0052-05

Skeleton Extraction Algorithm Using Reeb Graph Based on Facets

HUANG Kun-wu, TANG Jie, WU Gang-shan

(State Key Laboratory for Novel Software Technology, Computer Science & Technology, Nanjing University, Nanjing Jiangsu 210093, China)

Abstract: Skeleton is one of geometry shape presentations of 3D Models. It contains the topological feature of the model, so it's used widely. A practical skeleton extraction framework based on facets using Reeb graph was proposed. The model was preprocessed to assure the regulation of facets, create the dual graph of it, and then adapt Reeb Graph to extract the skeleton which contains enough information of the model, such as gesture and topological features, to be able to be used as descriptor of 3D model retrieval.

Key words: skeleton; 3d-model; reeb-graph; facet

引 言

随着计算机图形学的迅速发展, 对三维模型的研究日益深入。骨架作为形状表示的一种有效形式在三维模型的各个研究领域中广泛采用。Blum^[1] 1967年给出了骨架的最初定义: 骨架(中轴)是模型内部各个最大内切球中心的集合。它还有一个grassfire的模拟定义, 从模型表面开始点火, 各个方向上的火的相遇点所构成的集合。因为模型的骨架很好的保留了模型的拓扑连接性及其形态^[2], 所以经常被用于碰撞检测、三维动画、模型渲染^[3]、模型表面重建、模型检索^[6]等应用中, 也有研究人员采用骨架为模型的分解做矫正^[4]。不同的应用, 对于骨架应该保存的信息要求不完全相同, 故而抽取思路也不完全相同, 本文抽取的骨架主要用作三维检索的特征描述符。

对三维模型骨架的研究由来已久, 出现过很多方法, 有的是源于对二维图像的扩展, 有的是针对三维模型提出的, 大体上来说, 有如下几类: (1)基于拓扑细化技术。该类算法主要应用于采用体表示的模型上, 通过不断地进行不改变模型拓扑特性的体元素的削减来实现骨架抽取, 因为模型的体表示数据量巨大, 所以整个过程比较耗时。Gong^[5]等人提出

过一种并行细化算法, 通过先定义模型的简单顶点、删除谓词和两个细化元操作, 在抽取算法中不断迭代两个元操作进行模型细化; (2)基于距离矩阵。它一般的计算对象要求也必须是体表示的模型, 通过计算每个体元素的距离来求取模型的脊点。Sundar^[6]等人提出过基于该思想的一个算法思路, 通过计算每个体元素到边界的最小距离来得到骨架点, Sundar采用该算法创建了一个模型检索系统; (3)亦有学者将二维图像领域中的Voronoi图技术引入到三维骨架抽取中, Dey^[7]等人提出过一种利用Voronoi图直接近似中轴的算法。因为对三维模型而言只有部分Voronoi顶点能够汇聚成骨架, 所以他通过定义角度和比例这两个与大小、比重都无关的筛选标准来实现它的中轴近似算法, 并证明该方法能够保证收敛; (4)基于Reeb图思想。该类算法首先在模型上定义一个连续函数, 计算每个模型顶点的函数值, 将具有相同函数值的顶点聚合成一个顶点, 得到模型的骨架, 其中著名的有Hilaga^[8]等人提出的MRG, Hilaga定义的连续函数为顶点到整个模型表面所有顶点的最短距离与面积之积的和。(5)基于模型分解。Lien^[9]等人观察到对于保存模型连接性的分解过程就是一个骨架抽取的过程, 所以提出了基于模型近似凸面体分解的骨架抽取算法, 通过计算模型表面的桥识别模型表面的凹地, 计算每个顶点的凹陷性, 并对模型按照凹陷性进行分解, 不断迭代该过程创建骨架。(6)其它类。Ma^[10]等人提出过基于可见互斥力的骨架抽取算法, 它应用了物理学上的互斥力的概念来计算模型上的平衡点, 最终得到模型的骨架。

