

# 角度约束路径法的网格曲面兴趣区域边界快速交互选取

舒孝阳, 刘斌

(华侨大学 数字化视觉测量厦门市重点实验室, 福建 厦门 361021)

**摘要:** 提出角度约束路径法,快速获取三角网格曲面上任意两顶点间一条由网格边所组成的路径.该算法是一个从起始点开始不断向前传播的过程,计算量仅与两顶点间的曲面区域有关,故算法的时间复杂度( $O(n)$ )优于 Dijkstra 算法( $O(n \log n)$ ).试验结果表明:角度约束路径法的执行快速、有效;基于该方法可实现三角网格曲面兴趣区域边界的快速交互选取.

**关键词:** 兴趣区域;角度约束;交互;三角网格;边界选取

**中图分类号:** TP 391 **文献标志码:** A

兴趣区域(ROI),是指曲面上用户交互指定的一块操作区域.曲面上兴趣区域的交互选取广泛应用于网格融合<sup>[1]</sup>、网格分割<sup>[2]</sup>、网格建模<sup>[3]</sup>、特征重用<sup>[4]</sup>等领域.解决该问题的目标是快速、简单、有效地得到所需的兴趣区域边界.对兴趣区域的交互选取研究,主要有 3 种典型的方法.1) 基于 Dijkstra 最短路径<sup>[5]</sup>的方法<sup>[6-7]</sup>.这类方法的时间复杂度至少为  $O(n \log n)$ ,  $n$  为三角网格模型顶点数.2) 基于割集分割策略的方法<sup>[8-9]</sup>.由于该方法需要判断切平面  $T_p$  和网格模型上所有三角面片的距离是否小于给定阈值,该方法的时间效率与网格模型的规模呈负相关,即网格规模越大,时间效率越低.3) 基于智能剪切(intelligent scissoring)操作的方法<sup>[10-11]</sup>.该方法可将网格剪切成用户可感知的部分,然而该方法涉及到曲率计算,算法效率不高.因此,为了快速、有效地交互选取用户所需的兴趣区域边界,本文提出角度约束路径法,用于快速寻找网格曲面上两点间的一条路径.

## 1 角度约束路径法

三角网格曲面兴趣区域边界选取实例,如图 1 所示.其中,黑色线为该算法生成的边界实例.由图 1 可以看到:用户通过鼠标在屏幕上交互选定一系列二维点;然后,将这些二维点转化成三角网格表面上的顶点(黑色点);最后,采用角度约束路径法分别找出相邻两顶点间的路径,得到的总路径便是网格曲面上选取区域的边界.

### 1.1 算法的实现

角度约束路径法是一个从起始点  $S$  开始不断向前传播的过程.算法分为当前点判断和当前点更新两部分.该算法的伪代码为

输入:网格曲面  $M$ 、起始点  $S$ 、目标点  $T$ ;  
输出:两点间的路径点集  $Path = \{Pt_i | i = 0, 1, \dots, n\}$ ;  
1) 当前点  $Pt_i \leftarrow S$ ;

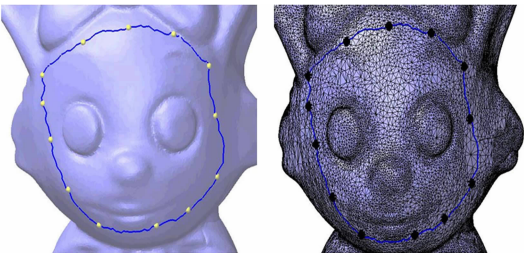


图 1 三角网格曲面兴趣区域边界选取实例  
Fig.1 Selected example of interest region boundary on triangular mesh surface

- 2) 将点  $S$  存入点集  $Path$  中;
- 3) While 当前点  $P_{t_i}$  不在  $T$  附近;
- 4) 更新当前点  $P_{t_i}$ ;
- 5) 将  $P_{t_i}$  存入点集  $Path$  中;
- 6) 结束 While;
- 7) 算法结束.

其中, 当前点判断用以判断  $P_{t_i}$  是否为目标点  $T$ , 若  $P_{t_i}$  为目标点  $T$ , 则算法停止. 当前点更新的目的是在当前点  $P_{t_i}$  的 1-ring 邻域顶点中, 寻找一个偏离实际测地线方向(文中算法采用  $P_{t_i}$  和目标点  $T$  之间的连线方向代替)最小的点. 算法执行如图 2 所示, 具体为 3 个步骤.

