

基于细分曲面的多分辨率空间变形*

邵瑾, 冯结青⁺, 金小刚, 彭群生

(浙江大学 CAD&CG 国家重点实验室, 浙江 杭州 310027)

Multiresolution Space Deformation Controlled by Subdivision Surface

SHAO Jin, FENG Jie-Qing⁺, JIN Xiao-Gang, PENG Qun-Sheng

(State Key Laboratory of CAD&CG, Zhejiang University, Hangzhou 310027, China)

+ Corresponding author: Phn: +86-571-87951045, E-mail: jqfeng@cad.zju.edu.cn, <http://www.cad.zju.edu.cn>

Received 2004-04-05; Accepted 2004-07-05

Shao J, Feng JQ, Jin XG, Peng QS. Multiresolution space deformation controlled by subdivision surface. *Journal of Software*, 2004,15(Suppl.):197~205.

Abstract: Space deformation is an important technique for shape modification and flexible animation generation in geometric modeling and computer animation. A novel multiresolution space deformation method is proposed in the paper which is controlled by subdivision surface of arbitrary topology. It supplies a solution for the complex shape edit while method is fast enough to achieve realtime interaction. Several subdivision schemes are adopted and compared in the method. Experimental criteria are given on how to determine appropriate subdivision scheme and subdivision depth. To achieve realtime interaction, parameterization is accelerated through space-subdivision and deformation is evaluated locally. All of above algorithms are integrated into a space deformation prototype system. The system has the functions of multiresolution space deformation and abundant control mesh generation methods.

Key words: free-form deformation; realtime interaction; arbitrary topology; control mesh; subdivision surface; multiresolution

摘要: 在几何造型和计算机动画中,空间变形是一种重要的几何外形编辑和柔性物体动画生成技术.提出一种基于细分曲面控制的多分辨率空间变形方法,解决了变形控制手段的复杂性和变形效率之间的矛盾.对采用多种细分规则的变形结果进行了比较和分析,得出了选择适当的细分规则和细分深度的经验性原则.在算法效率方面,通过采用空间剖分技术加速了参数化过程,并且利用细分的局部性缩短了变形的响应时间,从而达到了实时交互的目的.最后,在上述算法的基础上实现了一个完整的空间变形系统,提供了多种交互手段用以生成控制网格,并实现了具有多分辨率性质的变形控制.

关键词: 自由变形;实时交互;任意拓扑;控制网格 S;细分曲面;多分辨率

* Supported by the National Natural Science Foundation of China under Grant Nos.60373036, 60333010 (国家自然科学基金); the National Grand Fundamental Research 973 Program of China under Grant No.2002CB312101 (国家重点基础研究发展规划(973))

作者简介: 邵瑾(1980-),女,浙江兰溪人,硕士,主要研究领域为几何造型,计算机图形学;冯结青(1970-),男,博士,研究员,主要研究领域为几何造型,计算机动画,科学数据可视化;金小刚(1969-),男,博士,研究员,博士生导师,主要研究领域为计算机动画,计算机图形学,虚拟现实;彭群生(1947-),男,博士,教授,博士生导师,主要研究领域为真实感图形,虚拟现实,红外成像仿真,基于点的造型与绘制,科学计算可视化.

随着三维获取技术的发展,由大规模的多边形元素组成的几何物体已经非常普遍.如何对这些离散的网格模型进行外形编辑,几何分析以及进行动画生成等是当前迫切需要解决的问题.作为一种重要的几何外形编辑技术,空间变形技术通常是独立于物体的表示的,它可以用于网格模型的编辑.空间变形是以用户交互操作为目的的,因而变形的速度和控制手段的灵活性是该技术的两个重要指标.

Barr 提出的整体和局部变形方法实现高效^[1],但是缺乏直观的交互手段.Sederberg 和 Parry 提出了自由变形方法(free-form deformation,简称 FFD)^[2],其他人在此基础做了许多改进.但是其基本思想是通过 Bézier 或 B-样条体来变形物体,其控制手段是规则的平行六面体网格(lattice),难以近似任意拓扑的物体形状.

扩展 FFD 方法(extended FFD)的控制手段是组合多个简单的棱柱形控制网格(prismatic lattice)构成的复合网格^[3],这种复合控制网格可以较为容易地近似任意拓扑的物体形状;然而,为了维持不同简单网格之间的光滑过渡,变形的灵活性受到了很大限制.

1996 年,MacCracken 等人提出了控制网格为任意拓扑结构的 FFD 方法^[4],他们利用体细分技术在任意拓扑网格(lattice)上定义了变形空间.其控制手段是任意拓扑的网格,可以很好地近似任意拓扑的物体外形;但是在该方法中,生成具有任意拓扑的,贴近物体外形的空间网格本身就是件困难的事情;它的时间和空间开销很大,难以实时交互;并且其变形空间的连续性没有保证.

