

基于四叉树包围球和屏幕误差的 LOD 算法

王 倩¹, 高保禄¹, 高锐军¹, 阴桂梅²

(1 太原理工大学 计算机科学与技术学院, 山西 太原 030024; 2 太原师范学院 计算机科学与技术系, 山西 太原 030024)

摘 要: 针对大规模地形数据庞大、帧速低的问题, 提出一种基于四叉树包围球和屏幕误差的 LOD 算法. 该算法对地形进行分块编号, 实现部分地形读取; 预处理阶段对分块地形构造包围球, 减少数据存储量和实时阶段计算量. 实时绘制阶段, 依据基于投影和包围球的视锥裁剪方法实时调入可见地形块, 结合四叉树结构建立基于包围球屏幕误差的 LOD 模型, 实现大规模地形的实时绘制. 实验表明该方法可提高帧速率, 保证大规模地形绘制的流畅性和实时性.

关键词: 地形; 四叉树; 包围球; 屏幕误差; LOD

中图分类号: TP391.9

文献标识码: A

文章编号: 1000-7180(2016)05-0127-06

LOD Algorithm Based on Enclosing Ball Screen Error of Quad-tree

WANG Qian¹, GAO Bao-lu¹, GAO Rui-jun¹, YIN Gui-mei²

(1 College of Computer Science and Technology, Taiyuan University of Technology, Taiyuan 030024, China;

2 Computer Science and Technology Department, Taiyuan Normal University, Taiyuan 030024, China)

Abstract: Aiming at the problem of the huge terrain data and low frame rate in large-scale terrain rendering, this paper proposes a LOD algorithm based on Enclosing Ball Screen Error of Quad-tree. It divides many blocks and indexes data blocking for terrain data, implementing the local loading of terrain data. In the preprocessing phase, the algorithm builds the enclosing ball for every terrain block to reduce the data storage and computation in real-time rendering stage. In real-time rendering stage, it establishes LOD model based on enclosing ball and implements the real-time rendering for large-scale terrain, according to the viewing frustum culling based on projection and bounding sphere to loading terrain blocks, combined with the data structure of quad-tree to build bounding spheres. Experiments results show that the method can improve the frame rate to ensure smooth large-scale terrain rendering and the performance of real-time.

Key words: terrain; quad-tree; enclosing ball; screen error; LOD

1 引言

大规模地形的三维实时绘制在场景仿真、飞行模拟以及地理信息系统等各个领域有着广泛的应用, 同时也是虚拟现实领域的关键技术之一. 地形的高程数据与纹理数据之庞大, 无论是在数据的存储还是在数据的调度上都给三维渲染的实时性带来了很大的挑战. 因此, 如何高效地进行海量地形数据的处理与调度, 实现大规模地形的实时绘制一直是地

形可视化领域研究的热点与难点.

细节层次模型(level of detail, LOD)是解决大规模地形实时绘制问题的一种经典算法. 文献[1-2]采用静态的 LOD 模型虽然简单, 但绘制时因为在特定时刻地形块分辨率处处相同, 导致大量的数据冗余, 给实时绘制带来了很大的障碍; 文献[3]虽然采用了动态 LOD 模型, 但只根据物体与视点的距离单一判断地形块的细节层次, 没有考虑投影到屏幕上的绘制误差, 渲染地形时视觉上可能会有较大

收稿日期: 2015-07-09; **修回日期:** 2015-08-31

基金项目: 虚拟现实技术与系统国家重点实验室开放基金项目(BUAA-VR-15KF-17); 山西省基础条件平台项目(2014091004-0104)

的跳跃感;文献[4-5]在使用动态 LOD 模型的同时建立基于屏幕误差的 LOD,达到了较好的绘制效果,但文献[4]需要存储每个数据块采样点的信息存储量大,也给数据计算带来了负担.

基于此,本文从规则网格出发,在预处理阶段对地形进行分块编号,并对各个地形块构造相应的包围球,减少数据存储的同时减少渲染阶段的计算量;在地形绘制时,结合四叉树的数据结构,根据视点的位置进行基于四叉树包围球结构的视锥裁剪和包围球基础上的屏幕误差的 LOD 生成,弱化由于不同 LOD 造成的视觉跳跃感;同时,在绘制阶段两次使用包围球,避免了分别计算视锥裁剪范围和连续 LOD 的计算冗余,保证大规模三维地形绘制的流畅性和实时性.