收稿日期: 2006-03-29

修回日期: 2006-05-25

基金项目: 国家自然科学基金重点项目(60533080); 国家自然科学基金重点项目(60503058)。

作者简介: 黄坤武(1980-), 男, 江西宜春人, 硕士生, 研究方向为三维模型检索; 唐杰(1971-), 男, 江苏南京人, 博士, 副教授, 研究方向为CAD/CAM、计算机辅助几何设计、计算机图形学; 武港山(1967-), 男, 博士, 博导, 研究方向为 Web 信息检索、多媒体信息处理。

一般采用Reeb图进行骨架抽取的思路,都将计算对象放在模型的顶点上,而后根据顶点的函数值计算其所在面片落入的各个函数值区间,进行骨架创建。本文作者观察到模型的面片和骨架之间存在着这种映射,并从寻找这种映射出发,提出了一种针对模型面片进行直接骨架抽取的算法框架。首先定义模型面片之间的距离计算方法,创建模型的对偶图,然后在该对偶图上应用Reeb图的计算思想,在对偶顶点上定义一个连续函数并计算每个顶点的函数值,最终根据计算得到的函数值以及顶点对应的面片之间的邻接关系创建模型的骨架。

1 模型预处理

本文算法建立在对模型面片的计算基础之上,所以面片的规则性非常重要。模型预处理的目标就是使得其面片都比较规则,即所有面片都近似于等边三角形。本文考虑过两种不同的预处理方法,第一种是^[8]中提到的边切分,将长度大于用户指定阈值的边进行不断切分;第二种是自适应的网格细分技术,将满足一定条件的面片进行细分,不满足的进行保留。

1.1 边切分

如图1所示,其处理过程就是不断地寻找大于阈值的边进行切分。注意,切分时新添加的边可能必须进行再次切分。

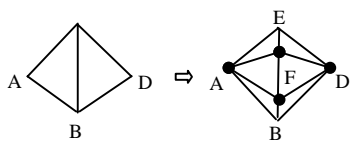


图1 边切分

1.2 面片细分

如图2(a)所示,若面片ABC包含两条边|AB|, |AC|均大于阈值(也可能只有一条边大于阈值), (b)图为细分后的结果。

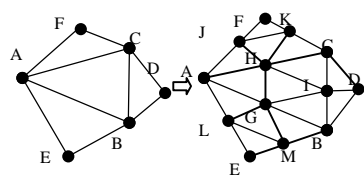


图2 面片细分

处理方法为:计算面片ABC上边顶点H, G, I的坐标,而后添加三个顶点,删除面片ABC,添加面片AGH, HIC, HGI, GBI, 递归的修改相应的相邻面片的信息。以H点为例,该边顶点的计算方法为^[11]:

$$V_H = \frac{3 \cdot \sqrt{V_A} + 3 \cdot \sqrt{V_C} + \sqrt{V_F} + \sqrt{V_B}}{8} \quad (1)$$

V_H 的坐标是通过与该边相邻的两个面片的四个顶点进行加权计算的,所以它将可能不落在边AC上,这样细分后的面片将更加的平滑。因为采用了自适应的策略,所以面片

BDC边长都小于一个阈值不需细分但因相邻面片被细分故被切分成了BDI, IDC。根据Loop细分的策略,它可以无限的循环细分下去,只是必须考虑到细分后的时间复杂度。对于不同的模型,因为创建的分辨率不一样,所以细分的次数是不一致的,必须在实际应用中加以控制。另外必须注意的是添加面片时顶点的顺序必须保持与原来面片的顶点顺序一致。另外,本文算法并没有如Loop细分中一样重新计算 V_A, V_B, V_C 的新坐标,是考虑到可能会剧烈修改原有模型的表面特性。