Bechmann 等人提出了另外一种可以具有任意拓扑形状连续自由变形技术(continuous FFD)^[5].他们使用复合的 Bézier 四面体作为控制网格.由于复合的 Bézier 单形可以具有任意复杂的外形,所以该方法在变形控制方面具有优势.然而,为了使物体变形以后保持 G^1 连续,需要在变形过程中对网格施加连续性约束.

冯结青等人提出的基于曲面控制的变形技术^[6],其控制手段一般是初始化在 Oxy 平面上的规则矩形网格(mesh),在该矩形网格定义的 B-样条张量积曲面上定义了变形空间.该方法变形速度快,空间开销小,然而这样的控制手段仍然不够灵活,因为定义在矩形拓扑上的张量积曲面难以逼近物体的外形.

Singh 提出了 Wires 方法^[7],该方法的控制手段是空间曲线和隐式函数,具有相当的灵活性,适合运用于动画设计领域.

在本文中,我们提出了一种新的交互式变形方法.该方法利用细分技术将基于曲面控制的变形算法推广到任意拓扑的控制网格的情况,通过任意拓扑的二维流形网格来控制物体的变形.因为该方法以具有任意拓扑的网格作为物体变形的控制手段,因而它属于使用二维控制工具的空间变形模型.这样,它既具有控制网格为任意拓扑结构的 FFD 方法^[4]的灵活交互手段,又保持了基于曲面控制的变形技术^[6]的低计算开销.同时,该方法可以简单地实现如图 1 所示的多分辨率变形控制效果.

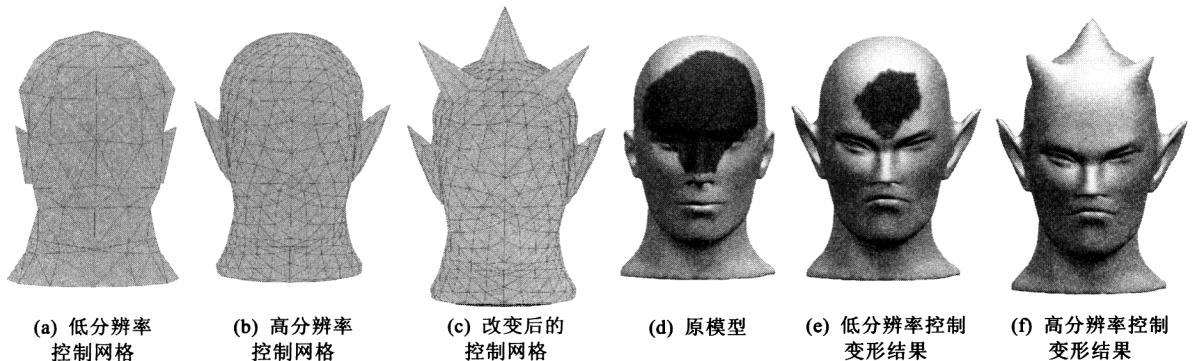


图 1 多分辨率控制变形结果及影响区域;阴影部分是模型上受影响的顶点区域

本文开始部分是引言和相关工作的介绍.第 1 节详细阐述基于细分曲面的空间变形方法.第 2 节介绍多分辨率变形控制的实现方法以及生成控制网格的交互手段.第 3 节给出一些变形结果的实例和算法的分析.第 4 节是结论与展望.

1 基于细分曲面的空间变形方法

1.1 变形过程

如图2所示,基于细分曲面的空间变形过程可以描述如下:

首先,有一个待变形的物体模型 M 和一个近似物体外形的二维流形控制网格 B ;通过细分控制网格 B ,得到 B 的细分曲面 S ,细分曲面 S 及其法向张成一个光滑的变形空间;然后,通过参数化过程,将物体模型 M 嵌入变形空间,求得物体模型 M 上每一个顶点的参数.至此,变形的准备阶段完成.

然后,用户可以交互编辑控制网格 B ,得到变形后的控制网格 B' ;通过细分变形后的控制网格 B' ,得到了变形后的细分曲面 S' ;利用在变形的准备阶段得到的物体模型 M 上每一个顶点的参数和变形后的细分曲面 S' ,我们可以计算得到物体模型 M 上每一个顶点的新位置,而在变形过程中,物体在变形空间中的参数保持不变.这样,控制网格 B 的变形就传递给了物体模型 M ,一次变形完成.

一旦变形的准备阶段完成,其后的几个步骤(图2下半部分的所有步骤)可以反复进行,直到获得用户满意的变形结果.

1.2 控制网格的细分

为了在任意拓扑的控制网格之上定义变形空间,最直接的想法是将模型的每一个顶点直接参数化到控制网格 B 上距离最近的点所在的三角片的局部坐标系上.然而,因为控制网格 B 由许多离散的面片组成,所以基于控制网格 B 的映射区域也是由离散的面片组成,显然它是不光滑的.作为结果,变形的结果也是不光滑的,并有明显的分段现象.图3就给出了这样的例子.



