

基于 GPU 的复杂三角网格模型多分辨率绘制

王化雨 刘惠义 冯艳蓉 李奎生

(河海大学计算机与信息学院 南京 211100)

摘要: 针对高度细节三角网格模型实时绘制问题,通常使用 LOD 技术对模型进行处理,本文提出了一种适用于 GPU 的针对零亏格的三角网格模型的多分辨率绘制方法。该方法先将三角网格进行球面参数化,然后利用参数化结果构建几何图像,并在其上建立四叉树模型,最后使用 GPU 着色器进行模型细节层次的选择并对网格进行简化,实现模型的多分辨率绘制。与已有方法相比,利用 GPU 高度并行运算的特点,减少了简化模型过程的计算时间。实验结果表明,该方法在保证绘制质量的同时达到实时绘制的要求。

关键词: 三角网格;LOD;着色器;几何图像;四叉树

中图分类号: TP3.05 **文献标识码:** A **国家标准学科分类代码:** 520.6030

Multiresolution rendering of complex triangular mesh model based on GPU

Wang Huayu Liu Huiyi Feng Yanrong Li Kuisheng

(Hohai University, College of Computer and Information, Nanjing 211100, China)

Abstract: In this paper, an approach is proposed for multiresolution rendering of triangular mesh model with zero genus which based on GPU. The triangular mesh parameterized onto a spherical domain, then the geometry images is constructed with the parametric results. The quad-tree is constructed on the geometry image, finally the LOD selection and mesh simplification is performed in the shader. The time of the simplified model process is reduced with use of GPU highly parallel computing features compared with the existing methods. The experimental results show that this approach ensure the quality of rendering and meet the requirements of real-time rendering.

Keywords: triangular mesh; LOD; shader; geometry image; quad-tree

1 引言

近年来,三维几何模型在虚拟现实、数字娱乐媒体以及计算机辅助设计等领域有非常广泛的应用。随着 3D 扫描仪的发展与普及,获取的三维模型的精度越来越高,细节越来越丰富,模型文件所包含的数据量也越来越大,高度细节复杂模型的实时绘制成为一个具有挑战性的问题。

为使高度细节网格模型达到实时绘制效果,需对模型进行简化,常用的模型简化方法是层次细节技术^[1](level of detail, LOD)近年来随着 GPU 的发展,出现了许多基于 GPU 的 LOD 绘制方法。Boubekeur 等人^[2]利用一个顶点着色器做通用的网格细分,但是这样的模式限制了一部分预先计算好的模式下的细化,在任意拓扑结构的三角网格中表现不佳。DeCore 和 Tatarchuk^[3]提出一种基于 GPU 的使用八叉树顶点聚类简化特征网格的方法,这种方法不需要进行预计算和存储顶点层次,但是绘制精度不高。Liang 等人^[4]提出一种完全运行在 GPU 上的针对顶点级

网格细分的视点相关 LOD 算法。

Gu 等人^[5]在 2002 年 SIGGRAPH 首先提出了几何图像的方法,首先根据网格连接找到任意一个初始切割,然后将网格参数化到一个平面上,根据参数化的变形程度循环调整切割路径,直到得到一个相对稳定的切割,这样将三维网格转化为二维图像然后可以利用传统图像处理的方法进行处理。随后又有许多研究对其进行改进,Carr 等人^[6]提出了四边形几何图像。Sander 等人^[7]提出了一种多片几何图像的方法,其核心思想是将三角网格划分为不规则的子块,并生成几何图像,以此作为原网格的规则化表示形式。Praun 和 Hoppe^[8]使用了可保持光滑性的参数化算法,将网格点参数化到球面上,该算法能够自动保证网格内部的 C^2 连续,以及矩形边界上大部分点的 C^2 连续,仅在四个角点上出现退化情形。Gobbetti 等人^[9]利用图像压缩技术实现了在网络上快速传输几何模型数据以及实时重建。近年也出现了一些基于 GPU 的几何图像 LOD 技术,例如 Ji 等