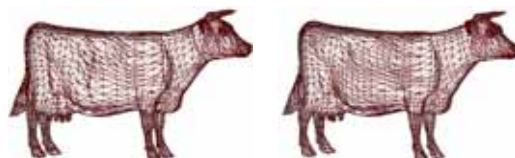


图3 预处理结果

图3(b)为采用面片一次细分计算得到的模型,而图3(a)则是采用边切分后的结果,边长阈值相同。从两个图中腹部的对比可以发现,采用细分技术处理的模型,面片比较均匀,虽然模型中可能还包含有大于阈值的边存在,但已经比较均匀。主要是因为:1.模型的特点,牛模型在的腹部网格不够均匀,比较多的面片不够规则;2.技术特点,(1)细分时并非简单的取现有平面上的顶点,而是进行了新顶点的拟合,将会改变原有面片的法矢;(2)细分引入的边都比较均匀,创建的面片也就均匀。所以当模型的某个区域包含较大数量不够均匀的面片时,采用曲面细分进行预处理的效果要略好于边的切分,但是如果模型本身已经比较规则,只有少量面片不规则时,那么采用边切分进行模型预处理将可能有着比面片细分更好的计算效率和效果。

2 创建模型对偶图

本文算法基于面片的聚合,而面片间除具有相邻关系外没有其它相关信息,为了后面计算的需要,必须定义面片之间的距离计算方法。同时为便于后续计算创建模型的对偶图。

本文定义面片间距离计算方法如下:若面片相邻,则定义它们之间的距离为面片之间重心的测地线距离与二面角距离的加权和;若面片不相邻,则定义它们的距离为面片通过相邻面片计算得到的最短路径。

本文创建面片的对偶图,为每个面片创建一个对偶顶点,取面片的重心。若两面片相邻,则在它们的对偶顶点间添加一条连接弧,计算两个面片的距离,也就是计算每个连接弧权重(弧长)。对于连接弧的权重,可以考虑采用不同的计算方式,本文采用对偶顶点之间的测地线距离与二面角距离的加权和。

如图4所示,面片ABC与面片BDC通过边BC相邻。E点和F点分别是它们的重心,G点是它们的测地线与BC边的交点。E点到F点的测地线距离就是:

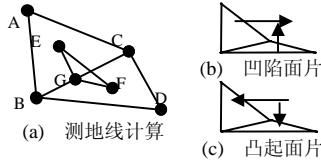


图 4 二面角距离计算

$$Geod(E,F)=|EG|+|GF| \quad (2)$$

G 点的计算方法如下:

$$\vec{V}_G = \vec{V}_B + t * (\vec{V}_C - \vec{V}_B) \quad (3)$$

t 为一个系数,若 G 点为 E, F 测地线与 BC 边交点,则 EG, FG 是 EF 在两个平面内的投影,三角形 EFG 的面积是最小的,即:

$$S = 0.5 * |\vec{EG} \times \vec{EF}| = 0.5 * |t \cdot \vec{BC} \times \vec{EF} + \vec{V}_B \times \vec{EG} + \vec{V}_E \times \vec{V}_F| \quad (4)$$

取最小值。最终可以计算到当 $t = \frac{|\vec{BF} \times \vec{FE}|}{|\vec{BC} \times \vec{FE}|}$, S 值最小。若 $0 < t < 1$ 表示, G 落在 BC 之间,如果 $t > 1$,那么表示 G 点落在了 C 点的外侧, G 点就应该取 C 点,如果 $t < 0$,则表示 G 点落在了 B 点的外侧, G 点应该取 B 点。

观察到面片之间的二面角,如果接近于 180° ,那么两个面片属于同一区间的概率比较大,反之,如果越大于 180° 或越小于 180° ,那么不属于的概率越大。这里定义面片之间的二面角距离计算方法^[12]:

$$Ang_Dist(E,F) = \xi (1 - \cos \beta) \quad (5)$$

其中 β 为二面角,比例系数 ξ 控制当二面角为如图 4(b)(c) 所示的两种不同二面角情况时采用不同的权值计算角度距离。判断方法可以通过面片的法矢与面片的角度间的关系进行确定。对于凸起面片 ξ 可以略小,而凹陷面片则略大,因为凸起面片聚合的概率较大,而凹陷面片则被分开概率较大,如凸起时 $\xi = 1$,而凹陷时 $\xi = 1.1$ 。

