基于切平面中值坐标的 Laplacian 网格变形算法

赵 健[1] 胡剑光[2] 吴玲达[1]
(国防科技大学信息系统与管理学院 长沙 410073)[1] (海军兵种指挥学院 广州 510431)[2]

摘 要 网格模型变形往往需要保持局部几何细节,Laplacian 网格变形算法能够较好地保持局部几何细节特征,但细节特征描述子-Laplacian 坐标的计算欠缺精确性。从平面多边形中值坐标的角度出发,对 Laplacian 坐标进行重新定义,将顶点的一阶邻域投影到顶点处切平面上,根据顶点相对投影点的中值坐标构建的 Laplacian 坐标能够精确地描述模型的局部几何细节特征,实验验证能够获得较好的编辑效果。

关键词 网格变形,Laplacian 网格编辑,中值坐标,几何细节保持
中图法分类号 TP391 **文献标识码** A

Mean Value Coordinates on Tangent Plane for Laplacian Mesh Deformation

ZHAO Jian[1] HU Jian-guang[2] WU Ling-da[1]
(College of Information Systems and Management, National University of Defense Technology, Changsha 410073, China)[1]
(Naval Arms Command Academe, Guangzhou 510431, China)[2]

Abstract A requirement for Mesh deformation is to preserve local geometric detail. Laplacian mesh editing can meet this requirement, but Laplacian coordinates is not a accurate form to describe the local detail. This paper defined the Laplacian coordinates through mean value coordinates. First projects 1-ring of a vertex onto tangent plane, then constructs Laplacian coordinates use the mean value coordinates of the vertex. The result shows that this defination is accurate to describe local detail.

Keywords Mesh deformation, Laplacian mesh editing, Mean value coordinates, Geometric detail preserving

1 引言

网格变形技术作为一种重要的造型手段,已经成为计算机图形学领域一个十分活跃的研究热点,在影视动画、造型设计等领域有着相当广泛的应用。随着三维扫描设备获取网格模型的精细程度越来越高,在模型变形编辑的过程中能否保持模型的几何细节特征,成为衡量变形算法好坏的一个重要标准。

Lplacian 网格编辑是近年来图形学领域一个新兴的并得到迅速发展的理论方法,其理论基础建立在网格的微分属性上,因此能够很好地保持网格的局部细节特征。算法处理过程是将非线性问题转化为线性问题,因此执行过程高效、鲁棒。本文对 Lplacian 网格变形算法的本质进行深入的剖析,并从平面多边形中值坐标角度出发,对网格的微分属性进行精确的描述,使得网格模型的几何细节特征能够更好地保持。

2 相关工作

自 Sederbeg 等开创性地提出 FFD[1] (Free Form Doeformation,自由变形)以来,网格变形领域取得了丰富的研究成果。根据理论基础可将其划分为3个阶段:自由变形、多分辨率网格编辑和微分域网格编辑。其中自由变形从某种程度上讲仍然是建立在传统曲线和曲面造型理论基础上的。该方法一方面交互过程复杂不直观,另一方面不能够保持模型的局部几何特征。多分辨率网格编辑主要利用网格细分、滤波等技术将网格模型进行层次化处理,得到一系列不同精细程度的网格模型。模型的几何细节特征被显式地表示为不同层次之间的差异,主要的表示方法有位移向量表示法[2]、法向量位移表示法[3]和位移体积表示法[4]。多分辨率网格编辑的不足主要在于需要将原始模型进行层次化处理,并且仅适用于整体效果的编辑,对于精细编辑仍需回到高精细层次模型上。

微分域网格编辑算法与前两种方法的不同之处在于,微分域网格编辑针对的是网格模型的微分属性,如曲率、梯度等,采用的是一种隐式的细节特征表示方法。从细节特征的表示方式可以分为 Laplacian 网格编辑和基于梯度场操纵的网格编辑[5]。前者将几何细节隐式地表示为 Laplacian 坐标;后者则是将细节特征蕴于网格的梯度场中,然后利用泊松方程进行网格重建。两者的研究基础都是建立在网格模型的微分属性之上的,因此都属于微分域网格编辑的范畴。

Lipman 等[6]改进了 Laplacian 网格编辑中的旋转不变性问题,通过初始解显式地求得一系列变换矩阵,用来修正原始 Laplacian 坐标序列,但算法计算复杂。Sorkine 等[7]提出了变换矩阵的一种隐式求解方法,适用于小角度的旋转,但对于