人^[10]提出基于 GPU 的 GIM 动态 LOD 技术。Niski 等人^[11]应用无缝纹理集表示三维几何模型,提出了多种 LOD 选择方法。Yao 等人^[12]在 AGIM 中提出一种 zipper 算法处理裂缝,但其实现复杂。Feng 等人^[13]提出了一种基于 GPU 的三角形几何图像的实时渲染,通过重新操作纹理坐标将高分辨率节点移动到低分辨率节点上来实现。

文献[14]针对零亏格三角网格模型提出一种特定边界的凸组合球面参数化方法,并将球面参数化结果转化为几何图像,然后在几何图像上建立二叉树^[15-16],通过对二叉树的遍历确定模型细节层次,达到对原网格进行简化并绘制。由于该方法在模型细节层次的选择、原始网格的简化以及绘制过程中需要大量计算,当模型规模逐渐增大时,CPU 负担较大。本文在其基础上进行改进,将二叉树模型以纹理的形式传送到片元着色器进行细节层次的选择,并对原始三角网格进行简化和绘制。最后通过对 Bunny 模型的绘制进行相关实验,保证了模型绘制质量,同时提高了绘制效率。

2 几何图像构建

参照文献[13]方法,以网格模型中心为球心作网格模型的包围球。从球心向网格模型每个顶点发出一条射线,该射线与球面的交点即为顶点在球面上的映射,若射线与网格模型除该顶点外没有其他交点则该顶点为非待求点,否则为待求点。对网格模型顶点进行遍历求得所有待求点,然后通过计算得到待求点的参数化结果。将得到的球面参数化结果通过线性变换映射球面内接的正八面体上,并展开为二维平面上的几何图像。几何图像构建过程如图 1 所示。

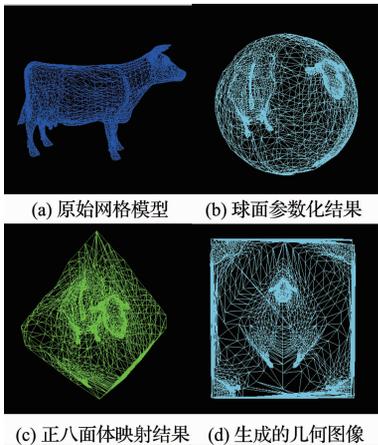


图 1 几何图像构建过程

3 基于 GPU 的多分辨率渲染

3.1 构建二叉树

将三角网格模型转化为二维空间上的几何图像,几何图像每一像素为模型表面采样点在二维空间的映射,像素存储的颜色值为该采样点的空间坐标。三维几何模型经过参数化得到的原始几何图像分辨率为 $2^n \times 2^n$,在数据预处理阶段,为了适应二叉树的结构,将分辨率扩展为 $(2^n + 1) \times (2^n + 1)$;然后将规模为 $(2^n + 1) \times (2^n + 1)$ 的节点分割为 4 个规模为 $(2^{n-1} + 1) \times (2^{n-1} + 1)$ 的子节点,每个子节点和相邻节点共享相同的一行一列。

二叉树结构中的每一个节点覆盖几何图像中一块矩形区域,用来表示表面几何时,上层的节点涉及的采样点较少,分辨率低,误差大,但绘制效率高;底层节点涉及的采样点较多,分辨率高,误差小,但绘制效率低。

构造几何图像二叉树结构的基本思想是:将整个几何图像作为二叉树的根节点,自顶向下递归构造树型结构,直至最终的子节点矩形区域只包含一个三角面或者构不成一个三角面。二叉树结构如图 2 所示。

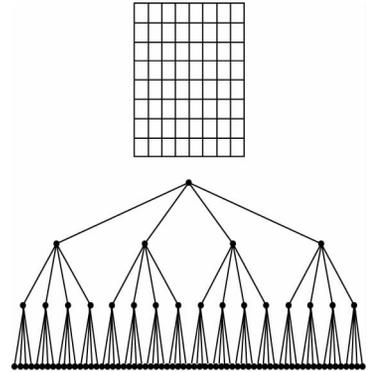


图 2 二叉树结构

针对二叉树结构特点,使用如下数据结构对节点进行描述:

```
Structquadnode
{
    intc_depth; //当前节点深度
    bool rendere; //节点是否可绘制
    boolisSplit; //节点是否可分割
    quadnoderect; //节点所代表的矩形区域
    intnumVerts; //当前矩形区域所含顶点数
    intnumTriangle; //当前矩形区域所含三角面
    quadnode * sub[4]; //指向节点的四个孩子
    quadnode * parent[4]; //父节点指针
    quadnode * neighbor[4]; //邻居节点指针
    vector<vertex> vlist; //当前矩形区域节点的存储容器
    vector<triangle> plist; //当前矩形区域三角面的存储容器
}
```