图3 直接参数化的变形结果

为了改进变形的结果,我们采用控制网格的细分曲面作为控制网格和物体模型的中间过渡,用控制网格的细分曲面和其法向张成的空间作为变形空间.细分曲面是根据某一个特定的细分规则,递归地细分原始网格的极限曲面.细分曲面通常至少是 G^1 连续的.这表明,我们所得到的变形空间是光滑的.多细分一次,其细分网格就会更接近于细分曲面.也就是说,一个 n 次细分的网格比一个 $n-1$ 次细分的网格要“光滑”.因此,我们可以采用不同细分深度的控制网格的细分,用以得到不同分辨率的细分网格来近似细分曲面.

另一方面,随着细分深度的增加,存储和计算开销也相应增长.因此,我们必须在光滑的结果和存储,时间开销之间找到一个折衷.实现结果显示,2~4次细分可以满足大多数情况的要求.

细分可以应用于具有任意拓扑的二维流形网格上,所以细分曲面具有很强的形状特征表示能力,它的形状是近似于初始网格的.因此,我们可以利用任意拓扑的控制网格近似物体形状,从而得到形状非常贴合物体外形的变形空间.这样,变形方法就可以为用户提供非常直观的操作方式.

同时,细分曲面具有局部性质.如果移动初始网格上的一个顶点,在最终的细分曲面或细分网格上,只有在邻近该顶点的有限区域上会发生改变.另外,细分还天然的具有 LOD 的性质.随着细分次数的增加,细分网格的形状总是近似于控制网格,但是网格却逐渐加密.这就为我们进行多分辨率的变形控制提供了便利.

为了细分的需要,控制网格必须是封闭的或有边界的二维流形网格.也就是说,对于控制网格的每一条边,有且只有两个面片与它相邻(对于边界上的边,可以只有一个面片与它相邻).

不同的细分规则会得到不同的细分网格,相应的变形结果也会有所不同.在实验中,我们选择实现了有代表

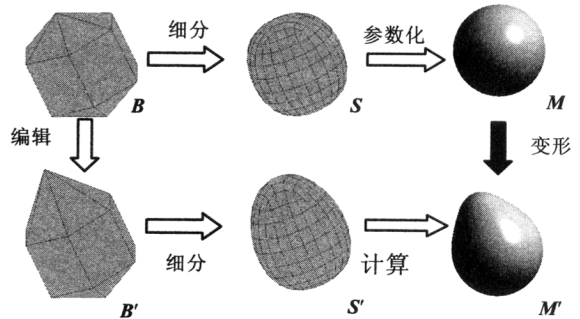
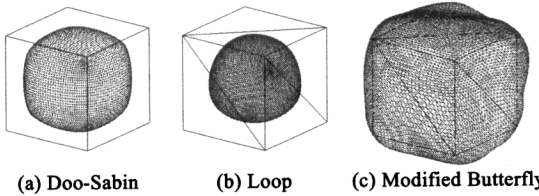


图2 基于细分曲面的空间变形过程

性的 3 种细分规则:Doo-Sabin,Loop 和 Modified Butterfly.事实上,基于细分曲面的空间变形方法是独立于细分规则的,也可以采用其他细分规则进行控制网格的细分.为了便于进行之后的参数化步骤,还需要三角化细分网格 S .

Doo-Sabin 细分^[8]是四边形网格的细分,它是分裂顶点的,近似的细分,并且在规则网格处是 C^1 连续的.Loop 细分^[9]是三角形网格的细分,它是分裂面的,近似的细分,并且在规则网格处是 C^2 连续的.Modified Butterfly 细分^[10]是三角形网格的细分,它是分裂面的,插值的细分,并且在规则网格处是 C^1 连续的.图 3 给出了分别采用这 3 种不同细分规则对 cube 网格进行 5 次细分的结果比较.

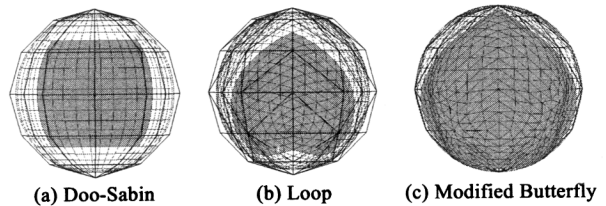


(a) Doo-Sabin (b) Loop (c) Modified Butterfly
图 4 采用不同细分规则对 cube 网格进行细分的结果

图 4(a)~图 4(c)的控制网格均为一个简单的立方体,当我们分别采用 Doo-Sabin,Loop, Modified Butterfly 细分规则对它进行了 5 次细分之后,就得到了不同的细分结果曲面(做 Loop 和 Modified Butterfly 细分之前,我们对 cube 网格作了三角化).从图 5 所作的对比中,我们可以得出以下几点结论:

- (1) Loop 细分方法得到的细分网格相对于控制网格收缩得最严重;
- (2) Modified Butterfly 细分方法得到的细分网格相对于控制网格没有收缩;然而,它得到的细分结果也最不光顺.
- (3) Doo-Sabin 细分方法得到的细分网格相对于控制网格的收缩程度介于前两种细分方法之间.

