

局部求精算法中的纹理动态拼装^{*}

梁 策, 潘万彬

(杭州电子科技大学 计算机科学与技术系, 浙江 杭州 310018)

摘要:模型的简化有时会导致用户所关注的模型细节的丢失。为了保持这些重要的细节,使用了局部求精的方法,在模型简化后,有选择性地恢复用户所关注区域的细节,并实现了在局部求精时对网格模型的动态修改。若模型附有纹理信息,则对简化模型重新映射其纹理,以实现纹理的优化。首先对模型进行分割从而形成一系列网格片,将它们参数化到纹理空间后使用了正方形贪心排列算法进行重排,得到排列更加紧凑、高效的纹理图。研究表明,该算法实现了对简化模型纹理的重新映射。

关键词:三维几何模型;局部求精;图块拼装;区域分割

中图分类号:TP391.41

文献标识码:A

文章编号:1001-4551(2010)09-0120-03

Dynamic texture packing during the region refinement

LIANG Ce, PAN Wan-bing

(Department of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou 310018, China)

Abstract: During the mesh simplification, some details the users concerned may be lost. In order to maintain these important details, local refinement was used to refine the region selectively which the users were interested. In this way, special details can be reserved. If there was texture on the models, the texture should be re-mapped in order to get an optimal texture for the simplified model. For realizing the texture re-mapping, the model's surfaces were partitioned into many disks-like charts parameterized into squares. And the squares were packed in to the texture space by a greedy algorithm. The results indicate that it can improve and optimize the utilization of the texture and modify charts dynamically generated by models' partition during region refinement process.

Key words: 3D geometric model; selective refinement; chart packing; regions partition

0 引 言

处理带属性的模型简化算法,其主要思路大致分为两种方向:第 1 种方向,在进行几何的简化同时处理模型属性。如 Hoppe^[1] 和 garland^[2] 的简化算法,这两种算法均采用边折叠方式,将属性作为能量函数中的一部分来考虑。第 2 种方向则是把几何简化和属性简化分开考虑^[3]。几何简化完毕后,建立原模型和简化后模型的映射关系,根据该映射关系给简化后的模型附上材质和属性^[4]。以纹理为例,这种方法处理后的简化模型,其纹理图集则需要重新生成。

本研究使用区域求精的方式,根据用户的需求有选择性地恢复简化模型局部的细节;对于属性的处理则采用了法向映射的方法,并采用了正方形贪心排序算法实现的简化模型完成纹理重映射。

1 相关工作

1.1 局部求精

局部求精较模型简化而言,是一个逆向的分裂过程。在简化时记录下一些信息,再根据这些简化信息,对简化后的模型进行局部的细化求精。如 Hoppe 提出的经典的 Progressive Meshes, PM 由一个初始模型 M_n 和一组记录细节信息组成 $\{vsplit_1, vsplit_2, \dots, vsplit_{n-1}\}$, 这些分裂信息用一个有序的链表存储,保存了边折叠过程中删除的边 $\{v_s, v_t\}$, 与这两个顶点均相邻的顶点 $\{v_s, v_r\}$, 以及分裂点最后产生的新顶点索引号,根据这些信息可以保证正确地恢复出模型的原始拓扑结构。

1.2 网格模型的分割和 Packing

对模型实现纹理映射,需要将网格模型进行参数

化,而整体参数化会产生比较大的扭曲效果,因此引进了网格分割技术,尽量地把模型表面剖分成一个个比较平坦的网格块,再对这些网格块分别进行参数化,减少扭曲等误差。最后在纹理空间中将这些块重新拼装(packing),以重新生成纹理坐标。下面介绍国内外学者的相关研究进展。

Sander 在 2001 年使用表面的贪心聚类法^[5],把每个三角面片作为一个单元,再各自和其相邻三角面片合并。以最小合并代价为标准,循环地执行合并操作,直到合并代价值超过用户给定阈值,停止合并操作,形成了最终的区域集合。完成分割后,将各区域集合参数化到多边形中,对这些多边形求最小包围盒,形成一系列矩形,将它们按照最大边长排序,在正方形纹理空间中逐行排列后再保证每个矩形到达最左、最下的位置^[6]。

Lévy 在 2002 年提出的算法^[7],首先根据各边的二面角估算形状特征,在离形状特征最远的地方设置种子面,然后增长合并这些种子面,使它们在高曲率区域的特征曲线上相交,由此使每个分割块尽可能多地包含网格模型中的顶点。然后使用不规定边界的保角映射参数化各分块。拼装的方式则采用类似俄罗斯方块的方式,将所有不规则的块根据其最大直径从大到小排序,并将各块竖直放置。用一条水平弹性曲线包含现有格局最上方的轮廓,在新的块放入的时候,寻找该块的轮廓和那条弹性曲线构成的间隙面积最小的位置,放置新块,并更新现有格局。循环进行此操作,完成所有块的拼装工作。该算法实现了不规则块的拼装,在纹理空间的利用率上比先前的算法都有比较大的提高。但较之以前的算法,该算法运算量较大,而且当网格模型比较复杂或者数据量比较大时,该算法的效率显得不够理想。

