

基于流线的重新网格化及多分辨率表示*

熊赞晖^{1,2+}, 李桂清¹, 韩国强¹, 彭莉¹

¹(华南理工大学 计算机科学与工程学院, 广东 广州 510640)

²(华南理工大学 理学院, 广东 广州 510640)

Remeshing and Multiresolution Modeling Method Based on the Flow Line

XIONG Yun-Hui^{1,2+}, LI Gui-Qing¹, HAN Guo-Qiang¹, PENG Li¹

¹(School of Computer Science and Engineering, South China University of Technology, Guangzhou 510640, China)

²(College of Science, South China University of Technology, Guangzhou 510640, China)

+ Corresponding author: E-mail: yhxiong@scut.edu.cn

Xiong YH, Li GQ, Han GQ, Peng L. Remeshing and multiresolution modeling method based on the flow line. *Journal of Software*, 2008,19(Suppl.):131-142. <http://www.jos.org.cn/1000-9825/19/s131.htm>

Abstract: This paper proposes a remeshing and multiresolution reconstruction approach based on Laplacian fields. The method starts with the establishment of Laplacian field on an original mesh, and then generates two groups of flow lines with 60° rotation. Finally, a rhombus-dominant mesh is constructed and triangulated into a triangular base-mesh. The method is used again to upsample the original mesh by refining the base-mesh model in order to produce a multiresolution representation. In general, the triangles of the resulting meshes are close to be equilateral. Experiments also show that the method may yield results of high quality.

Key words: remeshing; multiresolution; flow line; Laplacian field

摘要: 提出基于场的重新三角网格化和多分辨率表示生成算法.首先,在原三角网格模型上建立拉普拉斯标量场,据此生成两组夹角为 60°的流线.然后,从这两组流线构造以菱形面为主的网格并三角化得到三角基网格.最后,在基网格的基础上,再次使用流线技术对原始数据进行向上重采样,得到模型的多分辨率表示.一般地,基网格的三角形接近等边三角形,实验也表明该方法能够得到较高质量的结果.

关键词: 重新网格化;多分辨率表示;流线;拉普拉斯场

重新网格化(remeshing)是数字几何处理的一项重要内容,在 CAD 建模、快速原型制造和动画等领域有广泛应用.从三维激光扫描仪或其他硬件设备获得的数据中通常含有多余的顶点、边、面、不规则的多边形和噪声,这些都给几何模型的后续处理带来困难.严重时会导致数值解不稳定,影响对模型的正确分析与处理,因此,需要预先对这些原始数据进行加工,以获取高质量的模型.重新网格化为此类处理提供了一条有效的途径.此

* Supported by the National Natural Science Foundation of China under Grant Nos.60673005, 60573019 (国家自然科学基金); the Natural Science Foundation of Guangdong Province of China under Grant No.05006540 (广东省自然科学基金); the Open Fund of the State Key Laboratory of CAD & CG, Zhejiang University of China (浙江大学 CAD&CG 国家重点实验室开放基金); the Natural Science Foundation for the Youth of South China University of Technology under Grant No.B13E5060120 (华南理工大学自然科学基金)

Received 2008-05-03; Accepted 2008-11-14

外,重新网格化也是流体模拟、网格编辑、曲面重构、逆向工程、模型压缩和拼花图案装饰等应用的重要基础.

1 背景及相关工作

重新网格化可追溯到 Turk 的重新布点简化算法^[1],迄今国际上已积累大量相关文献^[2],国内刊物也有一些研究报道^[3-5].从所用的方法来看重新网格化大致可归为参数化和非参数化两类.从重新网格化的结果来看则有三角形、纯四边形网格及以四边形网格为主的 4-3 网格.

参数化方法把网格模型映射到规整的参数域,然后对参数域进行重新网格化,并映射回原网格得到重采样结果.对于封闭或亏格大于零的模型,参数化方法需要对模型进行分割,会导致曲面片扭曲和拼接等问题^[6-9],因此引出一系列改进工作^[10-12].

非参数化方法则直接在流形上重新网格化,往往需要在能够较好地反映模型结构的某种场的指导下对原网格进行切割^[13-17].如 Marinov 和 Kobbelt^[11]直接在流形上计算曲率张量场,根据曲率极值分布来生成四边形占优的网格;Dong 等人^[13]用调和标量场得到正交的梯度场和等参场,生成两组正交流线后得到四边形为主的网格,在文献^[14]中,他们进一步对面上的标量场进行分析,通过谱分解得到高质量的纯四边形网格,但仍须对四边形片作参数化以获得多分辨率表示;Tong 等人^[15]用离散调和形式得到一对标量场,得到两组正交等值线后生成新网格;Ray 等^[16]使四边形网格在参数域上正交于给定的向量场.

多分辨率表示在模型显示、简化、压缩、传输、层次细节表示、编辑和变形等应用中占有重要地位^[18].根据新顶点增加方式不同可分两类:一类基于细分连通网格,如 Eck 等人的三角网格模型多分辨率分析(ECK 算法)^[19];另一类可看成是网格简化的逆过程,相邻分辨率间只增加少量顶点和面,如 Hoppe 的渐近网格表示^[20].本文中考虑基于细分连通的多分辨率表示.

已有基于场的方法都是用来构造四边形或四边形占优网格的,在生成多分辨率表示时必须作参数化.本文利用拉普拉斯场来作重新三角网格化是一种新尝试;利用流线生成方法直接构造多分辨率表示,整个过程无须参数化,这是本文的另一个主要贡献;重采样时也不是引法向求交,而是采用一种网格上的步进求交方式,因此不存在误交现象.

