大规模地形快速渲染算法的研究①

涂　超, 毋河海, 王新生

(武汉大学 资源与环境科学学院 武汉　430070)

摘要　探讨了如何有效地进行大规模地形的快速绘制, 并详细介绍了 一个基于 mip- map 技术的地形渲染 算法的原理及其实现过程.

关键词: 渲染; LOD; mip- map

中图分类号: P208　　文献标识码: A　　文章编号: 1007- 855X(2002)01- 001- 05

目前能用于大规模地形快速绘制的算法已有很多, 例如采用视截体(view- frustum)对不可见地形区块进行裁剪的简单四叉树、基于多细节层次(LOD)的模型等, 所有这些技术都是为了减少绘制时的三角形的数目, 以加快绘制的速度. 这些快速绘制技术的结合运用能进一步提高渲染的速度, 实际上只要所需绘制的三角形数目三维硬件能正常进行处理, 并且不会引起 CPU 的计算超载即可. 在此将要介绍的是一种利用 mip- map 技术快速实现大规模地形渲染的算法.

1　地形模型的建立

地形模型可采用多种方式来表示, 在此我们选择一种当前许多实时地形渲染应用中广泛采用的地形网格来表达地形. 采用地形网格的原因是: 1) 容易实施; 2) 需存储的内容少(仅需保存网格的尺寸和各个结点的对应高程值); 3) 易于实现多细节层次的表达. 但这种地形数据的表示方式的缺点是不支持具有悬垂地形的模型, 这是由于地形网格是一个具有高程值的二维网格, 其每一个结点仅具有一个高程值, 无法描述对于同一平面坐标对应多个不同的高程值的情况.

地形网格是一个在垂直和水平方向各有"$2^n + 1$(其中 $n \in [1, -> 1]$ 个结点、每一结点具有相应的高程值的规则网格, 网格将地形分割成 2^n 个矩形, 每一矩形由 2 个三角形组成. 三角形是进行绘制时的基本图元. 每一个结点的 x 和 z 分量被设定为固定的值, 在整个处理过程中不再改变, 结点的 y 分量代表该处地形的高程值. 在地形加载时读入. 地形的高程值可以从一个维数与地形网格相同的具有 8 位灰度值的位图文件中生成. 可获得一个能满足视觉效果的地形模型. 对于专业人员可采用三维空间离散的采样值来建立某一区域的数字地形模型, 通过对随机分布的采样点的进一步加工, 获得一定密度的规则的网格式数字地形模型, 这也是生成精确地形的一种有效的方法.

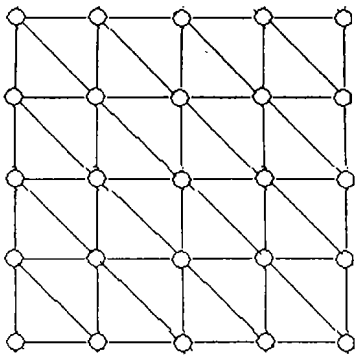

图 1　一个地形区块的网格的顶视图

接着将整个地形划分成多个区域, 每个区域具有固定的尺寸, 但必须是 $2^n + 1$. 这是为进行四叉树、视截体裁减作准备. 这些区块将作为地形模型的多细节层次表达中的 0 层实体. 如图 1 所示, 图中的地形区块有 5 × 5 个点, 实际应用中地形区块的尺寸可以根据情况自由选择. 采用这种地形区块分布的优点是: 地形区块有利于绘制时的优化处理, 可采用对区块绘制的重复调用实现对整个地形绘制. 若利用结点在网格中的索引会更方便. 采用地形区块的缺点是: 它与邻近的区块共享四个边上的所有结点, 这

① 收稿日期: 2001- 04- 09;
第一作者简介: 涂　超, 男, 1966 年生, 博士; 主要研究方向: 地理信息系统、虚拟现实.

© 1994-2012 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net

些结点进行两次处理.