3.2 基于 GPU 的多分辨率渲染

首先对整个二叉树进行深度遍历,确定当前视点所在细节层次,然后从当前层次所在的节点一直遍历到二叉树的叶子节点,最后对叶子节点的三角面与相邻节点的三角面进行合并,实现多分辨率绘制。

基于GPU的多分辨率渲染过程主要分为两个部分:第一是多模型细节层次的选择,第二是对模型中的三角面进行合并并进行绘制。

目前GPU中,着色器(shader)替代了传统的固定渲染管线,可以实现3D图形计算,GPU可编程性可以不受显卡的固定渲染管线限制,实现图像高画质效果。着色器分为顶点着色器(vertex shader)和片元着色器(fragment shader),顶点着色器是一组指令,这组指令在顶点被渲染时执行,同一时间内只能激活一个顶点着色器。片元着色器与顶点着色器一样也是一组指令,这组指令在像素被渲染时执行。

与顶点着色器相比片元着色器在运算处理上具有更高的并行性,在数据访问上也有具有更高的灵活性,但是片元着色器输入数据的基本结构为图像纹理形式。将基于四叉树的几何图像作为纹理发送到片元着色器,同时将在CPU中计算的视点位置,误差阈值等参数也作为输入传送到片元着色器中,对纹理进行一次筛选即将符合要求的点进行绘制。对四叉树每层所对应的不同分辨率几何图像分别传入片元着色器进行筛选即可得到符合要求的模型,考虑到多次的筛选过程时间消耗较多,将基于四叉树的几何图像整理为如图3所示的结构形式。

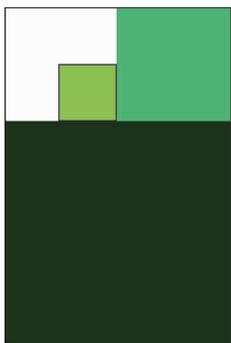


图3 几何图像纹理结构

图3下部对应四叉树最底层的节点,四叉树由底向上各层节点由右向左排列,将该结构形式作为纹理集输入片元着色器,并通过一次筛选得到待绘制的几何图像。筛选过程如下:

1)选取一个节点,若该节点为非空节点,则执行2),否则对下一节点进行测试。

2)为避免节点冗余先考察它的父节点,若父节点未通过筛选,则执行3),否则选取下一节点并返回步骤1)。

3)将节点的误差与全局误差阈值进行比较,若节点误差小于全局误差阈值则该节点通过测试并对节点进行输出,否则选取下一节点并返回步骤1)。

4)重复1)、2)、3)直至所有节点筛选完毕。

然后对未通过筛选的节点进行合并,合并是在拥有同一父节点的子节点间进行,过程如图4所示。具体步骤如下:

1)对四叉树进行遍历找到一个通过筛选的节点,在该

节点的子节点中选取包含一个完整三角面的节点,例如图4(a)中 f_1 。

2)判断 f_1 的边是否与邻接节点内的顶点构成一个三角面。边 v_0v_1 为三角面 f_1f_1 的公共边, f_1f_2 合并,取消顶点 v_0 与顶点 v_1 之间的邻接关系,由此构成新三角面 $\Delta v_0v_1v_3$ 。如图4(b)所示。