到稿日期:2008-09-25 返修日期:2009-01-07 本文受 863 高技术计划项目(2006AA01Z319)资助。
赵 健(1981—),男,博士研究生,主要研究方向为虚拟现实;**胡剑光**(1978—),男,讲师,主要研究方向为指挥自动化;**吴玲达**(1962—),女,教授,博士生导师,主要研究方向为多媒体信息系统、虚拟现实。

大角度旋转变形修正能力有限。周昆等[8]提出利用体图法(Volumetric Graph)构建模型内部和外部体网格，外部网格用于避免变形过程中的自交叉，内部网格用于避免网格模型的体积发生显著变化。该方法提高了编辑效果，但算法相对复杂。许栋[9]对基于梯度场操纵的数值解法进行了改进，提出了累进求解方法。Botach 等[10]对微分域网格编辑算法等线性算法进行了综述和比较。

3 算法基本思想

Lplacian 网格变形算法的本质是网格模型局部细节特征编码和解码的过程。编码过程是指网格顶点的欧氏空间坐标到 Laplacian 坐标的转换。Laplacian 坐标包含了网格的局部细节特征，因此 Lplacian 网格变形算法能够较好地保持网格模型的局部细节。解码过程是指通过微分坐标反求欧氏空间坐标，实质上是一个求解线性系统的过程。因此 Lplacian 网格变形算法高效、鲁棒。

Lplacian 网格变形方法的关键之处在于计算顶点的 Lplacian 坐标，并根据 Lplacian 坐标的平移不变性通过求解线性系统来获得变形后网格顶点的欧氏空间坐标。由于所求线性系统的系数矩阵是一个稀疏矩阵，因此先将矩阵进行 LU 分解，然后使用迭代法求解，是一个相对高效的求解方法。

Laplacian 坐标的表示对网格细节特征的保持起到了至关重要的作用。文献[6]中 Laplcian 坐标的计算欠缺精确性。下面根据平面多边形中值坐标给出 Laplacian 坐标的精确表示。

3.1 基于切平面中值坐标的 Laplacian 坐标

设闭合三角网格 $M=(V,E,F)$，其中 V 是顶点集合，$V=(v_1,v_2,\cdots,v_n)$，n 表示网格顶点的个数。E 是边集合，F 是面集合。给定顶点 i 的欧氏空间坐标 v_i，其单位法向为 n_i。设 H_i 表示顶点 i 的一阶邻域，$H_i=\{j\mid(i,j)\in E\}$。g_i 表示顶点 i 的度，即集合 H_i 中元素的个数，表示与顶点 i 一阶相邻的顶点的个数。

设 P_i 表示顶点 i 处的切平面，则平面方程为 $n_iv+d_i=0$，其中 $d_i=-n_iv_i$。设与顶点 i 一阶相邻的点为 v_i^k，$k=1,2,\cdots,g_i$，在 P_i 上的投影为 $\tilde{v}_i^k$，到 P_i 的有向距离为 d_i^k，则 $d_i^k=n_iv_i^k+d_i$。

在平面 P_i 上，v_i 的一阶邻域的投影点构成一个多边形，如图 1 所示。根据文献[11]中对于任意多边形所定义的中值坐标，v_i 可以表示为这些投影点的线性加权和，如式(1)所示。

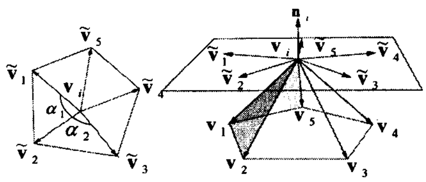

图 1 顶点与其一阶邻域在切平面的投影

$$v_i=\sum_{k=1}^{g_i}\lambda_i^k\tilde{v}_i^k,\text{且}\sum_{k=1}^{g_i}\lambda_i^k=1 \tag{1}$$