最终弧权重(弧长)计算方法为^[12]:

$$weight(E,F) = \delta * Geod(E,F) + (1 - \delta) * Ang_Dist(E,F) \quad (6)$$

δ 是比例系数,它控制测地线距离和角度距离的加权比例,是一个经验值,在试验中可以采用不同的值进行测试。

在创建模型的对偶图时,同时计算出了模型包含的所有连通分量。在后续的各种处理中本文算法都将不再以模型为单位,而是以对偶图中包含的连通分量列表中的连通分量为单位。这样对于包含多个连通分量的模型,将减小计算的时间复杂度,同时为其后续计算提供方便。一方面简化了问题空间,另一方面也为并行的处理提供基础。

3 计算每个顶点的函数值

创建了模型的对偶图后,就可以开始计算每个面片的函数值。因为每个面片采用它的对偶顶点表示,所以下文中提到的顶点如无说明都将指面片的对偶顶点而非模型原有顶点。

考虑到计算的效果和条件限制,这里采用^[8]中提到的顶点函数定义方式,定义顶点连续函数为:

$$\mu(v) = \sum_i^n g(v, b_i) \cdot area(b_i) \quad (7)$$

其中顶点 v 是指面片的对偶顶点, b_i 是指从某个连通分量上抽取出来的基本顶点, n 是该连通分量上抽取得到的基本顶点个数, $g(v, b_i)$ 是指顶点 v 到 b_i 的最短距离。

所以整个函数值的计算过程分为两步: 1. 以连通分量为单位抽取基本点集; 2. 根据基本点集计算每个连通分量上的每个顶点到其上所有基本点的最短距离,并累加求和。为了便于后面的骨架创建,在得到每个顶点的函数值之后,对其值进行了归一化,将所有的函数值放到 [0,1] 中。

3.1 抽取基本点集

抽取基本点集的方法采用^[8]中提到的基本点选取方法,同时考虑到计算对象不再相同,所以,做了一点修改。首先是基本点的选取步长阈值,采用了某个连通分量中的弧长的平均值的倍数,同时考虑到连通分量中包含的顶点个数的参差不齐,做了一个细致的划分。下面是本文倍数选取策略。

当连通分量中,顶点数大于 2000,倍数=16;顶点数在 [1000, 2000],倍数=8;顶点数在 [1000, 500] 时,倍数=5,在 [500, 100] 时,倍数=4,在 [100, 20] 的时,倍数=2,其它的倍数=1。之所以这样做是为了保证每个连通分量上的基本点集包含的顶点数在一定范围之内,既不会出现孤立点,也不会出现太多的情况。从目前作者试验过的模型来看这个区分效果还可以,只是在遇到大的连通分量的时候,得到的基本顶点数还是会很多,造成计算的时间复杂度提高。可以考虑在顶点数更多的时候将倍数调大,减少最终得到的基本顶点数。

确定了基本顶点的步长阈值之后,就可以采用 Dijkstra 算法求取连通分量上所有的基本点列表。本文作者通过一个最小二叉树模板类 binTree,用来实现 Dijkstra 算法,定义了每个顶点所属的基本点列表。

输入: 连通分量,步长阈值 ξ
输出: 基本点列表

1. 将所有顶点的最短路径长度置为无穷大,标记为未处理;
2. 取一未处理顶点作为基本点 v_b , 将其最短路径长度置为 0, 并标记该顶点为已处理, 插入 binTree, 同时插入基本点列表;
3. 从 binTree 中取最小顶点 v, 计算其相邻顶点 v_i 到它的距离 $length(v, v_i)$, 若 $g(v) + length(v, v_i) < \xi$, 且 $g(v) + length(v, v_i) < g(v_i)$, 则将 v_i 插入 binTree, 修改 $g(v_i) = g(v) + length(v, v_i)$, 并置 v_i 为已处理;
4. 若 binTree 不空, 回到 3; 若为空, 则查看是否有未处理的顶点, 若有则跳至 2; 若没有, 则退出。

图 5 抽取基本点

下面是本文作者计算得到的模型的基本点集试验结果,这两个模型都只包含一个连通分量。

如图 6 所示,每个模型表面都由不同颜色的区域组成,每个区域就是一个基本点所占的面片集合,区域中间的点为基本点。上述两个模型的基本点长度倍数分别采用了 16 和 8。从图上可以看出: 1. 基本点的步长很明显的会影响基本点选取的数量; 2. 最终选取的基本点很均匀的分布在模型的表面上,虽然每个基本点所包含的面片数量可能略微有点差异。