Sander, Hoppe 于 2003 年对 Sander 在 2001 提出的算法进行了改进^[8]。用基于最优聚类的方式剖分模型,主要解决了由于剖分后各块的边界问题引起的“表面重建不连续,有破洞”的情况。由于分割上的优化,使最后的拼装效率也得到了一定的提高。

2 算法基本思想的描述

2.1 区域求精中动态更新模型数据

本研究的主要工作在于“对模型区域求精过程中,对改变的模型的动态的纹理图块的拼装”。

PM 的模式为 $\{M^0, vsplit_0, vsplit_1, \dots, vsplit_{n-1}\}$ 。其产生的简化模型为 M^0 , 原始模型为 M^n , 对原始模型进行剖分后形成一系列的网格片, 这个过程即是将网格模中的三角面片归类到不同的簇中, 得到一个簇的集合 $Clusters: \{c_0, c_1, c_2, \dots, c_N\}$, 其中 C_n 为 $\{T_a, T_b,$

$T_c, \dots, T_m\}$, 它表示该块中所包含的模型三角面索引号的集合。假设对原始模型分割形成的簇集合为 $ClustersM^n$ 。

接着对简化模型 M^0 进行分割, 假设得到的簇集合为 $ClustersM^0$ 。则实施区域求精时, 根据所选取的待求精三角面片来判断集合 $ClustersM^0$ 中将有那些簇需要改变, 根据这些三角形在 $ClustersM^n$ 中的位置来对 $ClustersM^0$ 进行更新。假设某次求精中需要改变的三角面为 $\{T_a, T_b, T_c, \dots, T_n\}$, 可以在 $ClustersM^n$ 中找到这些三角形所处的簇(假设为 C_nM^n), 然后再修改在简化模型中分割得到的 $ClustersM^0$ 信息。如此遍历所有求精过程中涉及的三角面片, 用该方式完成对简化模型网格片的更新。

2.2 对模型 packing 的实现

在上一节中完成了对模型表面的分割, 得到顶点区域块的集合 $Clusters$ 。用调和映射的方式将每个网格片参数化, 采用了边界固定的参数化方式, 其边界为正方形。这样重新生成纹理图集的问题就转化成了对这些正方形集合的拼装。对矩形的拼装问题已有许多比较好的解决方式, 包括模拟退火算法、遗传算法、启发式算法等。笔者在矩形拼装问题贪心算法^[9]的基础上, 略加改变。拼装过程也即是正方形顺序地放入现有的格局中。新的正方形放入现有的格局时, 遵循以下策略:

(1) 每当一个正方形进入现有格局时, 总是选择占角穴度最大的位置。穴度 D_i 定义如下式:

$$D_i = 1 - \frac{d_{\min}}{\sqrt{S_i}} \quad (1)$$

式中 $d_{\min}$ —两正方形块间最小的曼哈顿距离; S_i —该正方形的面积。

(2) 当格局中有多个位置的穴度值都相同时, 选择贴边最多的位置。

(3) 若穴度值和贴边数都相同, 则以最左最下原则选取位置。

假设第 i 个正方形最左最下的顶点为 $P_i(x_i, y_i)$, 求出了 P_i 的值即为该正方形的摆放位。因此拼装问题其实就是需要求得所有正方形的 P_i 值, 最终解集 $Pos\{p_0, p_1, p_2, p_3, \dots, p_n\}$ 就是拼装的结果。由以上规则可知, 新放入的正方形所选取的位置, 都落在原有格局中各正方形的各个顶点上。设第 i 个正方形的顶点集为 $\{P_{i,1}, P_{i,2}, P_{i,3}, P_{i,4}\}$ 。则当新的正方形进入时, 候选的放置位置可从由顶点组成的解集 $Postion$ 中获得, $Postion$ 为 $\{\{P_{0,1}, P_{0,2}, P_{0,3}, P_{0,4}\}, \{P_{1,1}, P_{1,2}, P_{1,3}, P_{1,4}\}, \dots, \{P_{i-2,1}, P_{i-2,2}, P_{i-2,3}, P_{i-2,4}\}\}$ 。因此可以根据上述的排列策略在解集 $Postion$ 中选取最优

的解,从而实现优化的排列。

3 实 验

本实验中利用 Garland 的简化算法对模型进行简化,然后将简化过程中的相关信息存入自定义的多分辨率模型表示格式文件中,利用这些信息实现简化模型的求精,并完成了在区域求精中动态地修改网格片信息的工作。而对于拼装的工作,采用了一个较为简单的常用拼装算法作为对比,该参照算法的具体思想如下:根据正方形的边长不同进行排序,然后进行逐行排列。排列时,判断纹理空间每行所留的空间是否能够容纳未经排列的正方形网格片。如果可以,则找一个最适合该空间的网格片排列到此处,直到该行排列的宽度达到纹理空间的宽度。由于一行里面的正方形网格片边长不一,还可以继续在边长较小的网格片上排列大小合适的网格片。在新的行里继续排列剩下未经排列的网格片,直到所有网格片都排入纹理空间中。