同样,当我们采用不同的细分规则时,控制网格上的一个顶点的影响区域是不同的.这是不同细分曲面的局部性的体现.图 5 显示了分别采用 Doo-Sabin,Loop,Modified Butterfly 细分规则对 sphere 网格进行了 2 次细分之后,得到的细分网格和其中一个控制顶点在细分网格上的影响区域.从图 6 中我们可以看出:Modified Butterfly 细分的局部性弱于其他两种细分规则的局部性.



(a) Doo-Sabin (b) Loop (c) Modified Butterfly
图 5 采用不同细分规则对 sphere 网格进行 2 次细分后,其上一个控制顶点的影响区域(灰色部分)

1.3 参数化

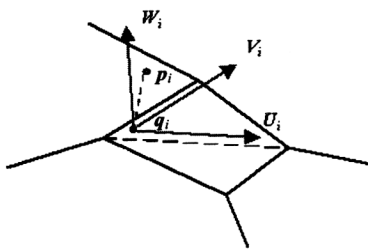


图 6 根据最近点原则参数化

我们根据最近点原则进行参数化.如图 6 所示,将物体模型上的每一点 p_i 参数化到细分的控制网格 S 上距离它最近的一点 q_i 的局部仿射坐标系内,并记录参数 $(j,k,(u_q,v_q),(u,v,w))$.其参数化的具体过程如下:

首先,找到在细分后的控制网格 S 上距离 p_i 最近的一点 q_i ,并记录下 q_i 所在的多边形面片的序号 j ,以及 q_i 所在的三角片在第 j 个多边形面片内的序号 k .假定 v_1,v_2,v_3 是 q_i 所在三角片的 3 个顶点,而 n_1,n_2 和 n_3 是这 3 个顶点相应的平均法向.注意:如果采用了 Loop 或 Modified Butterfly 细分规则,因为得到的细分网格本来就是三角形网格,无须做

三角化.因而,我们记录下的 j 是 q_i 所在的三角面片的序号,而没有必要额外记录 k .

然后,计算出 q_i 在 $\Delta v_1 v_2 v_3$ 内的重心坐标 $(u_q,v_q,1-u_q-v_q)$:

$$q_i = u_q v_1 + v_q v_2 + (1 - u_q - v_q) v_3 \tag{1}$$

在点 q_i 处的近似法向 n_q 可以通过对 n_1,n_2 和 n_3 的线形插值得到:

$$n_q = u_q n_1 + v_q n_2 + (1 - u_q - v_q) n_3 \tag{2}$$

这样,在最近点 q_i 处就建立起了一个仿射坐标系 (q_i,U_i,V_i,W_i) .如图 7 所示,仿射坐标系的 3 个坐标轴的方向向量

通过下面的公式给出:

$$\begin{aligned} U_i &= v_2 - v_1, \\ V_i &= v_3 - v_1, \\ W_i &= n_q \end{aligned} \quad (3)$$

最后,原始物体网格 M 上的一个顶点 p_i 被参数化到了局部坐标系 (q_i, U_i, V_i, W_i) 中,得到参数 (u, v, w) . 局部坐标 (u, v, w) 通过解下面的线性方程计算而得.

$$p_i = q_i + uU_i + vV_i + wW_i \quad (4)$$



(a) W_i 是三角形的法向,在边的两侧之间是不连续的 (b) W_i 是三角形顶点法向的插值,在边的两侧是连续的

图 7 在三角形边上的 W_i

通过上面的过程,我们就建立起了 p_i 和 q_i 之间的联系,就像是把 p_i 粘到了 q_i 上面一样,我们把这样的过程形象地称之为附着(attach).

在建立局部的仿射坐标系时,如果直接采用三角形 T_k 的法向量作为 W_i ,那么,所建立的变形空间在相邻的三角形之间将不是连续的,如图 7(a)所示.而我们在式(3)中的选择则可以保证所建立的变形空间是连续的,如图 7(b)所示.

在参数化的过程中,在 S 上找到距离 p_i 最近的一点 q_i ,是一个耗时的过程. q_i 可能是细分网格 S 上的顶点,也可能位于 S 的某一个三角片上.逐个比较 p_i 到 S 上的每一个三角片的距离,并不是一个有效率的作法.