2 基于流线的重新网格化

Dong 等人^[13]通过两组正交的流线来生成以四边形为主的多边形网格.受此启发,若构造两组夹角 60° 的流线则可以得到较规则的三角形基网格.假设输入的原始网格 M 为三角网格模型,重新网格化及多分辨率表示算法流程可分成 5 个步骤(如图 1 所示):

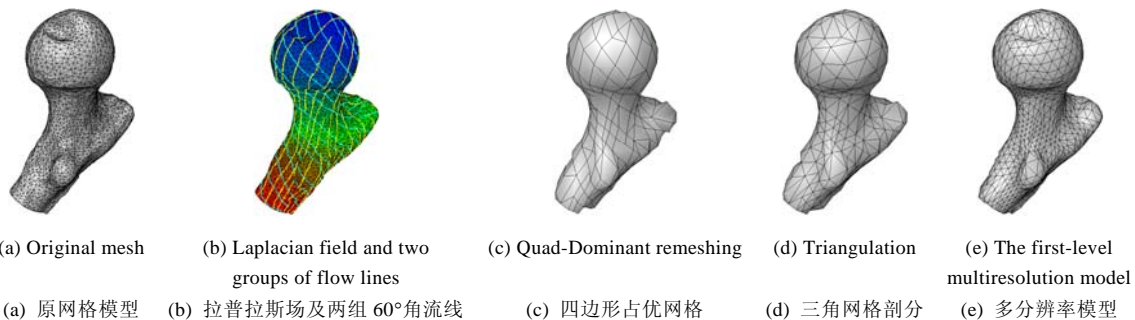


Fig.1 Remeshing flow chart

图 1 重新三角化

- (1) 在二维流形 M 上指定极大、极小值点,建立拉普拉斯场,并得到梯度场.
- (2) 选取梯度方向及与之夹角为 60° 方向得到两组方向场,生成两组夹角为 60° 的流线.
- (3) 计算两组流线的交点,得到菱形四边形为主的多边形网格.

(4) 将菱形四边形中为钝角的两顶点相连,得到两个三角形,每个顶点内角接近于 60° .

(5) 对剩余的 T 型域和其他多边形进行三角剖分,得到三角基网格.

2.1 拉普拉斯场的计算

设二维流形 M 上有 n 个顶点,以序号 $1, 2, \dots, n$ 进行标记,又 u 是定义在 M 上的标量函数,称为 M 上的标量场.记 u 在 M 的第 i 个顶点上的函数值为 u_i ,可定义函数 u 在 M 上各顶点处的离散拉普拉斯算子 Δ 如下:

$$\Delta u_i = \sum_{j \in N_i} w_{ij} (u_i - u_j) = \sum_{j \in N_i} \frac{1}{2} (\cot \alpha_{ij} + \cot \beta_{ij}) (u_i - u_j) \quad (1)$$

其中 N_i 为顶点 i 的 1-邻域所有顶点的集合, w_{ij} 表示 i 的邻接顶点 j 的权重,满足 $\sum_{j \in N_i} w_{ij} = 1$ 且 $w_{ij} \geq 0, \cot \alpha_{ij}, \cot \beta_{ij}$ 为边 ij 相邻三角形的两个对角的余切.如果 1-邻域三角形中包含有钝角三角形,则式(1)中会出现负系数.这种情况会导致后面 Laplace 线性方程的求解不稳定,而且也无法控制极值点的个数.因此式(1)只适用于 1-邻域中的所有 α_{ij} 和 β_{ij} 均为锐角的情况.对于存在钝角的情形,需要按如下方式修改将公式(1)中的 w_{ij} , 以保证 $w_{ij} \geq 0$:

$$w_{ij} = \frac{\lambda_{ij}}{\sum_{k \in N_i} \lambda_{ik}}, \quad \lambda_{ij} = \frac{\cot(\alpha_{ij}/2) + \cot(\beta_{ij}/2)}{\|u_j - u_i\|} \quad (2)$$

需要指出,公式(2)在文献[13]中采用正切(tan)而不是余切(cot).通过实验发现,当顶点 i 的函数值为极值时,在附近生成的等值线为椭圆而不是圆,故我们改回余切.

最后,记 $\mathbf{u} = [u_1, u_2, \dots, u_n]^T$, 且 $\Delta \mathbf{u} = [\Delta u_1, \Delta u_2, \dots, \Delta u_n]^T$, 则

$$\Delta \mathbf{u} = -L\mathbf{u}, \quad \text{其中}, L_{ij} = \begin{cases} 1, & i = j \\ -w_{ij}, & j \in N_i \\ 0, & \text{其他} \end{cases} \quad (3)$$

现在假定通过交互方式指定了 M 上某些顶点的场值,这些顶点称为控制顶点(边界点一律被设为控制点).此外,其他顶点均满足 Laplace 方程 $\Delta u = 0$,则可得到如下线性方程组:

$$A\mathbf{u} = \mathbf{B} \quad (4)$$

其中, $A_{ij} = \begin{cases} 1, & i=j \text{ 且 } u_i \text{ 为控制顶点} \\ 0, & i=j \text{ 且 } u_i \text{ 为控制顶点} \\ L_{ij}, & \text{其他} \end{cases}, \quad B_i = \begin{cases} c_i, & i \text{ 为控制顶点} \\ 0, & \text{其他} \end{cases}$.

由线性系统公式(4)所确定的函数 u 称为拉普拉斯场.

2.2 拉普拉斯场的梯度场

对于流形上的标量场,总可以在模型的每个三角形上定义一个梯度向量,从而得到梯度场.求出 Laplace 标量场后,文献[13]中通过解一个线性方程组来计算梯度场.当 3 个顶点场值相同的时,这种方法可能会出现奇异解.这里我们采用直接计算方式.根据三角形 3 个顶点的场值,分 3 种情况进行求解.不妨设三角形的 3 个顶点为 1, 2, 3, 它们的三维坐标依次为 p_2, p_1, p_3 .

1) 3 个顶点函数值互不相同情形.不失一般性,设 $u_2 < u_1 < u_3$, 显然边(2,3)上有一个点与顶点 1 有相同的场值,记此点为 p_0 , 则有(这里实际上用顶点函数值作线性插值来近似三角形边上和内部的场值)

$$p_0 = \frac{(u_1 - u_2) \times p_3 + (u_3 - u_1) \times p_2}{(u_3 - u_2)}.$$

例如,图 2(a)中分别为 $u_1=0.2, u_2=0.1, u_3=0.4$, 因此 $p_0=(2p_2+p_3)/3$. 线段 p_0p_1 就是三角形上的一条近似等参线.由于梯度方向与等参线垂直,将等参线旋转 90° 后选择函数值增大的方向,即可得到三角形上的梯度方向.

2) 两顶点函数值相同而与第 3 个顶点不同情形.两函数值相同顶点的边可看作为等参线,梯度方向就是垂直这条边且沿函数值增大的方向,如图 2(b)所示.

3) 3 个顶点的函数值均相同,这种情况比较少见.可用相邻的 3 个三角形上的梯度场加权平均得到,如图 2(c)所示.这个方法既简单又可避免解方程组方法中矩阵求逆出现问题.