2 视截体裁剪的基本原理[1]

在实际应用中, 地形的大部分区域对于观察视点来说往往都是不可见的. 因此, 它们并不需要渲染, 应事先裁剪掉, 以避免无谓的计算, 加快地形绘制的速度. 一种有效的快速裁剪方法是采用四叉树的数据结构. 四叉树在地形加载时生成, 仅由三维的包围盒组成. 整个地形的尺寸连同其各个地形区块的尺寸决定了四叉树的实际深度. 四叉树的结点由一个包含其所有所属子树的包围盒的三维包围盒组成. 其叶结点就是一个实际绘制的区块的包围盒, 并与包围盒所包含的地形区块相链接. 采用四叉树的原因是网格地形的分布是一个二维的分布, 因此可用二维的组织形式来存储. 若叶结点的包围盒部分或完全位于视截体内时, 该地形区块可标记为可见. 遍历整个树, 从根结点开始裁剪和处理, 将各个地块标记为可见或不可见. 这样就可得到要进行渲染地块的集合. 但由于地形通常十分复杂, 除非采用尺寸很小的地形, 否则很难达到所需的帧速. 即使要用了视截体裁剪算法也无法满足大规模网格地形的应用要求. 要进一步减少绘制时的三角形数目还需考虑进一步采用 LOD 算法. mip- map 技术是用来实现多分辨率纹理的一种有效的方法, 能大量减少对纹理缓冲区的占用. 由于地形网格可以看作具有不同高程值的二维分布, 因此可考虑采用 mip- map 技术来实现对地形网格的简化, 以实现进一步减少绘制三角形数目的目的.

3 mip- map 技术分析

mip- map 技术的基本思想是以适当大小的正方形来近似表达每一象素在纹理平面上的映射区域, 并预先将纹理图象表达为具有不同分辨率的纹理数组, 作为纹理查找表, 其中低分辨率的图象由比它高一级分辨率图象取平均得到. 简单地将低一级图象每边的分辨率取为高一级图象每边分辨率的二分之一, 而同一级分辨率的纹理数组则由红、绿、蓝三个分量的纹理数组组成. 由于这一查找表包含了同一纹理区域在不同分辨率下的纹理颜色值, 故称为 mip- map. 由 mip- map 表中的图象序列的生成方法可知, 对每一个纹理颜色分量图象分辨率的压缩过程可用一个四棱锥来描述. 该四棱锥的总层数为: $[\log 2^S] + 1$, 其中 S 为初始纹理图象每边的分辨率. 若最底层图象为给定的初始图象, 则第二层图象可由最底层图象与边长为 2 个象素的正方形滤波器作卷积运算而得. 一般地, 第 $k+1$ 层图象可由第 k 层图象在 2×2 的象素窗口内作相同的卷积运算而得到. 这意味着, 第 $k+1$ 层图象可由最底层图象与边长为 $2k$ 个象素得正方形卷积运算而得. 例如, 某一纹理图象的 mip- map 表的层数为 10 层, 每 10 层图象(顶层) 为一个象素, 它由原始图象(第 1 层) 经与边长为 512 个象素的正方形滤波器作卷积运算压缩得到, 其分辨率为 1×1. mip- map 方法在确定屏幕上每一象素内可见面的平均纹理颜色时需计算三个参数, 即屏幕象素中心在纹理平面上映射点的坐标(u, v)和屏幕象素内可见表面区域在纹理平面上所映射区域的边长 d, 其中(u, v)取为屏幕象素内可见表面在纹理平面上近似正方形映射区域的中心, d 取为该近似正方形的边长. 显然, d 决定了应在哪一级分辨率的纹理图象平面上查取 mip- map 表[2].

从上可知, mip- map 技术需要为每一个纹理建立一个 mip- map 纹理查找表, 其中的第一项是实际的初始纹理, 其后每一项的分辨率都是前一项的一半, 直到达到所希望的层数. 当纹理距视点的距离为 d 时, 从 mip- map 纹理查找表中选择一恰当的层级, 代替实际纹理进行渲染. 我们可考虑将 mip- map 技术也应用到三维的网格上.