我们采用空间均匀剖分方法加速求最近点的过程:

首先,把控制网格的包围盒作为空间剖分的原始空间,根据控制网格的平均边长来自适应地决定空间均匀剖分的单元格大小.

然后,把 S 上的每一个顶点都分配到空间均匀剖分的各个单元格中.这个过程只需要做一次.

最后,对于物体网格上的每一个顶点 p_i ,根据该顶点的位置,判断它落到了哪一个单元格中.利用空间剖分,可以快速地排除那些在空间分布上距离顶点 p_i 较远的三角片,它们所处的单元格和 p_i 所处的单元格既不相同,也不相邻.

1.4 修改控制网格和变形

控制网格 B 的修改可以通过使用传统的编辑工具移动顶点来实现,即允许用户通过移动顶点的位置来修改控制网格,而控制网格的拓扑是保持不变的.然后,控制网格的修改被传递给物体模型网格,这就是变形.

变形的过程主要包括如下两个步骤:

1. 修改后的控制网格 B' 被细分,其细分深度与 B 在第 1.2 节中所述的细分深度保持一致.然后, B' 的细分网格被三角化,得到 S' .

2. 根据局部坐标 $(j, k, (u_q, v_q), (u, v, w))$ 计算变形后的物体网格的顶点的新位置.

通过将原始物体模型 M 的每一个顶点 p_i 更新为 p'_i ,就可以生成变形后的物体网格 M' ,它和 M 具有同样的拓扑.参数 $(j, k, (u_q, v_q), (u, v, w))$ 对于给定的顶点 p_i 和它变形后的顶点 p'_i 是不变的.他们的功能与 FFD^[2] 中的局部坐标 (u, v, w) 的功能是类似的.

利用细分曲面的局部性,我们可以通过局部计算物体网格上顶点的新位置,加速变形后物体顶点的计算.当改变初始控制网格上少量点的位置时,细分的控制网格只会发生局部改变;同样的,在其上张成的变形空间也只会发生局部的改变.因而,物体网格上的顶点只有一部分会发生位移,我们只需重新计算这一部分受影响的顶点的新位置就足够了.

2 基于细分曲面的多分辨率空间变形

2.1 多分辨率控制变形

在我们提出的变形方法中,对于特定的控制网格,无论细分多少次,其上的每一个控制顶点在细分网格上的影响区域大小是基本固定的.而当我们加密控制网格后,以加密的网格作为新的控制网格进行细分,此时加密控制网格上的每一个控制顶点的影响区域必然会变小.这就为我们进行多分辨率控制的变形,雕塑物体局部特征提供了便利.同时,细分天然地具有 LOD(level of detail)的性质.随着细分次数的增加,细分网格的形状总是近似于控制网格,但是网格却逐渐加密.

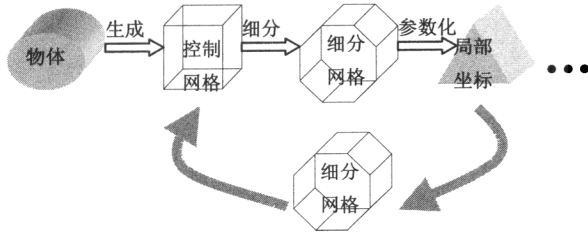


图 8 多分辨率变形控制的流程

因此,我们的系统采用细分的控制网格来作为加密的控制网格,进行多分辨率的变形控制.当用户先用粗控制网格对物体进行整体变形控制后,控制网格的细分网格形状也同时发生了改变,近似于变形后的物体外形;然后,用户可以选择多分辨率控制的变形,用修改后的控制网格的细分网格作为高分辨率的控制网格,细分新的高分辨率控制网格,重新参数化物体上的每一个顶点,通过交互编辑高分辨率的控制网格,对物体进行局部变形控制.多分辨率变形控制的流程如图 8 所示.

图 1 显示了多分辨率变形控制的控制网格,影响区域和变形结果.从图中可以看到,高分辨率控制网格上一个顶点的影响区域小于低分辨率控制网格上一个顶点的影响区域,因此,可以在高分辨率控制网格上进行细节编辑,例如让头上长角,而不会影响到模型的其他部分.

2.2 生成控制网格的交互手段

我们的变形方法支持整体变形,用户可以用一个很粗的简单网格,例如长方体网格控制变形,得到光滑的变形结果;但是,如果能够采用形状近似于物体模型形状的控制网格,就可以得到更加直观的变形结果,这种控制网格尤其适合用来控制对局部细节特征的塑造.而且可以使用任意拓扑,任意形状的控制网格,也正是我们方法的优越性所在.此外,我们的变形方法可以通过选择不同分辨率的控制网格,实现局部变形和整体变形.