3)对每一个包含完整三角面的节点以及新生成的三角面执行步骤2),如图4(c)所示。

4)不断重复2)、3)直至步骤1)选取节点的所有子节点合并完成,如图4(d)所示。

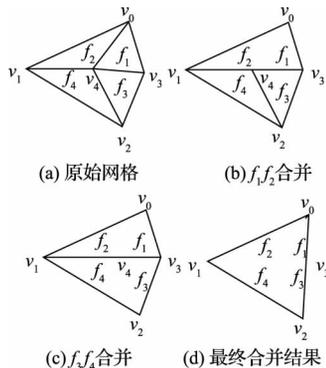


图4 三角面合并过程

将确定需要合并的三角面的顶点先传送到顶点着色器中进行处理,然后将这些三角面作为一个三角形扇传送到几何着色器中进行处理,得到合并后的结果并输出到光栅中显示,最后在片元着色器中进行颜色计算并绘制。

4 实验结果及分析

本文的实验在 Intel(R) Core(TM) i5-4200H CPU, 2.8 GHz, 8.00 GB 内存, Nvidia GTX850M 显卡, 编程语言 Visual C++, 最终效果通过 OpenGL, glsl 编程实现。选取 bunny 模型进行测试,当视点距离变化时的几何图像如图5所示。

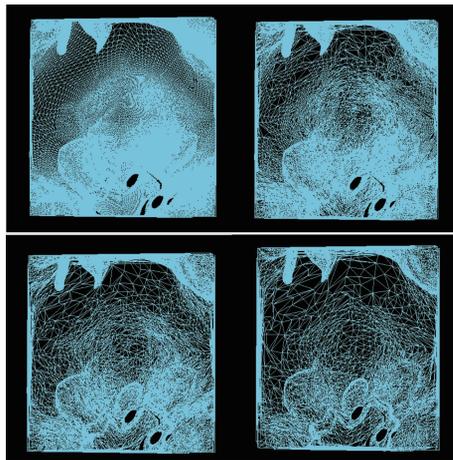


图5 不同分辨率几何图像

最终绘制效果如图6所示。



图6 不同分辨率的 bunny 模型绘制效果

由图6可以看出视点离模型越远,所绘制的三角面数量不断减少,而观察效果并未产生明显影响。将本文方法与文献[13]方法在绘制效率方面进行比较,比较结果如表1所示。

表1 两种方法的比较

三角面数量	本文方法/ms	文献[13]方法/ms
69444	6	17
52017	5	9
34627	3	5
17240	2	2

从表1中可以看出,文献[13]使用 CUDA 改进的光线追踪算法实现模型的绘制,当模型规模较小时效率很高,当模型规模逐渐增大时由于并行线程块数量的限制导致效率下降,本文的方法将多分辨率选择,模型简化以及绘制过程使用 GPU 的着色器进行处理提高了绘制效率。

5 结 论

将 LOD 细节层次的选择和三角面的网格简化过程使用 GPU 进行处理,提高了复杂三角网格模型绘制效率。针对更复杂的模型,由于顶点数量非常庞大,构建的四叉树模型底层节点数量也非常庞大,细节层次的选择过程需要的时间也相应增加,进一步工作研究提高并行运算速度的方法,减少时间的消耗,同时对参数化过程进行优化,减小模型形变程度,以达到更好的绘制效果。

参考文献

[1] LUEBKE D, ERIKSON C. View-dependent simplification of arbitrary polygonal environments[C]//Proceedings of the 24th annual conference on Computer graphics and interactive techniques. ACM Press/Addison-Wesley Publishing Co., 1997:199-208.

[2] BOUBEKEUR T, SCHLICK C. A flexible kernel for adaptive mesh refinement on GPU [C]//Computer Graphics Forum. Blackwell Publishing Ltd, 2008, 27(1): 102-113.

[3] DECORO C, TATARCHUK N. Real-time mesh simplification using the GPU[C]//Proceedings of the 2007 symposium on Interactive 3D graphics and games. ACM, 2007: 161-166.

[4] HU L, SANDER P V, HOPPE H. Parallel view-dependent refinement of progressive meshes [C]//Proceedings of the 2009 symposium on Interactive 3D graphics and games. ACM, 2009: 169-176.

[5] GU X, GORTLER S J, HOPPE H. Geometry images[J]. ACM Transactions on Graphics (TOG), 2002, 21(3): 355-361.