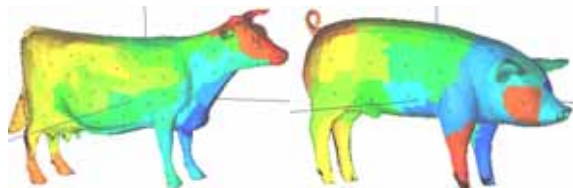


图6 基本点分布

3.2 计算每个顶点的函数值

得到了连通分量上的基本点集之后,就可以开始计算连通分量上每个顶点的函数值了。此时方法依然采用 Dijkstra 算法,计算每个顶点到所有基本点的最短路径,而后进行累加求和。整个计算过程包括如下三个步骤:

1. 对基本点集进行循环,采用与抽取基本点集类似的 Dijkstra 算法计算以它为源点到连通分量上各个顶点的最短路径;

2. 累加每个顶点到各个基本点的最短路径与基本点面积的积,并记录本连通分量中最大和最小的函数值;

3. 采用公式(9)将连通分量中顶点的函数值进行归一化。
$$\mu(v)' = (\mu(v) - \min(\mu(v))) / (\max(\mu(v)) - \min(\mu(v))) \quad (8)$$

说明: 1. 计算每个顶点的函数值采用的 Dijkstra 算法与在抽取基本点时稍微有点差别,即插入 binTree 的条件变成 $g(v) + length(v, v_i) < g(v_i)$; 2. 公式(9)中的 $\min(\mu(v))$ 和 $\max(\mu(v))$ 各个连通分量是独立的; 3. 本文作者考虑过计算顶点函数值时是否要采用基本点的面积进行加权的问题,并且针对两种计算结果作过试验,下面是计算结果对比。

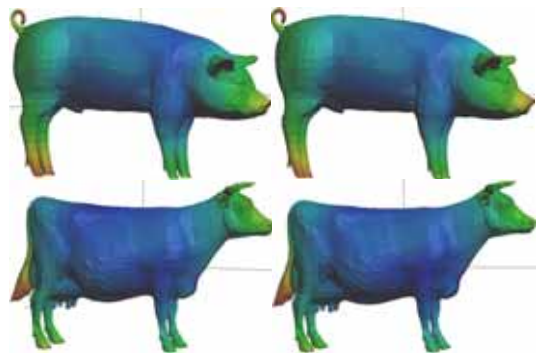


图7 面片函数值分布

如图7所示,模型表面由包含各种颜色的面片组成,面片的颜色跟据其函数值进行自动求取。其中右侧的模型函数值是累加到基本点的最短路径及其所占的面积,而左侧的则是简单的累加到基本点的最短路径的结果图。从图上可以看出: 1.模型的特征通过函数值都被很好的区分出来,头,脚,腿,尾巴都采用了不同的颜色进行标示; 2.函数值的计算过程中,面积的加权作用不是非常明显; 3.包含相同拓扑结构的模型计算出来的函数值的分布可能非常相近。

4 创建模型的骨架

得到了每个面片的函数值后就可以创建模型的骨架了。

首先将[0,1]区间平均分成若干子区间,然后根据面片的函数值将面片划分到相应的子区间中。同时根据面片之间的连接关系,将各子区间中的面片组合成相应的子连通分量,并同时记录各子连通分量之间的邻接关系。整个添加面片的过程是按照面片的邻接关系进行不断递归计算的过程。

最后为每个子连通分量采用公式(10)计算一个代表顶点,其中 \bar{v}_i 是每个面片的对偶顶点, \bar{v}_p 是计算得到的代表顶点, n 是子连通分量中包含的顶点数, $area(f_i)$ 表示面片的面积。根据子连通分量间的邻接关系添加代表顶点之间的连线。最终得到一个只包含顶点和边的图,这个图就是模型的骨架。

$$\bar{v}_p = \frac{\sum_{i=0}^n \bar{v}_i * area(f_i)}{\sum_{i=0}^n area(f_i)} \quad (9)$$