系数$(\lambda_i^1,\lambda_i^2,\cdots,\lambda_i^{g_i})$称为顶点 i 的中值坐标。

$$\lambda_i^k=\frac{w_i^k}{\sum_{n=1}^{g_i}w_i^n} \tag{2}$$

其中

$$w_i^k=\frac{\tan(a_{k-1}/2)+\tan(a_k/2)}{\tilde{v}_i^k-v_i} \tag{3}$$

a_k 是指切平面内边$(v_i,\tilde{v}_i^k)$与边$(v_i,\tilde{v}_i^{k+1})$之间的夹角，如图 1 所示。

由于 $\tilde{v}_i^k=v_i^k-d_i^kn_i$，代入式(1)，得

$$v_i=\sum_{k=1}^{g_i}\lambda_i^kv_i^k-n_i\sum_{k=1}^{g_i}d_i^k \tag{4}$$

定义网格 M 上顶点 i 处的 Laplacian 坐标为：

$$\delta_i=n_i\sum_{k=1}^{g_i}d_i^k=v_i-\sum_{k=1}^{g_i}\lambda_i^kv_i^k \tag{5}$$

从式(5)可以看出，δ_i 与顶点 i 处的法向同向，并且$||\delta_i||$为顶点 i 的一阶邻域到其切平面的有向距离之和，与顶点 i 处的平均曲率成正比。

假设 $\lambda_i^1=\lambda_i^2=\cdots=\lambda_i^{g_i}=\frac{1}{g_i}$，正是文献[6]中的 Laplacian 坐标表示形式，如式(6)所示。

$$\delta_i=v_i-\sum_{k=1}^{g_i}v_i^k \tag{6}$$

然而通过这种表示方式对法向和平均曲率进行描述，从一定程度上讲不够确切；现通过一个简单的例子验证，分析如下：

不失一般性，假设顶点 i 与其一阶邻域共面，都处在 XY 平面上，如图 2 所示。

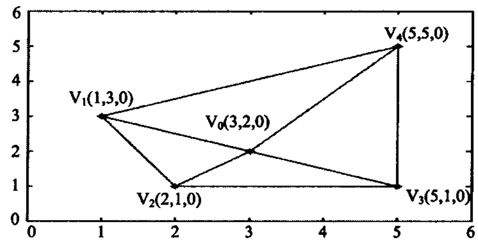

图 2 共面的一阶邻域

通过式(6)计算 v_0 处的 Laplacian 坐标的模大于 0，显然是不确切的，因为平面的曲率为 0。而使用式(5)进行计算能够获得正确的结果。

从欧氏坐标到 Laplacain 坐标的转换可以用矩阵形式表示，设稀疏矩阵 L 为

$$L_{ij}=\begin{cases}1 & i=j\\ -\lambda_i^k & (i,j)\in E\\ 0 & (i,j)\notin E\end{cases}$$

设 $\Delta=(\delta_1,\delta_2,\cdots,\delta_n)$为网格 M 的 Laplacian 坐标集，则有：

$$LV=\Delta \tag{7}$$

L 又称为网格的 Laplacian 算子。

3.2 由 Laplacian 坐标重建网格模型

网格的绘制需要利用网格顶点的欧氏空间坐标，因此必须通过 Laplacian 坐标求解欧氏空间坐标。似乎可以通过求解线性系统式(7)来得到欧氏空间坐标，但事实不可行。原因分析如下：L 不是满秩矩阵，对于连通网格，L 的秩为 $n-1$，因此式(7)没有唯一解。

网格变形是一个交互的过程，在交互过程中用户往往指定一些锚点(固定点)，再指定一些控制点。根据这些已知点和 Laplacian 坐标，以及 Laplacian 坐标的平移不变性，对于平

移变形操作可以构建能量函数：

$$E(V')=||LV'-\Delta||+\sum_{i=1}^{m}||v_i'-u_i|| \quad (8)$$

使式(8)取最小值，其意义可以理解为变形后网格的 Laplacian 坐标逼近于原始网格的 Laplacian 坐标，变形后网格上的指定点位置逼近于用户指定的位置。对能量函数求最小值，从变分的观点可以转换为求解下面的线性系统。

$$\left(\frac{L}{0|I}\right)V' = \left(\begin{matrix}\Delta \\ u_1:u_m\end{matrix}\right) \quad (9)$$

对于式(9)这样的线性系统，通常的方法是先对系数矩阵进行 LU 分解，然后用迭代法进行求解。

对于旋转变形，需要对原始网格的 Laplacian 坐标进行一定的修正，为每一个顶点隐式地添加一个变换矩阵 T_i，详见文献[7]。

4 实验与分析

对算法进行了实验。硬件环境为普通 PC，PⅣ2.8G，显卡 Geforce 8600，256M 显存，1G 内存。软件环境以 Windows XP 作为操作系统，VC.net 2003 作为开发平台，使用 OpenMesh，UMFPACK 和 CUDA（Compute Unified Device Architecture，统一计算设备架构）作为开发包。