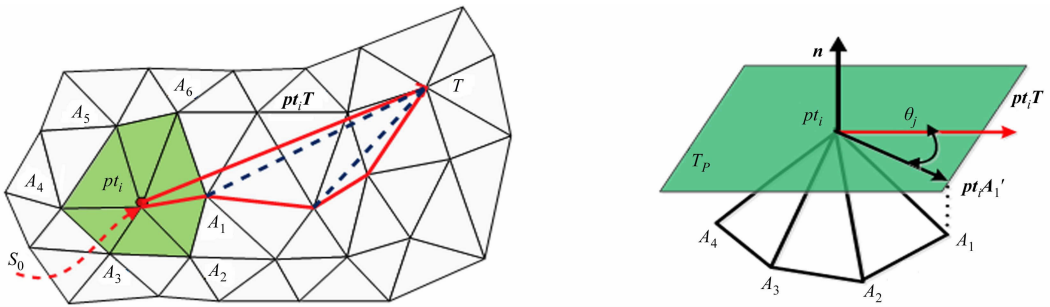


图 2 算法执行示意图

Fig. 2 Schematic of algorithm execution

- 1) 设  $Path$  传播到当前点  $P_{t_i}$  处, 寻找点  $P_{t_i}$  的 1-ring 邻域上的顶点  $A_j$  (设顶点的个数为  $m$ , 则  $j = 0, 1, \dots, m$ , 下同), 计算  $P_{t_i}$  点处的法矢为  $\mathbf{n}$ , 并建立  $P_{t_i}$  点的切平面  $T_p$ .
- 2) 分别将所有向量  $\mathbf{P}t_iA_j$  与向量  $\mathbf{P}t_iT$  分别投影到切平面  $T_p$  上, 并且单位化上述向量, 得到对应的投影向量  $\mathbf{P}t_iA'_j, \mathbf{P}t_iT'$ .
- 3) 分别计算  $\cos \theta_j, \theta_j$  为  $\mathbf{P}t_iA'_j, \mathbf{P}t_iT'$  之间的夹角. 保存当  $\cos \theta_j$  为最大值  $(\cos \theta_j)_{\max}$  时, 所在的顶点为  $B_1$  和  $\delta_1 = (\cos \theta_j)_{\max}^1$ , 并计算余下顶点的  $\omega_k$  值. 其中:  $\omega_k = (\cos \theta_j)_{\max} - \cos \theta_k, k = 0, 1, \dots, m-1$ . 此时, 根据  $\omega_k$  值的不同进行不同的处理, 设  $\zeta$  为给定的阈值.
  - I) 当  $\omega_k < \zeta$  且  $\omega_k$  取最小值时, 保存此时的顶点为  $B_2$ ,  $\cos \theta_j$  值为  $\delta_2$ , 分别求解  $P_{t_i} = B_s, s = 1, 2$  时, 所得到的  $(\cos \theta_j)_{\max}^s$ . 令  $t_1 = (\cos \theta_j)_{\max}^1 + \delta_1, t_2 = (\cos \theta_j)_{\max}^2 + \delta_2$ , 取  $P_{t_i} = B_s$  (若  $t_1 < t_2, s = 1$ ; 否则  $s = 2$ ), 并将点  $P_{t_i}$  存入点集  $Path$  中.

然后, 判断  $P_{t_i} = T$  是否为真. 若为真, 算法停止, 点集  $Path$  便为所求路径上的所有顶点; 否则, 返回步骤 1), 继续循环.

II) 直接取  $P_{t_i} = B_1$ , 并将点  $B_1$  存入点集  $Path$  中, 判断  $P_{t_i} = T$  是否为真. 若为真, 则算法停止, 点集  $Path$  便为所求路径上的所有顶点; 否则, 返回步骤 1), 继续循环.

由于该算法仅与  $S, T$  两顶点间的局部区域有关, 算法的时间复杂度小于  $O(n)$ ,  $n$  为网格模型顶点的个数. 因此, 可用于快速交互选择三角网格曲面上的兴趣区域.

## 1.2 算法的改进

在算法的实际执行过程中发现, 当两点间的曲面区域较平坦时, 采用上述方法, 能得到较直的一条路径; 然而, 当两点间的曲面区域较陡峭时, 获得的路径效果较差, 如图 3(a) 所示. 这是由于角度约束路径法是基于两点间连线方向进行角度约束, 当两点间的曲面区域较陡峭时, 两点间连线方向和实际测地线方向相差较大. 因此, 文中对该算法进行了改进, 即在两点间插入 1 个过渡点, 使两点间的曲面区域趋于平坦, 如图 3(b) 所示.