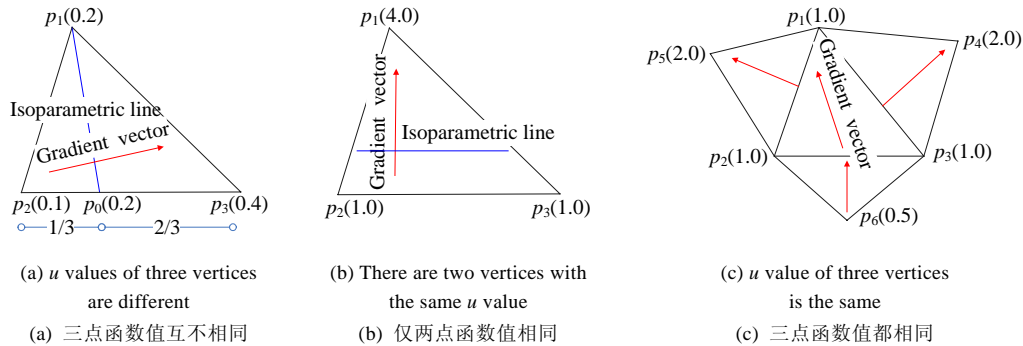


Fig.2 Determination of the gradient vector of a triangle

图 2 三角形梯度向量的确定

在模型表面每个三角形上都计算梯度场(图 3 中的横向左箭头线),可将选取梯度场逆时针旋转 90° 方向作为等参场的方向(图 3 中的纵向下箭头线).

2.3 流线生成

由于作三角重新网格化而不是四边形网格化,我们需要构造两组夹角为 60° 的流线,因此这里不用等参场.首先选取梯度场方向生成一组流线,称为 u 线(图 4 中的水平深色流线).然后将梯度场方向旋转 60° 得到另外一组流线,称为 v 线(图 4 斜浅色流线).事实上,流线夹角可以是任意值,为了生成接近等边三角形的基网格而取 60° .

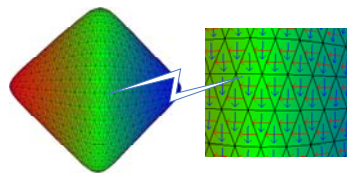


Fig.3 Gradient and isoparametric vector fields

图 3 梯度场和等参场

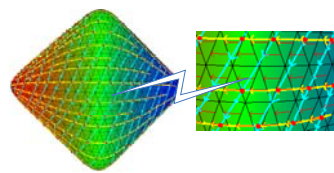


Fig.4 Two groups of flow lines

图 4 两组流线

2.3.1 流线生成步骤

整个流线生成分成 5 个步骤(如图 5 所示):

- 1) 获取所有极小值点作为初始的 u 线种子点,任选其一,生成第 1 条 u 线,见附录 1 算法 2.
- 2) 由这条 u 线得到其他 u 线的种子点和 v 线种子点.做法是:在 u 线上根据 h_2 的间隔放置种子点,这部分种子点加入到 v 线种子点集合中;在 u 线的两侧距离为 h_1 的位置放置种子点,这部分种子点既加入到 u 线种子点集合,同时也加入到 v 线种子点集合中.用重心坐标计算这些种子点的函数值,并按函数值将 u 线(v 线)种子点集合从小到大排序.这里 h_1 和 h_2 与几何形状有关,具体来说^[13],

$$h_1 = \frac{h}{1 + \alpha \log_{10}(1 + k_n^1)}, \quad h_2 = \frac{h}{1 + \alpha \log_{10}(1 + k_n^2)} \quad (1)$$

其中, h 和 α 为常数,调节距离的大小; k_n^1, k_n^2 分别表示 v 线和 u 线方向的法曲率.这样可以保证较平坦的区域步长大,而较弯曲的区域步长小.

- 3) 因两组流线夹角为 60° ,必须在极小值点处再生成 $360^\circ/60^\circ = 6$ 条 u 线,也就是说,每个极小值点附近都有 6 条 u 线,生成这些 u 线后同样也得到若干 u 线(v 线)种子点.

- 4) 所有极小值点附近的 u 线生成完成后,逐个从 u 线种子点队列前面取出函数值最小的种子点生成新 u

线,并以公式(5)中的 h_1 作为阈值对流线进行检测,控制 u 流线的生成密度.重复步骤 2)得到新的 u 线和 v 线种子点,计算这些种子点的函数值并重新排序.直至所有 u 线种子点集合为空.

5) u 线生成结束后再开始 v 线生成,见附录 1 中算法 3.每次选择函数值最小的 v 线种子点生成 v 线,并以公式(5)中的 h_2 作为阈值对流线进行检测来控制 v 流线间的密度,直至不能生成新的 v 线为止.

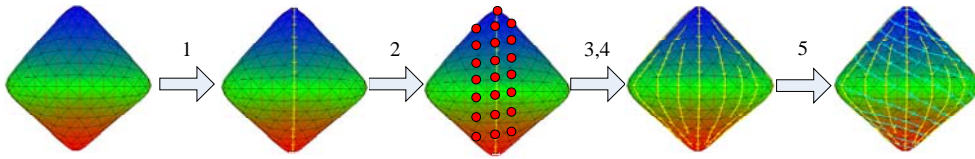


Fig.5 Flowchart for generating the two groups of flow lines

图 5 两组流线的生成步骤

流线的起点有极小值点和种子点,到达一个极大值点或满足终止条件则停止.极小值点是顶点函数值小于相邻所有顶点函数值(图中最顶端的点),极大值点指的是顶点函数值大于相邻所有顶点函数值(图中最低端的点).流线的生成过程中, u 线(或 v 线)间不能相交,终止条件由公式(1)来控制, u 线碰撞检测距离为 h_1 , v 线碰撞检测距离为 h_2 .

2.3.2 极值点附近的 v 线

与正交流线的 v 线不同的是夹角为 60° 的 v 线起点与终点不封闭,尤其在极值点附近处理起来更困难.这部分区域的处理好坏会极大地影响最终网格化的效果.本文采取一种折衷方案,先在极值点附近生成一条与梯度场成 90° 的封闭 v 线(图 6),以便阻止其他 v 流线进入到极值点附近,从而降低问题的难度.

2.4 新网格的构成

得到两组夹角为 60° 的流线之后,接下来要计算每个三角面上这两组流线的交点(图 7(a)),见附录 1 中算法 4.并确定这些交点间的前后左右连接关系,然后根据这些拓扑关系构造以菱形状为主的多边形网格(图 7(b)),见附录 1 中算法 5.最后是生成三角网格,此即为构造多分辨率所需的基网格.因极值点处的流线分布十分复杂,多边形网格的构造难度要比两组正交流线情形^[13]大许多.从多边形网格到三角网格比较简单,其中的三角形面不需要特别处理,其他面则分成菱形状四边形、T 型域和一般多边形 3 种类型分别处理(图 7(c)).