我们在上述算法的基础上,实现了一个完整的空间变形系统.系统提供了大量的交互手段用以生成变形控制网格,如基本形体曲面,旋转面,swEEPing 曲面,曲面布尔运算以及一系列网格编辑操作等;此外,系统还可以基于 Reeb Graph 技术自动生成与物体外形一致控制网格.由于篇幅所限,我们将另文讨论上述网格生成方法.系统可以分别采用 Doo-Sabin 细分,Loop 细分和 Modified Butterfly 细分以及不同的细分深度,执行基于细分曲面的空间变形;系统还实现了具有多分辨率特点的物体变形控制,便于用户更好地雕塑物体细节特征.

3 实验结果与分析

我们提出的变形是光滑的.变形空间由控制网格的细分曲面和其法向张成.虽然,我们在实现过程中只采用了细分曲面的近似结果(即细分网格)来得到变形空间,但是变形空间仍然是光滑的.这是因为,我们采用了插值的法向,而不是三角形的真实法向.这个思想类似于 Phong Shading 采用插值的法向产生光滑的着色效果.

因为每一个顶点 p_i 和 S 上距离它最近的三角形 T_k 之间存在着简单的对应关系,撤销/重复(undo/redo)操作可以通过重新载入前一步或后一步的控制网格 B 来实现,或者通过在交互操作时,由用户将控制顶点直接拖回原来的位置.而在每一次变形步中存储控制网格将会造成额外的存储开销.

本文提出的变形算法在平移和旋转变换下具有不变性,这是因为细分曲面具有平移和旋转不变性.然而,在错切变换(shearing transformation)下,它不具有不变性.与 FFD 相似,这种变形的自由度是 3,因而它比基于参数曲面的变形具有更高的自由度和更好的控制力.

通过选择不同分辨率的控制网格,本文提出的变形算法可以将整体的变形和局部的变形集成到一起.如果

控制网格的分辨率比较高,并且只有一个或者少数几个顶点被移动,那么就可以实现局部的变形.我们的变形算法的局部性质为用户提供了有力的细节特征雕塑手段.此外,这个局部性质还使我们可以通过局部的变形计算来缩减交互时间.

图9给出了分别采用 Doo-Sabin 细分, Loop 细分和 Modified Butterfly 细分对 Dolphin 模型进行整体和局部变形的结果.我们用一个很粗的长方体面网格作为控制网格,进行整体变形控制.采用这3种细分规则的整体变形结果从整体上来看是近似的.然而,采用 Modified Butterfly 细分的变形结果有局部不光滑的情况,这是因为 Modified Butterfly 细分曲面的连续性低于其他两种细分曲面.我们用一个较为精细,更贴合模型外形的控制网格,进行局部变形控制,拉长 Dolphin 模型的嘴和拉高背鳍.采用这3种细分规则的局部变形结果从整体上来看也是近似的.然而,仔细观察采用 Loop 细分的变形结果的嘴的部分可以看出,其嘴部拉长的程度比其他两种细分规则的变形结果要小.这是因为,Loop 细分曲面的收缩情况比其他两种细分曲面要小,因此其得到的参数化结果和变形结果也就有了细微的不同.

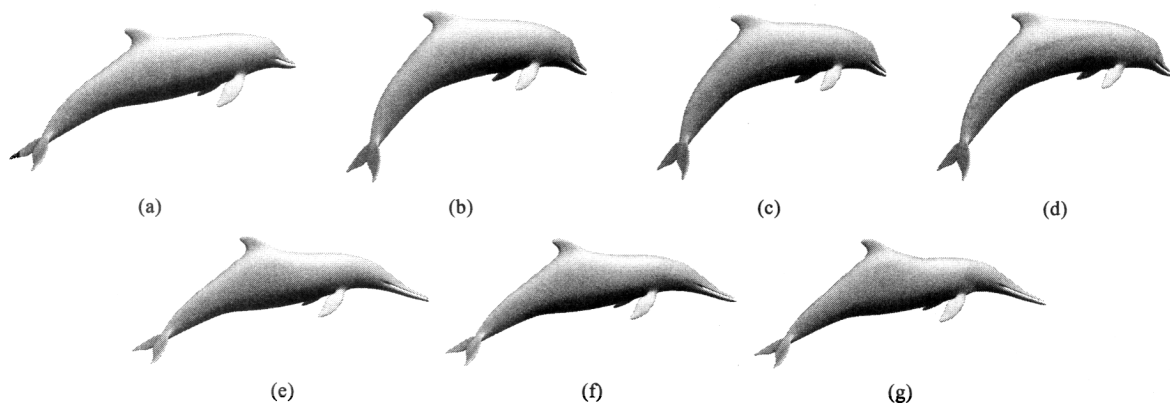


图9 采用不同细分规则的 Dolphin 模型变形结果.图9(a)为原模型;图9(b)~图9(d)分别为采用 Doo-Sabin, Loop, Modified Butterfly 细分的整体变形结果;图9(e)~图9(g)分别为采用 Doo-Sabin, Loop, Modified Butterfly 细分的局部变形结果

表1列出了分别采用 Doo-Sabin, Loop, Modified Butterfly 细分规则,对 Dolphin 模型进行变形的时间统计数据.其中控制网格 B 有44个顶点,48个多边形面片(当采用 Loop, Modified Butterfly 细分规则时,控制网格预先被三角化,面片数为84),物体模型 M (Dolphin) 有15 447个顶点.表格中的测试数据是在一台 Pentium-IV 1.7 GHz CPU, 256 MB 内存, Windows 2000 操作系统的 PC 机上获得的.