4 mip- map 技术在地形网格上的应用

4.1 基本思想

在实际应用中, 当某一地形区块距离视点很远时, 并不需要采用与距视点较近的地形区块相同的细节来进行渲染, 可采用一个更低分辨率的近似模型来代替它, 以动态减少绘制的三角形数量, 提高渲染的速

© 1994-2012 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

度. 这就是 LOD 技术的基本思想[3]. 在此我们将介绍利用 mip- map 方法来实施 LOD 技术的过程. 设 mip- map 中的高分辨率纹理的三维等价体为地形区块, 运用 mip- map 思想通过减少地形区块的结点数来获得不同分辨率的地形模型以建立一个 mip- map 地形查找表(如图 2 所示, 为一个具有两个 mip- map 层级的地形区块). 由于地形区块的尺寸是相同的, 此方法可直接应用于整个地形的各个区块, 在地形加载时进行计算并将计算出的 mip- map 地形查找表存储在内存中. 下面将讨论如何使用不同分辨率的地形区块, 其层次从 0 到 N (其中 $N \in [1, ->]$ 为最低分辨率层次).

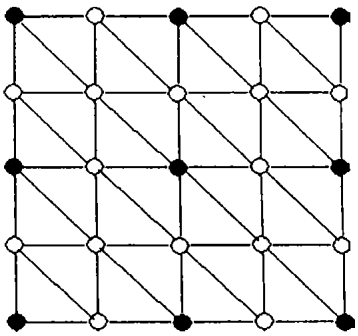

图 2　一个具有 2 层分辨率的地形 mip- map 网格

4.2　地形 **mip- map** 层次的选择

建立了 mip- map 地形查找表后, 下一步就是要确定当地形区块距视点的距离为 d 时, 应如何为地形区块选择恰当的分辨率层级. 简单的方法是预先为每一个 mip- map 地形查找表中的层级设定一个距离 d, 但这会导致突跳现象的出现. 即当以一个固定的距离作为从一个层级转换到另一个层级的条件时, 几何形状突然发生较大的改变. 这是由于许多结点突然从模型中删除或添加到模型造成的. 见图 2, 当从 0 层切换到 1 层时, 0 层级中白色的点将被删除. 这会引起在抵近观察时, 地形的不正确. 在纹理 mip- map 中层级的选择计算是由硬件完成的, 我们无须关心. 对于地形的 mip- map 可采用同样的思想, 以减少突跳的几率.

当从地形 mip- map 的 0 层切换到 1 层时, 细节的减少将导致不正确的地形出现. 这是由于结点被同删除引起的, 导致了高程 h 的变化(见图 3 所示) . 根据透视原理, 当距离 d 越大时, 高程 h 就越不明显, h 与 d 成反比. 我们可将 h 投影成屏幕空间中的长度 L, 这是用户最终所能看到的. 当 L 大于某一阀值 τ 时, 误差会很明显, 因此不允许切换更高的层级, 直到 L 比 τ 更小时才允许切换. 但由于每一地块的 mip- map 层级含有多个此类误差(每个被删除的结点都会引起误差), 需考虑将哪一个结点的 h 投影成屏幕空间中的长度 L. 我们可以从 $\max(h_0, h_{n-1})$ (其中 n 为地形 mip- map 的层级中被删除的结点数) 中取得 h 来计算 L. 若该 L 比 τ 小, 则整个地块 mip- map 层级的误差就比 τ 小. 若该 L 比 τ 大, 则至少存在一个明显的误差. 在建立地形 mip- map 时就预先选择并保存 h. 后面所提到的 h 或 L 就是指 $\max(h_0, h_{n-1})$ 或 $\max(L_0, L_{n-1})$ (其中 n 为地形 mip- map 的层级中被删除的结点数).