设网格曲面  $M$  上的顶点集合为  $V$ , 两点为  $p, q$ . 对于过渡点的确定, 首先, 以顶点集合建立 K-D 树; 然后, 采用沿给定方向向量搜索距离该方向向量最近的顶点. 方向向量  $\mathbf{dir}$  的确定, 如图 4 所示. 设  $\mathbf{n}_p, \mathbf{n}_q$  分别为  $p, q$  两点的法矢, 则将  $\mathbf{pq}$  沿着  $\mathbf{n}_q \times \mathbf{n}_p$  轴逆时针旋转  $90^\circ$  得到的向量便是  $\mathbf{dir}$ .

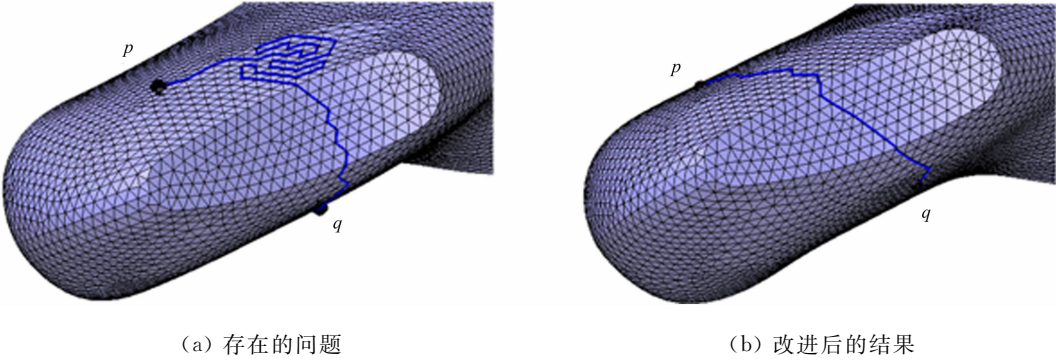


图 3 角度约束路径法存在的问题及其改进示意图

Fig. 3 Schematic diagram of existing problems and improvement results of angular constraint path method

2 兴趣区域边界获取

基于角度约束路径法可实现网格曲面上兴趣区域边界的快速获取,具体步骤为

- 1) 用户根据自己的需要,交互选取一些二维点,并保存之;
- 2) 将上述二维点投影到网格曲面上,得到距离各投影点最近的网格顶点,并保存之;
- 3) 采用角度约束路径法得到任意相邻两点间的一条路径,保存路径上的顶点.此外,还需要得到首末位置顶点间的一条路径,取出路径上的重复点,最终得到一条封闭的兴趣区域边界,如图 5 所示.

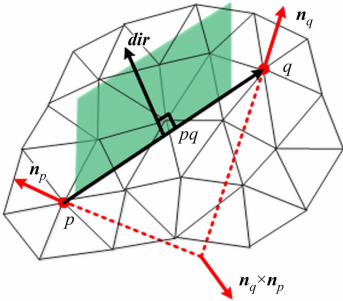


图 4 方向向量  $dir$  的确定

Fig. 4 Determination of the direction vector  $dir$

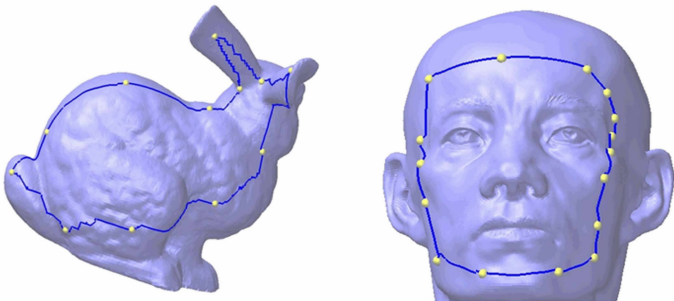


图 5 三角网格曲面兴趣区域边界选取实例

Fig. 5 Selected example of interest region boundary on triangular mesh surface

3 实验验证

文中算法是基于 Microsoft Visual Studio 2008 开发工具和 OpenGL 函数库编写的. 为了验证角度约束路径算法的有效性和时间效率,在 CPU 为 2.26 GHz AMD Athlon(tm) II X2 240 Processor,内存为 2.0 GB 的 PC 机上运行该算法.