根据对三维模型的分割,以及对分割得到的网格片的参数化,在纹理空间中得到了一系列大小不等的正方形块。然后分别使用上述参照算法和本研究中提出的排列算法对这些正方形块进行排列以得到新的纹理图集。其效果如图 1、图 2 所示。

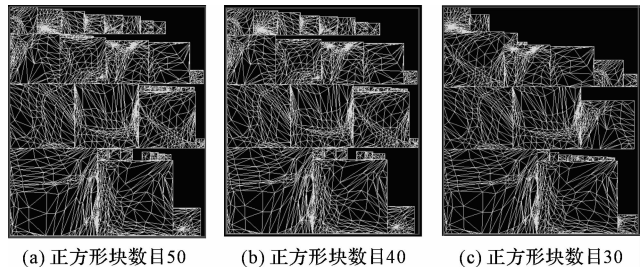


图 1 参照算法排序结果

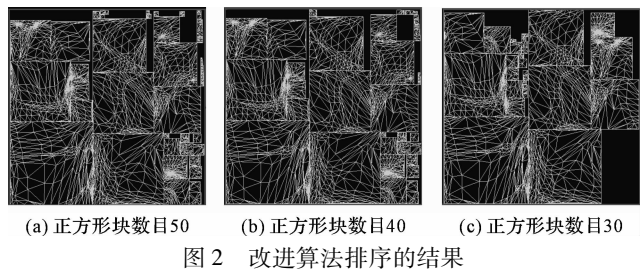


图 2 改进算法排序的结果

使用参照算法排列得到的结果如图 1 所示,根据排列中正方形块数目的不同,分为 3 个效果图。其中图 1(a)显示了 50 个大小不等的正方形块排列的结果,每个正方形都是通过对应的网格片参数化得到的。使用本研究中算法排列得到的效果图如图 2 所示,对比图 1 和图 2 的排列结果,可以看到采用改进后的排列算法得到的排列结果更加紧凑,未利用空间(即黑色部分)面积也相对要少。

表 1 两个算法排列的效率对比

packing 方式	排列的块的总个数			
	30	40	50	
参照算法	纹理大小	1.331×1.047	1.301×1.112	1.301×1.114
	排列效率	78.6%	85.2%	85.7%
	耗时/ms	1.72	2.25	2.12
改进算法	纹理大小	1.151×1.151	1.171×1.171	1.171×1.171
	排列效率	83%	90.2%	90.6%
	耗时/ms	1.67	2.31	2.15

4 结束语

研究表明,从计算速度上看,当所需排列的正方形数目较小时改进算法的耗时更少,当正方形数目较多时,会因为排列策略的复杂性而导致排列速度相对较慢,但降低得不是很明显。然而改进后的拼装方法能使排列效率和纹理空间大小的节省上有较明显的提高(其中纹理大小是由最终纹理图的大小和网格片总边长的比例所决定的)。

在“快速规则驱动的小区三维建筑物建模技术的研究及应用”的项目中,应用到了数据量比较大的三维网格模型,主要是建筑模型(比如住宅楼房等)。由于这些建筑模型包含有较多数量的网格面片,导致模型处理速度缓慢,需要对模型进行简化,本研究的主要工作在于处理这些简化后的建筑模型的纹理属性的优化。

参考文献(References):

[1] HOPPE H, GU Xian-feng, STEVEN J, et al. Geometry Images[C]//ACM SIGGRAPH 2002 Conference Proceedings. San Antonio: [s. n.], 2002:355 – 361.

[2] GARLAND M, HECKBERT P S. Simplifying Surfaces with Color and Texture using Quadric Error Metrics[C]//Proceedings of IEEE Visualization 98. Chapel Hill: [s. n.], 1998:263 – 269.

[3] 张华琪,王毅刚. 基于法向映射的复杂模型的实时绘制技术[J]. 系统仿真学报, 2003, 15(3):342 – 346.

[4] CIGNONI P, MONTANI C, SCOPIGNO R, et al. A General Method for Recovering Attribute Values on Simplified Meshes[C]//IEEE Visualization. Chapel Hill: [s. n.], 1998: 59 – 66.

[5] COSGOVE I. Adaptive Unwrapping for Interactive Texture Painting [C]//Symposium on Interactive 3D Graphics. Chapel Hill: [s. n.], 2001:209 – 216.

[6] SANDER P V, SNYDER J, GORTLER S J, et al. Texture Mapping Progressive Meshes[C]//SIGGRAPH. Los Angeles: [s. n.], 2001:409 – 416.

[7] LÉVY B, PETITJEAN S, RAY N, et al. Least Squares Conformal Maps for Automatic Texture Atlas Generation[C]//SIGGRAPH. San Antonio: [s. n.], 2002:362 – 371.

[8] SANDER P V, WOOD Z J, GORTLER S J, et al. Multi-Chart Geometry Images [C]//Eurographics Symposium on Geometry Processing. Aachen: [s. n.], 2003:146 – 155.

[9] 陈端兵,黄文奇. 求解矩形 packing 问题的贪心算法[J]. 计算机工程, 2007, 33(4):160 – 162. [编辑:李 辉]