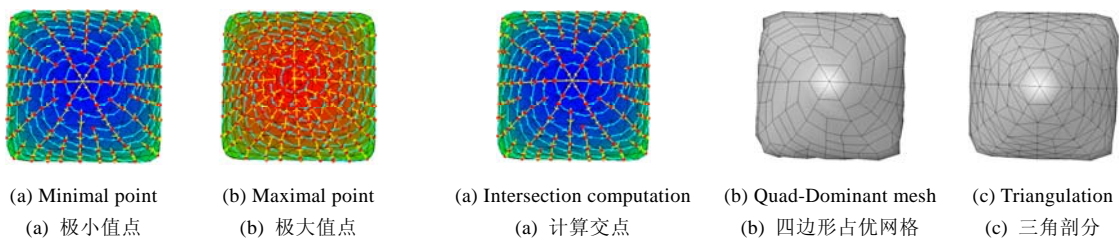


Fig.6 Flow lines near an extreme point

图 6 极值点附近的流线

Fig.7 Generation of the new mesh

图 7 新网格生成

- 1) 将菱形状的四边形钝角的两个顶点相连,构成两个三角形;
- 2) 将 T 型域分成三角形和四边形处理,四边形再按菱形处理;
- 3) 多边形区域通常出现在极值点附近,且不一定就是凸的,还可能会是凹的,其处理相对复杂些.我们采用了一种贪婪算法,每次选择最接近 60° 的顶点,将其与相连的两个顶点构成一个三角形,依次循环直至多边形剩下 4 个顶点,再按菱形处理.

构造流线时,可适当延长流线的终点或反向延长流线的起点,以尽可能减少多边形区域.

3 多分辨率表示生成

得到基网格后,用流线生成法建立模型的多分辨率表示.常见的多分辨率表示方法是把原模型参数化到平面,在参数域上进行规则划分,最后映射回原始模型 M 获取新顶点位置.我们给出一种新的不依赖参数化的方法.把基网格 M_0 (图 8 中 ΔACB 和 ΔABD 是基网格上的两个三角形)上每个三角形一分为四,每条边需要加入一个新顶点.新顶点的采样方法是,选取过相应边的一个有向平面,在有向平面的约束下用类似流线生成的方法求得原始网格 M 上连接此边两端点的折线,折线中点即为所要的新顶点.如图 8 中的 ΔABC 三边分别得到向上采样中点 P_{AB}, P_{AC}, P_{BC} ,从而将 ΔABC 分成 4 个三角形,分别为 $\Delta AP_{AB}P_{AC}, \Delta BP_{AB}P_{BC}, \Delta CP_{BC}P_{AC}, \Delta P_{AB}P_{AC}P_{BC}$.

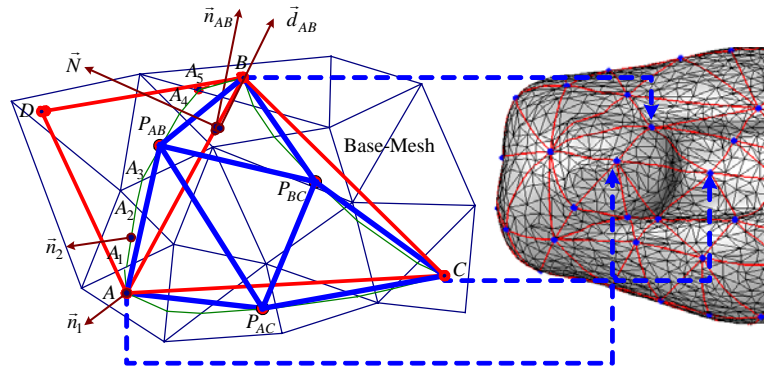


Fig.8 Illustration of the up-resampling approach

图 8 重采样方法图示

以基网格 M_0 的 AB 边为例,细分时要在该边上加入新顶点 P_{AB} ,其生成步骤如下:

- 1) 选 A 为起点, B 为终点,则可得方向 \vec{d}_{AB} ;
- 2) 计算边 AB 相邻三角形 ΔABC 和 ΔABD 的法向,取两者的法向平均作为边 AB 的法向 \vec{n}_{AB} ;
- 3) 计算 $\vec{N} = \vec{n}_{AB} \times \vec{d}_{AB}$ 作为有向平面的法向,且此有向平面过直线 AB ;
- 4) 获取 A 点所在三角形 Δ_1 ,得到该三角形的法向 \vec{n}_1 ;
- 5) 从 A 点出发,在三角形 Δ_1 上沿方向 $\vec{r}_1 = \vec{N} \times \vec{n}_1$ 得到交点 A_1 ,并得到 Δ_1 的相邻 Δ_2 及其法向 \vec{n}_2 ;
- 6) 再从 A_1 出发,在三角形 Δ_2 上沿方向 $\vec{r}_2 = \vec{N} \times \vec{n}_2$ 得到交点 A_2 ,并得到 Δ_2 的相邻 Δ_3 及其法向 \vec{n}_3 ;
- 7) 用类似于 6) 的方法依次生成 $A_3A_4A_5$;
- 8) 这样就可以得到一组流线段 $AA_1A_2A_3A_4A_5B$ (图 8 中以浅细线表示),取此流线段中点位置 P_{AB} 作为边 AB 中点向上重采样点.
- 9) 对边 AC 和边 BC 采用步骤 1)~8),即可得到相应的中点向上采样点 P_{AC} 和 P_{BC} .

将基网格 M_0 上所有边用相同方法得到中点向上采样点之后,将每个三角形都一分为四得到第 1 次细分后的多分辨率模型 M_1 ,再将 M_1 看成基网格,用同样的方法得到第 2 次细分后的模型 M_2 .依此类推,即可得到一系列不同分辨率表示的模型 $M_1M_2\dots M_n$.流线生成法建立多分辨率表示的特点是引入了有向平面法向 \vec{N} ,而不是简单将 \vec{d}_{AB} 投影到网格表面,在复杂区域能够使流线从起点正确、快速地到达终点.

4 实验结果

算法的实现基于 CGAL 图形库,计算机为 Intel 双核 1.86GHz,内存 1G,WindowsXP 操作系统.模型质量从两方面来评价:顶点价和三角形内角的频率分布.选取的比较对象为原始模型、本文算法得到的基网格模型及 ECK 算法得到的基网格模型.

这里一共给出了 4 个模型(分别见图 9~图 13)多分辨率表示生成的实验数据,见表 1.从表 1 可以看出, h 越

小, uv 流线条数就越多, uv 流线的交点个数也越多,生成的基网格顶点数也就越多,从而所花费时间越长.原始网格的规模也是影响时间效率的重要因素.