表1 运行时间统计. SD: 细分深度; PT: 参数化时间; DT: 变形时间

| 细分规则 | SD | Depth=1 | Depth=2 | Depth=3 | Depth=4 |
|--------------------|--------|---------|---------|---------|---------|
| Doo-Sabin | PT (s) | 0.275 | 0.308 | 0.430 | 0.950 |
| | DT (s) | 0.015 3 | 0.017 3 | 0.026 4 | 0.058 6 |
| Loop | PT (s) | 0.219 | 0.255 | 0.402 | 1.023 |
| | DT (s) | 0.015 9 | 0.019 8 | 0.034 2 | 0.097 3 |
| Modified Butterfly | PT (s) | 0.219 | 0.256 | 0.397 | 1.012 |
| | DT (s) | 0.015 9 | 0.020 2 | 0.038 3 | 0.111 4 |

从表1中,我们可以得出结论,采用这3种不同细分规则的运行时间是在一个数量级上的;运行时间会随着控制网格细分深度的增加而迅速增长;对于 Dolphin 模型这样数量级的模型,采用4以下(含4)的细分深度,响应时间都满足实时交互的需要.这是因为对于一个给定的物体和它的控制网格,在整个变形过程中参数化过程只需要执行一次,因此变形控制的响应时间只和变形时间相关.相比而言, Doo-Sabin 的运行时间略少于其他两种细分规则.

图10显示了用稀疏的球面网格作为控制网格,把一个球变形为一个碗的例子,和用长方体面网格作为控制网格,将一块平板变形为一块瓦片的例子.

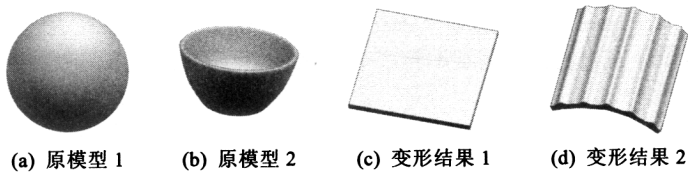


图 10 球面网格和长方体网格控制的变形

图 11 中使用的控制网格是我们通过包括三维网格布尔求并在内的各种交互手段生成的.我们首先生成一个旋转面网格包住壶身,然后用 Extrusion 操作拉伸出包住壶嘴的网格;同时用 sweeping 曲面手段生成包住壶把手的网格;最后,用三维网格布尔求并操作将这两个部分并成一个封闭的二维流形网格,经过三角化操作后,就得到了图中的控制网格.从图中我们可以看到,对壶把手的局部变形控制效果是很直观的,并且不影响壶的其他部分;而进一步对壶身和壶嘴的变形,也丝毫不影响壶把手的形状.

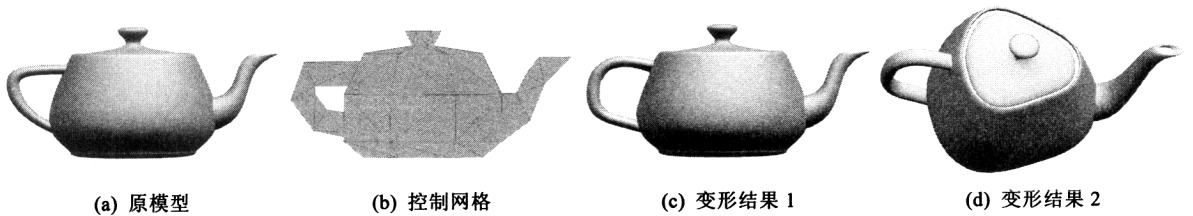


图 11 teapot 模型的变形

图 12 中控制 horse 模型变形的控制网格是这样生成的,先利用 sweeping 曲面网格生成从马头到马脖子,再到马身子的管状网格,并利用局部缩放功能,使控制网格的局部尺寸和模型相适宜;然后,利用多次 Extrusion 操作,在管状网格的合适部位伸出控制 4 条马腿的控制网格.