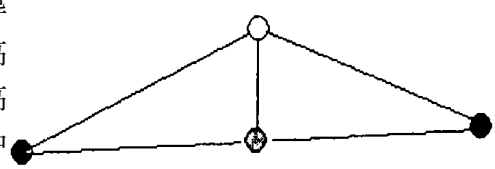

图 3　图中白点删除后, 将导致高程的丢失, 结点由灰点取代

对于给定的 d, 要决定如何选择一个合适的地形 mip- map 层级, 需从 0 层开始, 将当前层级的 L 值与 τ 比较. 若 $L > \tau$, 则切换到更高层级将引起明显的误差. 否则可以切换到更高层级. 若下一层级的 L 也比 τ 小, 则继续切换, 直到达到正确的层级为止.

4.3　加快地形 **mip- map** 层次的选择

由于三维硬件的栅格处理能力能绘制大量的三角形, 我们没有必要去计算出具有最小数目的三角形几何体. 若采用一个不很精确的算法可减少 CPU 的负载, 只要生成的三角形数目不超过三维硬件的处理能力, 就可以尝试着采用.

在 mip- map 层次的选择中, 需为每一帧中的每一个可见地块的 mip- map 计算 L 并与 τ 相比较, 这是一个相当大的计算量. 由于 L 是由 d 以及视点的位置决定的, 无法预先为每一个 d 求出 L 并保存到查找表中. 但若将视点的方向矢量看作是一个水平矢量(不考虑其垂直分量), 就可以预先求得 L, 但结果可能会不太精确. 至此我们基本上已经使 CPU 的计算负载降到了很低的程度.

要进一步加快处理的速度, 在地形载入时可为每个 mip- map 层计算一个扩充的数据字段, 将其设置

© 1994-2012 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net

为该层级能被采用时的最小距离, 记为 minimum $d(D)$. 现在, 在为每一帧的每个可见的地形 mip- map 选择合适的层级时, 仅需将 d 与 D_n(n 是地形的层级数) 相比较. 可采用以下公式为每个地形 mip- map 层级预计算 D_n, 在此设置一个对称的视截体, 其参数为(l, r, b, t, n, f):

$$D_n = | h | \cdot C$$

其中 $C = \frac{A}{T}$, $A = \frac{n}{| t |}$, $T = \frac{2 \cdot \tau}{v_{res}}$, (其中 n 是接近视点的裁剪平面, 为视截体的顶部裁剪平面, v_{res} 是屏幕的垂直分辨率). 由于每一次都需实时地计算视点到地形 mip- map 的距离, 可用比较 D_n^2 和 d^2 来替代 D_n 和 d 的比较以减少计算时间, 公式如下:

$$D_n^2 = h^2 \cdot C^2$$

$$d^2 = (e_x - c_x)^2 + (e_y - c_y)^2 + (e_z - c_z)^2$$

(式中 e 是$[e_x, e_y, e_z]$ 为视点的当前位置, C 是$[e_x, e_y, e_z]$ 为当前正在处理的地形区块的中心).

4.4 地块拼接时几何裂缝问题的解决

当地形的两个邻近的区块具有不同的细节层次时, 具有较高细节的地块具有更多的结点, 比细节少的地块具有更多的高程信息, 当两者共享边界时, 就会出现缝隙. 这是任何地形渲染应用都不能接受的, 必须避免. 解决这个问题的基本原则就是要调整结点的连接关系, 以使两者的边界真正彼此吻合. 有几种方法可供选择.

一种方法是在两个邻接区块中的具有较低细节区块的边界上添加结点, 以使其与高细节区块的边界相互吻合. 这导致较低细节区块的结合分布以及结点的连接关系的动态改变, 并要随邻接区块层级的改变不断更新. 因此, 需生成并保存区块的原始地形 mip- map, 既慢又耗费内存, 这一方法应尽量避免使用.

另一方法是减少具有较高细节地块的边界上的结点, 以与低细节地块的边界相吻合, 但会引起高细节地块的失真.