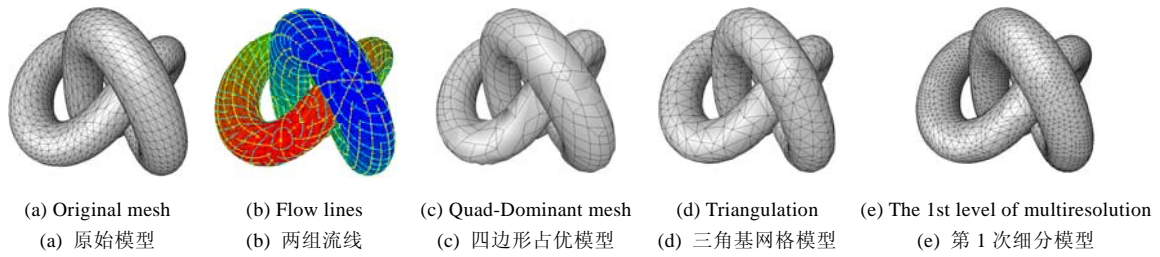


Fig.9 Remeshing and multiresolution of texturedknot model ($h=0.5, a=0$)

图 9 Texturedknot 模型的重新网格化及多分辨率表示($h=0.5, a=0$)

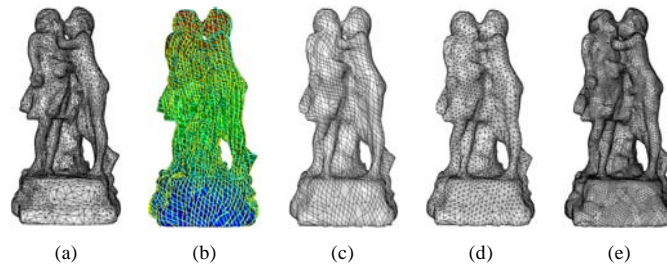


Fig.10 Remeshing and multiresolution of sculpture model ($h=0.24, a=0$)

图 10 Sculpture 模型的重新网格化及多分辨率表示($h=0.24, a=0$)

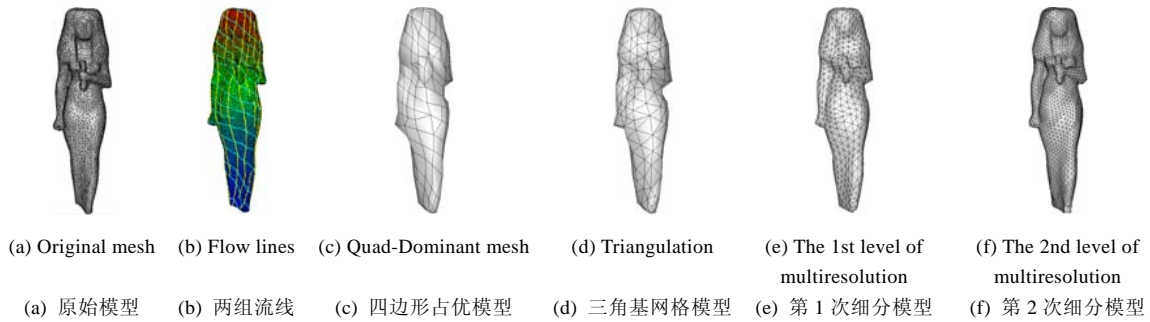


Fig.11 Remeshing and multiresolution of Isis model by using our method ($h=1, a=0$)

图 11 本文算法对 Isis 模型的重新网格化及多分辨率表示($h=1, a=0$)

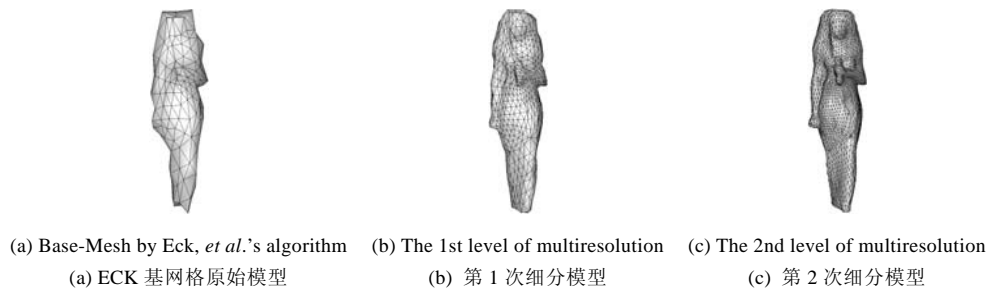


Fig.12 Remeshing and multiresolution of Isis model by using ECK

图 12 ECK 算法对 Isis 模型的重新网格化及多分辨率表示

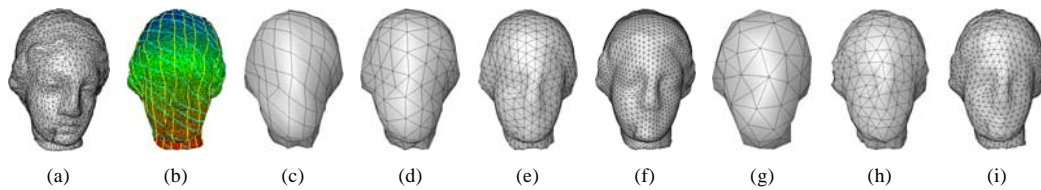


Fig.13 Remeshing and multiresolution of venus-head model by our method ((a)~(f)) and

ECK ((g)~(i)) ($h=1, a=0$)

图 13 本文算法((a)~(f))和 ECK 算法((g)~(i))对 venus-head 模型的重新网格化及多分辨率表示($h=1, a=0$)

Table 1 Efficiency of our algorithm

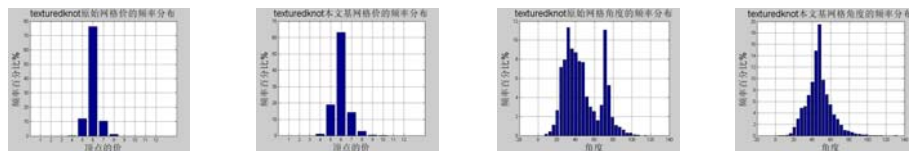
表 1 本文算法的运行效率

模型名称	顶点数	面数	亏格数	参数 h/a	建立场的 时间(s)	生成流线 时间(s)	生成基网 格时间(s)	基网格 顶点数	生成第 1 层多 分辨率时间(s)
Texturedknot (Fig.9)	2 871	5 742	1	0.5/0	0.219	3.203	0.25	1 241	1.968
Sculpture (Fig.10)	7 550	15 108	3	0.24/0	1.64	11.718	0.4	3 617	7.012
Isis (Fig.11 and Fig.12)	5 865	11 726	0	1/0	1.25	1.469	0.031	223	2.063
Venus-Head (Fig.13)	4 200	8 396	0	1/0	0.625	1.219	0.016	248	1.28