[6] CARR N A, HOBEROCK J, CRANE K, et al. Rectangular multi-chart geometry images [C]//Proceedings of the fourth Eurographics symposium on Geometry processing. Eurographics Association, 2006: 181-190.

[7] SANDER P V, WOOD Z J, GORTLER S J, et al. Multi-chart geometry images: Eurographics Association, US7265752 B2[P]. 2007.

[8] PRAUN E, HOPPE H. Spherical parametrization and remeshing[C]//ACM Transactions on Graphics (TOG). ACM, 2003, 22(3): 340-349.

[9] GOBETTI E, MARTON F, RODRIGUEZ M B, et al. Adaptive quad patches: an adaptive regular structure for web distribution and adaptive rendering of 3D models [C]// Proceedings of the 17th International Conference on 3D Web Technology. ACM Press, 2012: 9-16.

[10] JI J, WU E, LI S, et al. Dynamic lod on gpu[C]//Computer Graphics International 2005. IEEE, 2005: 108-114.

[11] NISKI K, PURNOMO B, COHEN J. Multi-grained level of detail using a hierarchical seamless texture atlas [C]//Proceedings of the 2007 symposium on Interactive 3D graphics and games. ACM, 2007: 153-160.

[12] YAO C Y, LEE T Y. Adaptive geometry image[J]. IEEE Transactions on Visualization and Computer Graphics, 2008, 14(4): 948-960.

[13] FENG W W, KIM B U, YU Y, et al. Feature-preserving triangular geometry images for level-of-detail representation of static and skinned meshes[J]. ACM Transactions on Graphics (TOG), 2010, 29(2): 397-408.

- [14] 朱杰, 刘惠义, 张银川. 特定边界的凸组合球面参数化[J]. 系统仿真学报, 2013, 25(9):2184-2187.
- [15] 李白云, 赵春霞. GPU实时构建四叉树的快速地形渲染算法[J]. 计算机辅助设计与图形学学报, 2010, 22(12):2259-2264.
- [16] 张兵强, 张立民, 艾祖亮, 等. 屏幕空间自适应的地形 Tessellation 绘制[J]. 中国图象图形学报, 2012,

17(11):1431-1438.

作者简介

王化雨, 1990年出生, 硕士研究生, 主要研究方向为计算机图形学。

刘惠义, 1961年出生, 博士, 教授, 博士生导师, 主要研究方向为计算机图形学、虚拟现实、计算可视化。

(上接第21页)

- [11] 人机交互的进化与未来[J]. 互联网周刊, 2014(3): 56-57.

E-mail: linzhihua123@yeah.net

平雪良, 1962年出生, 博士, 教授, 主要从事机电一体化、机器人相关技术研究。

E-mail: ping@jiangnan.edu.cn

作者简介

林智骅, 1994年出生, 机械电子工程专业本科生, 从事机器人方向研究。

(上接第30页)

- [9] 邢孟道, 保铮. 一种逆合成孔径雷达成像包络对齐的新方法[J]. 西安电子科技大学学报, 2000, 21(3):93-95.
- [10] 李建阳, 常文革, 王亮. 一种应用于条带 SAR 的改进 PGA 算法[J]. 现代雷达, 2007, 29(8):52-53.
- [11] 金光虎, 高勋章, 黎湘, 等. 基于扩展距离像序列的 ISAR 成像相位补偿方法[J]. 系统工程与电子技术, 2011, 33(1):58-63.

E-mail: 15600581431@163.com

王俊岭, 博士, 北京理工大学讲师, 研究方向空间目标探测, 雷达信号处理。

E-mail: emailwj1@gmail.com

马幼鸣, 博士, 北京理工大学副教授, 研究方向为嵌入式系统应用。

E-mail: mym62272561@bit.edu.cn

郭宝锋, 博士研究生, 主要研究方向为雷达信号处理、雷达成像技术。

E-mail: guobao_feng870714@126.com

作者简介

倪书爱, 工学硕士, 北京理工大学在读硕士研究生, 主要研究方向为 ISAR 成像, 雷达信号处理等。