2 地形数据预处理

如图 1 所示,本文将整个地形绘制过程分为预处理阶段和实时绘制阶段.在预处理阶段对地形数据进行分块编号,并且构造基于四叉树的包围球,不仅在数据空间上减少了存储量,还减少了实时绘制阶段的数据计算量.

地形数据块块数为 $(2^{m-n}) \times (2^{m-n})^{[5]}$.图 2(b)所示为地形数据块分块的结果.地形数据分块后,对每一个数据块的信息进行保存,作为预处理阶段构造包围球的基本单位,同时,在视锥裁剪阶段,根据视点的变化,判断每个数据块的可见性.

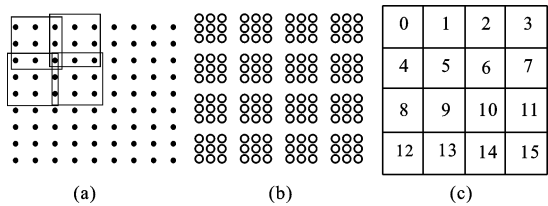


图 2 地形数据的分块以及编号

对地形数据分块后,为了数据块提取检索方便,根据空间结构,对数据块进行从上到下,从左到右的编号工作.编号从 0 开始,如图 2(c)所示.对数据块进行这样的编号,一方面计算相对简单,不会增加太多的计算量;另一方面可以根据数据块的编号很容易的找到地形数据,方便部分地形数据的提取.

2.2 基于四叉树的包围球构造

在构造包围球时,三角形二叉树以及四叉树是多用的数据结构.本文采用基于四叉树结构的包围球,大大减少了地形实时渲染阶段的计算量.

在地形渲染过程中,运行速度是一个很重要的判断依据.地形实时绘制时,根据每一帧中视点位置的不同,判断数据块的可见性以及每个可见数据块的 LOD.虽然在漫游过程中,视点的位置是不可预知的,但是每个数据块的基本几何信息却是不变的.因此,算法通过在预处理阶段对每个地形块进行预处理,提前计算每个数据块的一些信息参数,进而提高在实时渲染时的速度.在该阶段,只根据地形数据分块结构按照空间顺序得到地形块的各个结点,并进行相关参数的计算以供实时调用,但并不决定采用哪些结点以及调用哪些参数(这些都在实时渲染阶段进行),从而减少了数据渲染时的部分运算,提高渲染速度.

在对地形进行分块之后,对每个地形块构造包围球结构,以供在实时渲染时,视锥裁剪判断以及计算屏幕误差判断地形是否细分使用.

构造一个正方形的包围球,如图 3 所示.其中,四边形 ABCD 是正方形的数据块, O_i 为 ABCD 的对角线交点,构造的包围球是以 O_i 为球心,半径为 R_i 的球体, $R_i^2 = \epsilon_i^2 + c^2$, c 为正方形的对角线长的一半, ϵ_i 是取正方形区域内所有采样点和对角线交点之间的最大高程差.但这里,为了减少数据的存储量, ϵ_i 与球心处所存在的空间几何误差用同一个值存储.

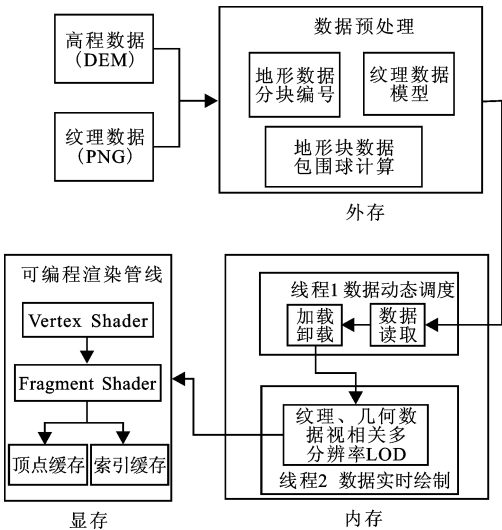


图 1 大规模地形绘制的过程

2.1 地形数据分块及编号

为满足建立基于四叉树结构的细节层次模型,规则方形网格模型(regular square grid, RSG)空间地形分块要求每个地形块大小满足 $(2^n+1) \times (2^n+1)$ (不足部分用空值补齐),并且为了方便地形绘制时的裂缝处理,要求两个地形块之间有重复数据.对于 $(2^m+1) \times (2^m+1)$ 的地形数据,可以划分成多个相同的 $(2^n+1) \times (2^n+1)$ 数据块($n < m$),则分成的