由上述可知:该算法的时间复杂度小于  $O(n)$ . 因此,本算法的时间效率应优于 Dijkstra 最短路径法(时间复杂度为  $O(n \log n)$ ). 为了验证文中算法的时间效率,分别采用角度约束路径法和 Dijkstra 最短路径法测试鞋楦模型、斯坦福兔子模型和头像模型上选定的路径点,2 种算法顶点位置选择相同,得到的实验数据如表 1 所示. 由表 1 可知:角度约束法具有运算速度快的特点,适用于交互选取三角网格曲面上的兴趣区域边界.

表 1 角度约束路径法和 Dijkstra 最短路径法时间效率对比

模型	模型顶点数	路径顶点个数	Dijkstra 最短路径法耗时/s	角度约束路径法耗时/s
鞋楦	4 263	6	0.245	0.188
斯坦福兔子	35 181	12	2.701	0.828
头像	83 920	16	7.480	1.826

4 结束语

文中提出角度约束路径法,用以实现三角网格曲面兴趣区域边界的快速交互选取. 然而,在算法的实际执行过程中发现,当相邻两点间曲面区域较陡峭时,得到的路径效果较差. 针对这一问题,采用在两点间插入一个过渡点的方法,从而使相邻两点间的曲面重新趋于平坦. 试验结果表明:改进后的角度约束路径法获取的路径效果好. 不仅如此,对比 Dijkstra 最短路径法,该方法具有算法执行快速快的特点,适用于兴趣区域的快速交互选取.

参考文献:

[1] JUNGE K, BINNEBÖSEL M, ROSCH R, et al. Impact of proinflammatory cytokine knockout on mesh integration [J]. *Investigative Surgery*, 2009, 22(4): 256-262.

[2] 孙晓鹏, 李华. 三维网格模型的分割及应用技术综述[J]. *计算机辅助设计与图形学学报*, 2005, 17(8): 1647-1655.

[3] KOBBELT L, CAMPAGNA S, VORSATZ J, et al. Interactive multi-resolution modeling on arbitrary meshes[C]// *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*. New York: ACM, 1998: 105-114.

[4] SCHMIDT R, SINGH K. Drag, drop, and clone: An interactive interface for surface composition[R]. Toronto: University of Toronto, 2010: 1-10.

[5] DIJKSTRA E W. A note on two problems in connexion with graphs[J]. *Numerische Mathematik*, 1959, 1(1): 269-271.

[6] SCHMIDT R, SINGH K. Sketch-based procedural surface modeling and compositing using surface trees[J]. *Computer Graphics Forum*, 2008, 27(2): 321-330.

[7] SHARF A, BLUMENKRANTS M, SHAMIR A, et al. SnapPaste: An interactive technique for easy mesh composition[J]. *The Visual Computer*, 2006, 22(9/10/11): 835-844.

[8] KHO Y, GARLAND M. Sketching mesh deformations[C]// *Proceedings of the 2005 Symposium on Interactive 3D Graphics and Games*. New York: ACM, 2005: 147-154.

[9] 王隼, 张宏鑫, 许栋, 等. 勾画式泊松网格编辑[J]. *计算机辅助设计与图形学学报*, 2006, 18(11): 1723-1729.

[10] FUNKHOUSER T, KAZHDAN M, SHILANE P, et al. Modeling by example[C]// *ACM Transactions on Graphics*. New York: ACM, 2004: 652-663.

[11] LEE Y, LEE S, SHAMIR A, et al. Mesh scissoring with minima rule and part salience[J]. *Computer Aided Geometric Design*, 2005, 22(5): 444-465.

Fast Interactive Selection of Interest Region Boundary on Mesh Surface Based on Angular Constraint Path Method

SHU Xiao-Yang, LIU Bin

(Xiamen Key Laboratory of Digital Vision Measurement, Huaqiao University, Xiamen 361021, China)

**Abstract:** In this paper, an angular constraint path method is presented, which can get a path composed of mesh edges rapidly between any two vertices on triangular mesh surface. The algorithm is a continuous transmission process from its starting point. The amount of calculation is only related to the surface area between two vertices, so the time complexity of the algorithm ( $O(n)$ ) is superior to Dijkstra algorithm ( $O(n \log n)$ ). Experimental results show that the performance of angular constraint path method is quickly, effectively. And fast interactive selection of interest region boundary on triangular mesh surface is realized based on this method.

**Keywords:** interested region; angle constraint; interaction; triangular mesh; boundary selection