我们还对上述 4 个模型的重新网格化质量作了分析.因为原始模型图 9(a)是用细分方法得到的,从图 14 可以看出,本文算法对价没有改进反而下降,6 度价的的比例从图 14(a)的 76.3%降为(b)的 63%,但对角度的改进比较明显,从散乱分布的(c)改善为占 19.46%的 50°角(d).若原始模型质量不好,则本文算法在价和角度的改进上就比较明显,如图 15 的 sculpture 模型(如图 10 所示),6 度价的的比例从(a)的 36.45%上升到(b)的 67.32%,内角频率分布也将原始模型最大比例为 6.84%的 36°角提高到 15.2%的 48°角.

图 16 和图 17 中是本文与 ECK 算法的比较结果,图 16 中 Isis 模型(如图 11 和图 12 所示)6 价顶点比例从(a)的 38.35%上升到(b)中的 52%,角度占最大比例为 7.06%的 36°角(d)改善为 12.14%的 44°角(e).图 17 中 venus-head 模型(如图 13 所示)6 价顶点比例从原始模型(a)的 38.05%上升到基网格(b)的 60%,角度最大比例为 7.9%的 40°角(d)改善为 13.4%的 48°角(e).ECK 算法在图 16(c)中 6 价顶点比例上升到 82.79%,而在图 17(c)中仅为 39.13%,几乎没有什么改善.从图 16 和图 17 的(e)(f)可以看出,对角度的改进效果,ECK 算法略逊于本文算法.

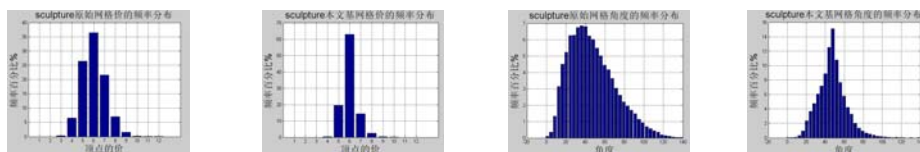
从实验结果和与 ECK 算法的比较可知,本文算法对模型质量具有比较稳定的改善效果.



(a) Original mesh valence (b) Valence of Fig.9(d) (c) Original mesh angle (d) Angle of Fig.9(d)
(a) 原始模型价分布 (b) 图 9(d)价分布 (c) 原始模型角度分布 (d) 图 9(d)角度分布

Fig.14 Valence and angle distribution of Texturedknot model

图 14 Texturedknot 模型顶点价和角度的频率分布



(a) Original mesh valence (b) Valence of Fig.10(d) (c) Original mesh angle (d) Angle of Fig.10(d)
(a) 原始模型价分布 (b) 图 10(d)价分布 (c) 原始模型角度分布 (d) 图 10(d)角度分布

Fig.15 Valence and angle distribution of the Sculpture model

图 15 Sculpture 模型顶点价和角度的频率分布

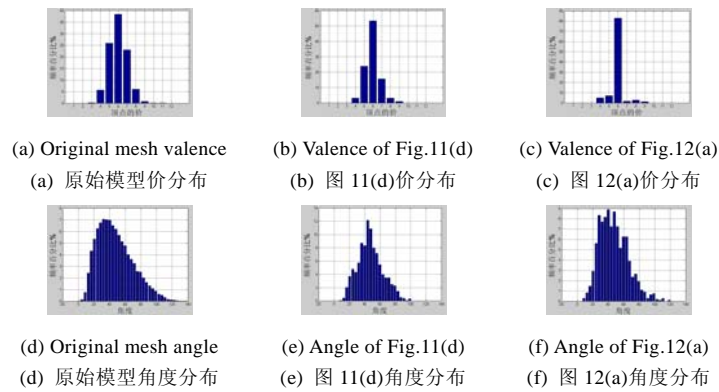


Fig.16 Valence and angle distribution of Isis model

图 16 Isis 模型顶点价和角度的频率分布

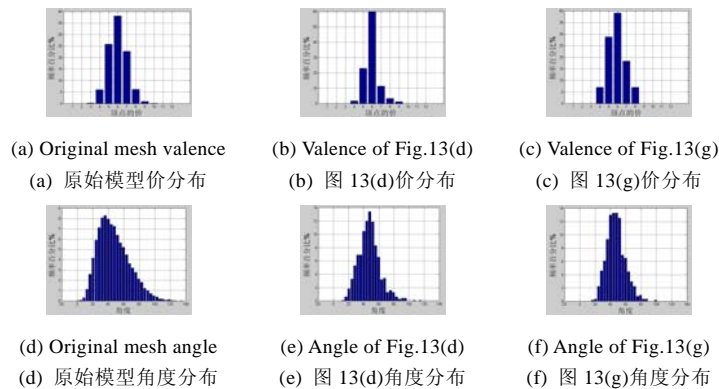


Fig.17 Valence and angle distribution of venus-head model

图 17 Venus-Head 模型顶点价和角度的频率分布

5 结 论

本文通过在模型表面建立拉普拉斯场,生成两组夹角为 60° 的流线,获取两组流线的交点及连接关系,最后生成新的三角基网格并构在此基础上建立模型的多分辨率表示.在实现重新网格化和多分辨率表示过程中无需参数化过程,取得的效果与 ECK 算法接近,为重新网格化和多分辨率表示提供了一种新思路.算法中若夹角改为 90° 就可以实现四边形网格化.

算法还存在需要改进之处:目前没有考虑模型的特征特别是尖锐特征的保持;两组 60° 夹角流线与两组正交流线有很大差异,不能简单地归结为夹角不同,在极值点附近多边形网格重构非常复杂,导致该区域网格元素质量不是很好;第 3 节中的多分辨率表示方法在实验中很少出现翻转情况,但这一点并不能从理论上得到保证,需要进一步从理论上探讨不翻转的算法以适应复杂模型;基网格模型未作任何优化.