5 试验结果及参数

本文作者采用 VisualC++6.0 对该算法进行了实现,并在一些有代表性的模型上进行了测试。图8、图9是本算法的试验结果与 MRG 算法^[8]得到的结果的对比如。左侧为本算法结果,右侧为 MRG 的结果。下面是相应的实验参数。



图8 牛骨架对比

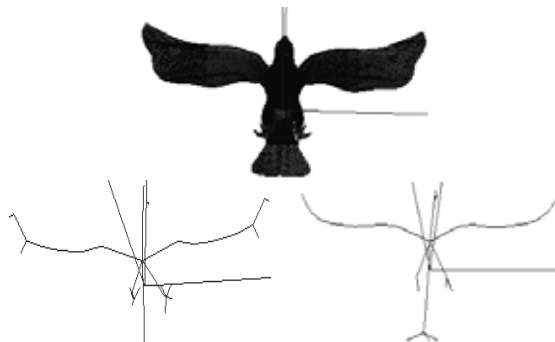


图9 鹰骨架对比

1. 进行预处理的边长域值,即大于该阈值的边将被切分或者细分,图中试验结果均采用了细分面片进行预处理,阈值为 0.05;

2. 弧长计算中,凸面角与凹面角的权重 ξ 、测地线距离和角度距离的权值 δ ,本文作者试验过多个参数,结果显示凸起时 $\xi=0.9$,凹陷时 $\xi=1$, $\delta=0.9$ 效果比较好;

3. 切分的份数,这里采用过 4, 8, 16, 值得指出的是,不同的模型不同切分份数结果可能相差很大。

表1为试验参数及结果记录,并与本文实现的 MRG 进行抽取的算法进行了效率上的对比。所有试验均在: P4 2.8G, 504M RAM, Windows xp Professional 环境下进行。

表 1 试验数据对比

| 模型 | 顶点数 | 面片数 | 边数 | 连通分量数 | 最大连通分量面片数 | 切分次数 | 计算时间 | MRG 处理时间 |
|----|-------|-------|--------|-------|-----------|------|---------|----------|
| 猪 | 3522 | 7040 | 10560 | 1 | 7040 | 8 | 0.953s | 12.062s |
| 牛 | 2904 | 5804 | 8706 | 1 | 5804 | 16 | 0.625s | 9.781s |
| 马 | 48485 | 96966 | 145449 | 1 | 96966 | 16 | 92.282s | 144.422s |
| 兔子 | 35947 | 69451 | 104176 | 1 | 69451 | 16 | 49.344s | 81.312s |
| 鹰 | 16542 | 33072 | 49608 | 1 | 33072 | 16 | 13.890s | 44.828s |

6 时间复杂度分析

预处理阶段, 采用边切分的方式, 时间复杂度为 $O(E)$, 而采用细分技术的话, 决定于细分的次数如果只进行一次细分, 那么它的时间复杂度是 $O(F)$; 创建模型的对偶图, 因为采用了递归添加的方式, 所以其时间复杂度为 $O(F')$; 抽取基本点集, 这个时候计算的单位为连通分量, 所以它的时间复杂度决定于顶点数最大的连通分量, 它的时间复杂度为 $O(\text{MAX}(V) \cdot \log(\text{MAX}(V)))$; 对于函数值的计算, 时间复杂度为 $O(C \cdot \text{MAX}(V) \cdot \log(\text{MAX}(V)))$; 骨架创建的时间复杂度为 $O(C \cdot \text{MAX}(V))$ 。

其中, E 为边数, F 为面片数, F' 是经过预处理后的面片数, C 为模型包含的连通分量数, V 为该连通分量包含的顶点数, 即经过预处理后的面片数。最终时间复杂度决定于顶点数最大的连通分量, 时间复杂度为 $O(C \cdot \text{MAX}(V) \cdot \log(\text{MAX}(V)))$ 。