OpenMesh 用于网格对象的操作，该开发包使用的是三角网格的半边结构，方便求得顶点的一阶邻域（点、边、三角面片）。CUDA 是 nVidia 公司对其 Geforce 8 系列显卡发布的用于 GPGPU（General Purpose Computation on Graphic Hardware，GPU 通用计算）技术的开发包，使得 GPGPU 技术更加透明。UMFPACK 主要用于稀疏线性系统的求解。实验过程中高效地结合 CUDA 中的 CUBLAS 基本线性代数算法库和 UMFPACK 的 LU 分解求解，提高了算法的执行效率，实验结果如表 1 所列。从表中可以看出，系数矩阵的 LU 分解占去了大量的时间。但是对于固定的控制点和锚点，式(9)的系数矩阵是相同的，因此只需要进行 1 次分解即可。由于求解的速度迅速，因此基本可以满足实时性的要求。

表 1 算法执行时间统计

Model	Vertices	Trangles	LU Factor(s)	Solve(s)
Armadillo	2767	5518	0.153	0.171
Cow	3066	5806	0.207	0.185
Dragon	4303	8586	0.495	0.213
Venus	19755	43357	3.164	0.828
Bunny	34834	69451	10.725	1.539

下面给出使用该算法的一些编辑效果图，如图 3 和图 4 所示。

图 3 对 Bunny 模型进行形变：(a) 原始 Bunny 模型；(b)拉长左耳朵；(c)拉长右耳朵；(d)最终编辑结果

图 4 对 Dragon 模型进行形变：(a) 原始 Dragon 模型；(b)抬高上颚；(c)拉长下颚；(d)最终编辑结果

从图中可以看出，Bunny 模型的耳朵和 Dragon 模型的嘴部都很好地保持了形状特征。由此可以看出本算法能够获得比较好的编辑效果。

结束语 本文对 Laplacian 网格变形算法进行了细致分析；在此基础上根据平面多边形中值坐标对 Laplacian 坐标进行了定义；根据顶点一阶邻域在其切平面上的投影点推导出的 Laplacian 坐标能够比较准确地表示网格模型的几何细节特征。实验表明，本算法能够获得较好的编辑效果。Laplacian 网格变形算法能够保持几何细节特征，并且计算过程为求解线性系统，因此算法执行高效、鲁棒。随着计算机硬件水平的提高及 GPGPU 技术的广泛应用，下一步主要考虑如何使用非线性方法及简单直观的交互方式获得更加精细的编辑效果。

参考文献

[1] Sederberg T W，Parry S R. Free-form Deformation of Solid Geometric Models[C]//Siggraph'86. ACM Comp. Graph，1986

[2] Guskov I，Sweldens W，SCHRÖDERP. Multiresolution signal processing for meshes[C]//Proceedings of ACM SIGGRAPH' 99. ACM Press，1999

[3] Guskov I，Vidimce K，Sweldens W. Normal meshes[C]//Proceedings of ACM SIGGRAPH. 2000

[4] Kobbelt M，Botsch L. Multiresolution surface representation based on displacement volumes[J]. Computer Graphics Forum，2003，22(3)：483-491

[5] Yu Y，Zhou K，Ku D，et al. Mesh editing with poisson-based gradient field manipulation [C] // Proceedings of ACM SIGGRAPH'04. ACM Press，2004

[6] Lipman Y，Sorkine O，Levin D. Differential Coordinates for Interactive Mesh Editing[C]//Proceedings of Shape Modeling International. IEEE Computer Society Press，2004

[7] Sorkine O，Lipman Y，Cohen-OR D，et al. Laplacian surface editing[C]//Proceedings of the Eurographics/ACM SIGGRAPH Symposium on Geometry Processing. ACM Press，2004

[8] Zhou Kun，Huang Jin，Snyder J，et al. Large mesh deformation using the volumetric graph Laplacian[C]//Proceedings of ACM SIGGRAPH'05. 2005

[9] Xu Dong. Differential domain mesh processing technology [D]. Zhejiang：Zhejiang University，2006

[10] Botsch M，Sorkine O. On linear variational surface deformation methods[J]. IEEE Transactions on Visualization and Computer Graphics，2008，14(1)：213-230

[11] Kai H，Michael S. Mean Value Coordinates for Arbitrary Planar Polygons[J]. ACM Transactions on Graphics，2006，25(4)：1424-1441