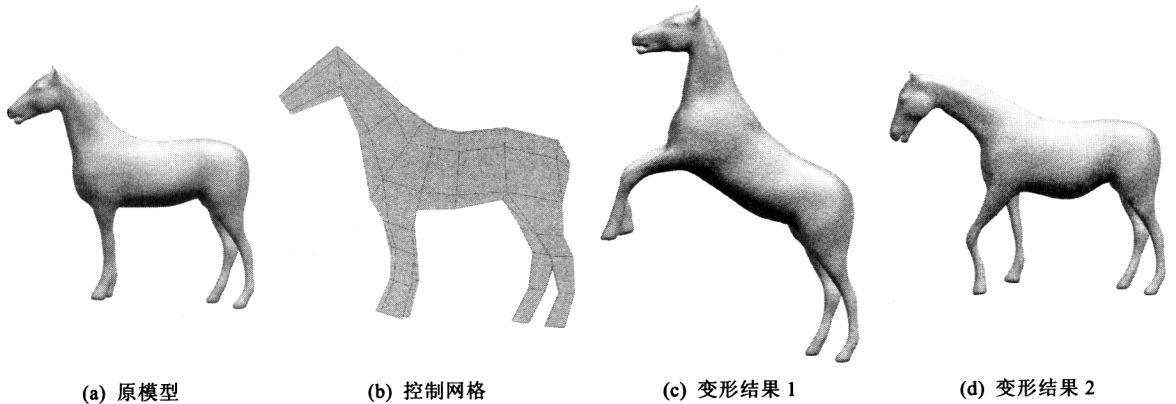


图 12 horse 模型的变形

4 结 论

空间变形是几何造型和计算机动画中一种重要的几何外形编辑和柔性物体动画生成技术.自 20 世纪 80 年代以来,此领域已经取得了大量的科研成果,许多算法已经作为几何造型和计算机动画系统的一部分,为用户提供了便利,显著的提高了工作效率.

本文对现有成果做了较为细致的总结,比较分析了现有的变形方法的控制手段和变形速度,提出了基于细分曲面的空间变形方法;与已有的自由变形算法相比,本文提出的算法控制手段灵活,直观,可以实时交互控制,并且可以对局部细节进行良好的编辑.在以上算法的基础上,实现了一个完整的空间变形系统.系统提供了大量的交互手段,如基本形体曲面,旋转面,sweeping 曲面,曲面布尔运算以及一系列网格编辑操作等,用以生成用户所需要的二维流形控制网格,实现变形操作;并结合细分曲面的 LOD 性质,系统还实现了具有多分辨率特点的

物体变形控制,便于用户更好地雕塑物体细节特征。

我们希望今后在以下方向进行深入研究:(1) 根据控制网络的属性,自适应地选择细分规则和细分深度;(2) 支持多分辨率变形,即支持物体模型的 LOD 表示;(3) 自动生成多边形物体的二维流形控制网格;(4) 在已有算法的基础上,研究直接操纵(direct manipulation)^[11]的基于细分曲面的空间变形方法。

References:

- [1] Barr AH. Global and local deformation of solid primitives. *Computer Graphics*, 1984,18(3):21~30.
- [2] Sederberg TW, Parry SR. Free-Form deformation of solid geometric models. In: *Proc. of SIGGRAPH 1986*. 21~30.
- [3] Coquillart S. Extended free-form deformation: A sculpturing tool for 3D geometric modeling. *Computer Graphics*, 1990,24(4): 187~193.
- [4] MacCracken R, Joy KI. Free-Form deformation with lattices of arbitrary topology. In: *Proc. of the SIGGRAPH'96, Computer Graphics Proc., Annual Conf. Series. ACM Press/ACM SIGGRAPH, 1996*. 181~188.
- [5] Bechmann D, Bertrand Y, Thery S. Continuous free form deformation. *Computer Networks and ISDN System*, 1997,29(14): 1715~1725.
- [6] Feng JQ, Ma LZ, Peng QS. A new free-form deformation through the control of parametric surfaces. *Computer & Graphics*, 1996, 20(4):531~539.
- [7] Singh K, Fiume E. Wires: a geometric deformation technique. In: *Proc. of the ACM SIGGRAPH 1998*. 1998. 405~414.
- [8] Doo D, Sabin M. Behavior of recursive division surface near extraordinary points. *Computer Aided Design*, 1978,10(6):356~360.
- [9] Loop CT. Smooth subdivision surfaces based on triangles [MS. Thesis]. Utah: University of Utah, 1987.
- [10] Zorin D, Schroder P, DeRose T, Stam J, Warren J, Weimer H. Subdivision for modeling and animation. In: *SIGGRAPH'99 Course Notes, Course 37. ACM SIGGRAPH, 1999*.
- [11] Hu SM, Zhang H, Tai CL, Sun JG. Direct manipulation of FFD: Efficient explicit solutions and decomposable multiple point constraints. *The Visual Computers*, 2001,17(6):370~379.