References:

- [1] Turk G. Retiling polygonal surfaces. *Computer Graphics*, 1992,26(2):55-64.
- [2] Bischoff S, Kobbelt L, Rössl C. Geometric modeling based on triangle meshes. In: *SIGGRAPH 2006 course#29*.
- [3] Peng L, Li GQ, Xiong YH, Dai Z. A new multiresolution representation approach based on curvature. *Journal of Computer-Aided Design & Computer Graphics*, 2008,20(6):700-706 (in Chinese with English abstract).
- [4] Zhang LY, Nie JH, Zhou LS, Zhou RR. Research on adaptive remeshing of triangle meshes. *Journal of Computer-Aided Design &*

- Computer Graphics, 2002,14(3):204–208 (in Chinese with English abstract).
- [5] Fu Y, Zu XM, Zhou BF. 3D model remeshing using circular parameter domain and importance sampling. Chinese Journal of Computers, 2007,30(12):2124–2131 (in Chinese with English abstract).
- [6] Alliez P, Meyer M, Desbrun M. Interactive geometry remeshing. ACM TOG, 2002,21(3):347–354.
- [7] Alliez P, de Verdiere EC, Devillers O, Isenburg M. Isotropic surface remeshing. In: Proc. of the Shape Modeling Int'l. Washington: IEEE Computer Society, 2003. 49–58.
- [8] Gu XF, Gortler SJ, Hoppe H. Geometry images. ACM TOG, 2002,21(3):355–361.
- [9] Sander PV, Wood ZJ, Gortler SJ, Snyder J, Hoppe H. Multi-Chart geometry images. In: Proc. of the Eurographics/ACM SIGGRAPH Symp. on Geometry Processing. Aachen: Eurographics Association, 2003. 146–155.
- [10] Marinov M, Kobbelt L. A robust two-step procedure for quad-dominant remeshing. Computer Graphics Forum, 2006,25(3): 537–546.
- [11] Marinov M, Kobbelt L. Direct anisotropic quad-dominant remeshing. In: Proc. of the 12th Pacific Conf. on Computer Graphics and Applications. 2004. 207–216.
- [12] Alliez P, Cohen-Steiner D, Devillers O, Levy B, Desbrun M. Anisotropic polygonal remeshing. ACM TOG, 2003,22(3):485–493.
- [13] Dong S, Kircher S, Garland M. Harmonic functions for quadrilateral remeshing of arbitrary manifolds. Computer Aided Geometric Design, 2005,22(5):392–423.
- [14] Dong S, Bremer PT, Garland M, Pascucci V, Hart JC. Spectral surface quadrangulation. ACM TOG, 2006,25(3):1057–1066.
- [15] Tong YY, Alliez P, Cohen-Steiner D, Desbrun M. Designing quadrangulations with discrete harmonic forms. In: Proc. of the SGP 2006. Cagliari: Eurographics Association, 2006. 201–210.
- [16] Ray N, Li WC, Levy B, Sheffer A, Alliez P. Periodic global parameterization. ACM TOG, 2006,25(4):1460–1485.
- [17] Zhang E, Hays J, Turk G. Interactive tensor field design and visualization on surfaces. IEEE Trans. on Visualization and Computer Graphics, 2007,13(1):94–107.
- [18] Garland M. Multiresolution modeling: Survey & future opportunities. In: Proc. of the Eurographics 1999—State of the Art Reports. Milan, 1999. 111–131.
- [19] Eck M, DeRose T, Duchamp T, Hoppe H, Lounsbery M, Stuetzle W. Multiresolution analysis of arbitrary meshes. In: Mair SG, Cook R, eds. Proc. of the SIGGRAPH 1995. New York: ACM, 1995. 173–182.
- [20] Hoppe H. Progressive meshes. In: Proc. of the SIGGRAPH 1996. New York: ACM, 1996. 99–108.

附中文参考文献:

- [3] 彭莉,李桂清,熊赟晖,戴专.按曲率选取基点的多分辨率表示重构算法.计算机辅助设计与图形学学报,2008,20(6):700–706.
- [4] 张丽艳,聂军洪,周来水,周儒荣.自适应三角网格模型重新布点算法的研究.计算机辅助设计与图形学学报,2002,14(3):204–208.
- [5] 付妍,朱晓明,周秉锋.基于圆形参数域和重要性采样的三维模型网格重建.计算机学报,2007,30(12):2124–2131.

附录 1. 本文 5 个主要算法的详细描述

先统一 3 个概念.流线段端点:单条流线段的起点和终点,根据落在三角形上的位置分成顶点型,边点型和面点型.流线段:由起点和终点构成的一条有向线段.有 u 线段和 v 线段两种类型.流线:由若干条流线段构成的有向折线.有 u 线和 v 线有两种类型.

算法 1. 单条流线段生成.

输入:三角网格模型 M ,梯度场 T ,流线段起点 SP .

输出:流线段终点 EP .

1. Switch(流线段起点类型)

2. Case 顶点型: //起点正好是原始模型顶点位置

遍历该顶点所有三角形,在每个三角形上都按梯度场方向得到下一终点位置,计算这些终点的函数值,取最大的作为流线段的终点 EP .具体做法见图 18(a),假设起点 SP 为三角形的 P 点,连接起点 SP 与三角形另外两个顶点 Q, R 构成两条相交直线,将整个平面区域分成 4 部分区域,判断三角形梯度场方向 F 落在哪个区域,最后终

点位置则分别对应(a1)~(a4)这4种情况.