球心处的空间几何误差为

对于节点 i , 有

$$\epsilon_i = |Z_a + Z_b + Z_c + Z_d/4 - Z_i| \quad (1)$$

式中, Z_a 、 Z_b 、 Z_c 、 Z_d 分别为正方形四个端点 A 、 B 、 C 、 D 处的高程值; Z_i 为对角线交点(即球心)处的高程值. 图 3 为以地形块正方形 $ABCD$ 为水平面, 以 O_i 为球心, HdD (最大高程值) 为半径, 构造的地形块 $ABCD$ 的包围球.

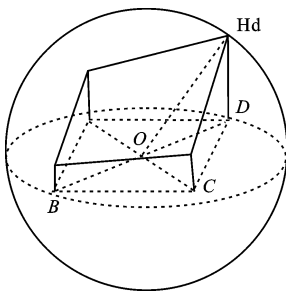


图 3 包围球的构造

包围球半径为

$$\epsilon_i = \max\{|Z_a - Z_i|, |Z_b - Z_i|, |Z_c - Z_i|, |Z_d - Z_i|, \left| \frac{Z_a + Z_b + Z_c + Z_d}{4} - Z_i \right|\} \quad (2)$$

$$c = \frac{1}{2} \sqrt{(X_a - X_c)^2 + (Y_a - Y_c)^2} \quad (3)$$

$$R_i^2 = \epsilon_i^2 + c^2 \quad (4)$$

式中, X_a 、 X_c 、 Y_a 、 Y_c 分别为 A 、 B 点在水平面上的横纵坐标.

3 基于投影和包围球结构的视锥裁剪

在特定场景进行漫游时, 人眼的可见范围有限, 即视锥范围为可见范围, 图 4(a) 为人眼视锥结构. 因此, 在建立 LOD 模型时, 只需绘制当前视点所在位置的视锥范围. 本文采用的方法虽与文献 [6] 中的一种方法相似, 都是将视锥体投影到 $x-y$ 平面上, 但本文只对视锥投影, 而因为在构造包围球时选择的水平面与 $x-y$ 平面一致, 因此不需对地形块包围球进行投影即可以判断地形块的可见性, 减少了投影操作的计算. 在具体判断时, 如果视锥体投影的三角形与地形块的包围球有交叉, 则认为该地形块在视锥范围之内, 属于可见地形块, 如图 4(b) 所示. 同时, 本文在视锥裁剪和判断 LOD 模型时, 都使用包围球结构, 减少了分别计算视锥裁剪方法和 LOD 模型的计算冗余, 保证了实时绘制的实时性.

基于以上方法, 判断地形块是否可见, 只需判断地

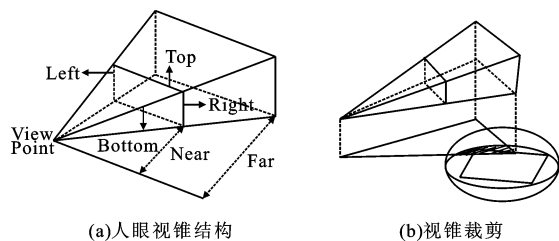


图 4 视锥结构以及基于投影和包围球的视锥裁剪

形块包围球的球心到视锥体投影三角形的三条边的距离, 如果三个距离都大于包围球的半径, 则认为地形块不可见; 否则认为地形块可见. 具体公式如下:

三角形三条边的直线公式为

$$AX_i + BY_i + C_i = 0 \quad (i = 1, 2, 3) \quad (4)$$

计算球心点 O_i 到三条边的距离公式为

$$d_i = \frac{|AX_i' + BY_i' + C_i|}{\sqrt{A^2 + B^2}}$$

(X_i' , Y_i' 为球心 O_i 在水平面上的坐标) (5)

设以 O_i 为球心的包围球半径为 R_i , 若满足

$$d_i \leq R_i \quad (i = 1, 2, 3) \quad (6)$$

则认为该地形块在视锥范围之内, 属于可见地形块.