7 结论

本文介绍了一种针对面片采用 Reeb 图进行模型骨架抽取的算法框架, 将 Reeb 图的应用推广到了模型的面片上。为了试验保证算法的准确, 对模型进行一定的预处理, 尽量使得模型的面片规则化。本文作者试验了采用自适应的 Loop 细分技术和对长度大于阈值的边切割技术进行预处理。定义了模型上任意面片之间的距离计算方法, 创建模型的对偶图同时计算相邻面片间的距离。在模型对偶图上应用 MRG 中提出的函数定义, 并对其方法进行了修改, 采用了一种自适应的基本点选取策略保证每个连通分量的基本点个数, 测试了函数值计算过程中基本点的面积加权影响, 对于函数值的归一化按照连通分量为单位进行。最终依赖计算得到的面片函数值以及面片之间的相邻关系将模型的面片划分到相应的区间中, 创建模型的一个合适的骨架。

整个算法能够处理包含多个连通分量的模型, 并且当模型中的连通分量比较均匀的时候有着比 MRG 等算法更佳的计算效率。抽取的骨架保留了原有模型的姿态, 连通性, 可应用于三维模型的检索。

参考文献:

- [1] P A Maragos, R W Schafer. Morphological Skeleton Representation and Coding of Binary Images [J]. IEEE Transactions on Acoustics, Speech, and Signal Processing, (S0096-3518), 1986, 34(5): 1228-1244.
- [2] 车武军, 杨勋年, 汪国昭. 动态骨架算法[J]. 软件学报, (S1000-9825), 2003, 14(4).
- [3] Nikhil Gagvani, Parameter-Controlled Skeletonization-A Framework for Volume Graphics. A thesis submitted to the Graduate School-New Brunswick Rutgers [D]. The State University of New Jersey in partial fulfillment of the requirements for the degree of Doctor of Philosophy Graduate Program in Electrical and Computer Engineering. 2001.
- [4] Li Xuetao. Decomposing Polygon Meshes for Interactive Applications [C]// Symposium on Interactive 3D Graphics Proceedings of the 2001 symposium on Interactive 3D graphics, 2001, 35-42.
- [5] WeiXin Gong, Gilles Bertrand. A Simple parallel 3D Thinning Algorithm [M]. In ICPR, ICS Press, ed. 1990, 188-190.
- [6] H Sundar, D Silver, N Gagvani, S Dickinson. Skeleton Based Shape Matching and Retrieval [C]// International Conference on Shape Modeling and Applications 2003, May 12-15, 2003, Seoul, Korea.
- [7] Tamal K Dey, etc. Approximating the Medial Axis from the Voronoi Diagram with a Convergence Guarantee [J]. Algorithmica. (S0178-4617), 2001, 38(1): 179-200.
- [8] Masaki Hilaga, Yoshihisa Shinagawa. Topology Matching for Fully Automatic Similarity Estimation of 3D Shapes. International Conference on Computer Graphics and Interactive Techniques [C]// Proceedings of the 28th annual conference on Computer graphics and interactive techniques table of contents, 203-212, 2001.
- [9] Jyh-Ming Lien, Nancy M Amato. Simultaneous Shape Decomposition and Skeletonization Using Approximate Convex Decomposition [R]. Technical Report, TR05-015, Parasol Laboratory, Department of Computer Science, Texas A&M University, Dec 2005.
- [10] FuChe Wu, WanChun Ma, etc. Skeleton Extraction of 3D Objects with Visible Repulsive Force [C]// Eurographics Symp. On Geometry Processing, 2003.
- [11] 王国瑾, 汪国昭, 郑建民. 计算机辅助几何设计[M]. 高等教育出版社&施普林格出版社, 2001.
- [12] S Katz, A Tal. Hierarchical Mesh Decomposition Using Fuzzy Clustering and Cuts [C]// ACM Transactions on Graphics, 2003, 22(3): 954-961.

广告索引

| | |
|---|----|
| 世界上完整的 3D、虚拟现实和可视化解决方案[美国科视数字系统公司]..... | 封二 |
| 液压六自由度运动平台[亿美博科技]..... | 封三 |
| 山东科技大学信息科学与工程学院简介..... | 封四 |