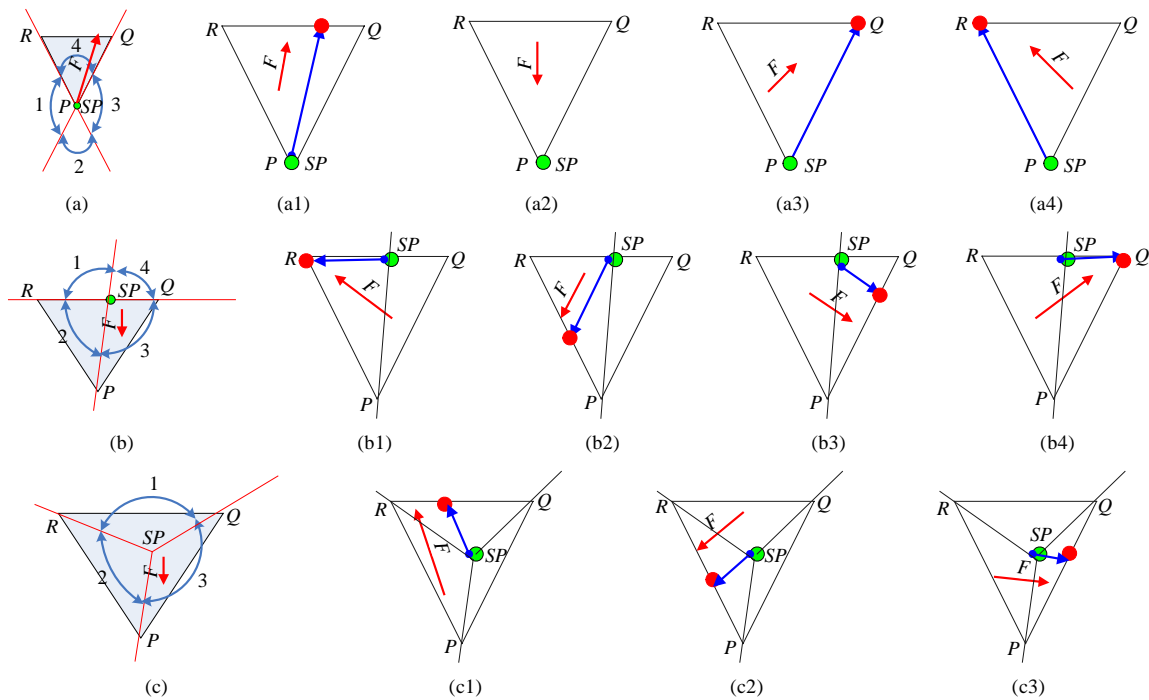


Fig.18 Generation of a stream line. (a) Vertex start point; (b) Edge start point; (c) Face start point

图 18 单条流线段生成.(a) 起点为顶点型;(b) 起点为边点型;(c) 起点为面点型

3. Case 边点型: //起点正好落在原始模型半边上

获取起点所在半边结构对应的三角形,根据该三角形上的梯度场,计算流线段终点.具体做法如图 18(b)所示,假设起点 SP 落在边 RQ 上,连接 SP 与另外一个顶点 P ,与边 RQ 所在直线构成相交直线,将整个三角形所在平面分成 4 个区域,判断三角形梯度场方向 F 落在哪个区域,最后终点位置则分别对应(b1)~(b4)这 4 种情况.

4. Case 面点型: //起点正好落在原始模型三角面内

获取起点所在三角形,根据该三角形上的梯度场,分 3 种情况计算流线段终点.具体做法如图 18(c)所示,连接 SP 与 3 个顶点 P, Q, R 构成 3 条射线,将整个三角形所在平面分成 3 个区域,判断三角形梯度场方向 F 落在哪个区域,最后终点位置则分别对应(c1)~(c3)这 3 种情况.

算法 2. u 型流线生成.

输入:三角网格模型 M ,梯度场 T .

输出:一条 u 线(由若干折线段组成),三角形上单条 u 线段表.

1. 选取 u 线种子点 s ;
2. 从 s 出发,按算法 1 增长,直到极大值点或与周围 u 线的距离小于碰撞检测距离为止,得到一条 u 线;
3. 每求出一段 u 线段,把它加入所在三角形中 u 线链表中;
4. 在此 u 线上从起点开始根据碰撞检测距离的 h_2 放置 v 线种子点,根据 h_1 在 u 线两边放置 u 线、 v 线种子点,计算它们的拉普拉斯函数值,将这些种子点加入到 u 线、 v 线种子点集合中并重新按函数值排序;
5. 若种子点 s 同时又是极小值点,则需从此极值点处生成 $360^\circ/60^\circ=6$ 条方向对称的 u 线.
6. 从 u 线种子点集合中选取函数值最小的点,重复上述过程,直到 u 线种子点集合为空为止.

算法 3. v 型流线生成.

输入:三角网格模型 M ,梯度场 T .

输出:一条 v 线,三角形上单条 v 线段表.

1. 在所有极值点(极大值和极小值)处附近距离为 h_2 的周围生成一个 90° 的封闭 v 线;
2. 从算法 2 中生成的 v 线种子点集合中选择函数值最小的 s ;
3. 从 s 出发,将梯度场逆时针旋转 60° 作为 v 线前进的方向场,按算法 1 增长 v 线,直到极大值点或与周围 v 线的距离小于碰撞检测距离为止;
4. 每求出一段 v 线段,把它加入所在三角形中 v 线链表中;
5. 从 v 线种子点集合中选取函数值最小的点,重复上述过程,直到 v 线种子点集合为空为止.

算法 4. 两组流线求交.

输入:每个三角形的 u 线段表和 v 线段表.

输出:各单条 u 线、 v 线的相交点集合.

1. 对原始网格的每个三角形 t
2. 获取 t 上所有 u 线段及其相邻三角形中的 u 线段,构成 u 线段集合 $ulist$,
3. 获取 t 上所有 v 线段及其相邻三角形中的 v 线段,构成 v 线段集合 $vlist$,
4. 依次取出 $ulist$ 流线段 $uline$
5. 依次取出 $vlist$ 流线段 $vline$;
6. 判断 $uline$ 与 $vline$ 是否相交,若相交求交点,并将此交点按所在流线段的顺序放置到 $uline, vline$ 所在单条 u 线、单条 v 线的相交点集合中.

算法 5. 新网格生成.

输入:各单条 u 线、 v 线上的交点集合.

输出:CGAL 的新网格.

1. 每个交点都落在一条 u 线和 v 线上.将 u 线上的点的连接关系定义成南北连接, v 线上点的连接关系定义成东西连接,也就是说,一个点最多有东西南北 4 个相邻点.
2. 每个交点按东南西北 4 个方向将平面分成 4 个区域,按逆时针方向分别为东北、西北、西南、东南.
3. 遍历所有单条 u 线,根据与 v 线交点出现的顺序提取交点间的南北拓扑关系;
4. 遍历所有单条 v 线,根据与 u 线交点出现的顺序提取交点间的东西拓扑关系;
5. 从任意一个交点 p 出发,按东北、西北、西南、东南的顺序找出没有生成多边形区域;
6. 从此区域出发,按逆时针方向前进,直至回到交点本身,从而构成一封闭集合,这就是新网格上的一个多边形面.
7. 用类似的方法遍历 p 点的 4 个区域,直至所有交点的 4 个区域都遍历完成,即可构成新多边形网格.



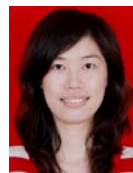
熊贇晖(1972—),男,江西樟树人,博士生,讲师,主要研究领域为 CAGD,DGP.



韩国强(1962—),男,博士,教授,博士生导师,主要研究领域为多媒体数据压缩,图像重建.



李桂清(1966—),男,博士,副教授,主要研究领域为 CAGD,DGP,RE,复杂物体的几何造型.



彭莉(1984—),女,硕士生,主要研究领域为 DGP,三维动画.