第三种方法是不改变边界上结点, 仅对较高细节地块的结点的连接特性进行调整. 见图 4. 图中的地块为一个与另一个较低细节的地块共享左边界的具有较高细的地块. 灰色的结点是引起裂缝多余的结点. 通过忽略这些结点, 并重排与之相连的边, 就可实现与低细节地块的无缝吻合. 可通过对边界采用多个三角形扇形(triangle- fans) 来实现快速绘制. 其他的结点按正常方式绘制. 对四个边都要进行类似的处理. 此方法的缺点是有可能造成地形表面明暗度的少许变化, 在实时交互中, 这点变化根本就察觉不到.

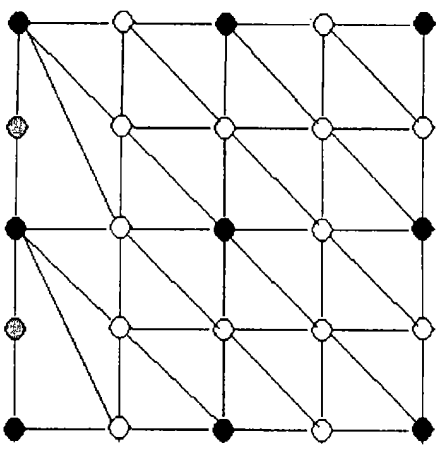

图 4 对具有较高细节的地块的结点间的连接关系进行调整

从以上分析可知, 第三种方法不需要增加或减少结点, 有效地填补了裂缝, 渲染速度快, 不失为解决裂缝问题的最佳方法.

4.5 算法小结

前面对将 mip- map 技术应用地形网格的思想以及需要考虑的问题进行了讨论, 在此对其整个实施过程进行总结. 要实现地形 mip- map, 需对每个地块实施以下步骤:

1) 加载地块;

2) 计算地块的包围盒并将其存入恰当的叶结点中;

3) 为此地块计算其所有的地形 mip- map 层级;

4) 为每一个地形 mip- map 层级计算 D_n, 并存入该地块的四叉树叶结点中;

5) 从内存删除所有的地形几何层级的信息, 对下一个地块重复执行以上步骤.

© 1994-2012 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

经过上述处理, 就得到了整个地形的四叉树, 每一个叶结点包含其所有地块的 D_n 值, 注意我们并不需要存储任何地形 mip- map 的模型数据, 保存的仅是其相应的描述参数, 具体数据仍保存在地形网格中.

运行时, 四叉树的可见叶结点的 D_n 值与当前的 d 值相比较, 并生成合适的地形 mip- map 层级. 若 d 比当前地形 mip- map 层级的 D_n 小, 则需进一步生成新的地形 mip- map 层级.

5　结论

大规模地形模型的快速渲染一直是三维图形学领域研究的重点, 它是多种不同技术的集成应用, 涉及视域裁剪、模型简化、三维硬件的利用、LOD 算法、合理的数据结构等多个方面的算法, 对每个算法的改进都不同程度地加快渲染的速度. 改进的目的在于减少绘制的三角形数量、减少处理的时间以及合理地组织数据等方面. 本文介绍的利用 mip- map 技术来实现对三维地形的快速绘制的方法, 有效地实现了上述目的, 是一种高效的地算法.

参考文献:

[1] Donald hearn, M. Pauline Baker. 计算机图形学[M]. 北京: 清华大学出版社, 1998. 431~ 492.

[2] 彭群生等. 计算机真实感图形的算法基础[M]. 北京: 科学出版社, 1999. 249~ 255.

[3] Peter Landstrom, David Koller, et al. Real Time: Continus Level of Detail Rendering of Height Fields[J]. Proceedings of ACM SIGGRAPH 96, August 1996. 109~ 118.

Research on the Rendering Algorithms of large- Scale Terrain

TU Chao, WU He- hai, WANG Xin- sheng

(Institute of Resource and Environment Science, Wuhan University, 430070, China)

Abstract The paper discusses how to implement fast large- scale terrain rendering. It introduces the principle and process of a terrain rendering algorithm based on mip- map technology in detail.

Key words: rendering; LOD; mip- map

© 1994-2012 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net