当认为数据块可见时, 根据图 2(c) 的编号, 计算可见数据块的坐标点, 读取相应地形数据. 地形块数据具体读取时,

$$m = \text{Block_Index} \bmod (\text{hblock}),$$

$$n = (\text{Block_Index} - m) / \text{hblock}.$$

式中, m 和 n 表示该数据块的列号和行号, Block_Index 是数据块的编号, hblock 是每行所包含的数据块. 若用 Block_first 来表示每个地形块的第一个数据, row 来表示整个地形数据块的行数, Block_width 表示每个地形块的行数, 则有

$$\text{Block_first} =$$

$$(n \times \text{row} \times (\text{Block_width} - 1) + m \times (\text{Block_width} - 1)) \times \text{sizeof}(\text{double})$$

当要读取确定的数据块中的第 i 行第 j 列数据时, 对应公式如下:

$$t_HeightMap[i \times \text{Block_width} + j] = (\text{Block_first} + i \times \text{row} + j) \times \text{sizeof}(\text{double}).$$

具体实现时, 用 $t_HeightMap[]$ 一维数组来存取每个地形块的数据.

4 基于四叉树包围球和屏幕误差的 LOD 生成算法

在基于 RSG 规则网格划分的地形实时绘制算法中, 连续 LOD 生成. 本文采用四叉树结构, 对每

个可见地形块进行孤立分割,分别建立对应的四叉树,进行基于四叉树包围球结构的误差函数的动态 LOD 生成。

4.1 误差函数设计

节点的误差一般从两方面讨论,即空间几何误差和屏幕误差。节点的空间几何误差与视点无关,在渲染过程中是固定不变的。因此,在预处理阶段,对地形数据的采样点提前进行空间几何误差的计算,以供渲染时调用。而屏幕误差是空间几何误差投影到屏幕之后的误差。在三维可视化领域,所有的物体最终都要投影到屏幕上,因此一般采用屏幕误差作为节点重要度的判断依据。在进行地形实时渲染时,若能够控制该误差在一定的范围内,就可以解决由于不同 LOD 层次带来的视觉上的跳跃感,保证视觉上地形的流畅渲染。

基于包围球的屏幕误差,是一个与视点、节点空间的几何误差和包围球半径有关的数学函数。用 $\rho(\epsilon, R, e)$ [5] 表示屏幕误差,其中 ϵ 是与视点无关的节点空间几何误差, R 为某地形块的包围球半径, e 为视点。本文取方形地形块的中心为地形剖分点,令屏幕误差阈值为 τ ,根据式(1) 计算几何误差,再利用式(2) ~ (4) 得到节点 i 的屏幕误差函数:

$$\rho(\epsilon_i, R, e) = \lambda \frac{\epsilon_i}{d_i - R_i} \quad (7)$$

式中, $\lambda = \frac{\omega}{2 \tan \frac{\varphi}{2}}$, ω 是视域广角 φ 的像素个数, d_i 为

视点位置到包围球球心的距离, R_i 为节点 i 包围球的半径。当进行地形是否细分时,和屏幕误差阈值 τ 进行比较,若屏幕误差大于阈值,则地形需要进一步细分;否则,地形不需要再细分,直接渲染即可。节点 i 的判断以及转换具体如下:

$$\begin{aligned} \text{divide}(i) &\Leftrightarrow \rho_i(\epsilon_i, R_i, e) > \tau \\ &\Leftrightarrow \lambda \frac{\epsilon_i}{d_i - R_i} > \tau \Leftrightarrow \frac{\lambda}{\tau} \epsilon_i > d_i - R_i \\ &\Leftrightarrow \left(\frac{\lambda}{\tau} \epsilon_i + R_i \right)^2 > d_i^2 \end{aligned} \quad (8)$$

本文设计的误差函数,在绘制阶段,只需要根据视点的位置计算每个地形块的包围球与视点的距离 d_i ,只有在进一步细分时,才需要进一步计算细分地形块的包围球。预处理阶段的数据处理,为绘制阶段的实时性提供了更大的可能。同时,在存储空间上,只需要存储视锥范围内的地形块对应的四叉树的节点的空间几何误差,并且采用包围球的结构,也进一步减少了存储空间。最后,在整个算法中,在视锥裁

剪以及判断 LOD 模型时,都采用包围球的结构,避免了在这两个步骤中分别采用不同的结构进而计算其相关参数的数据计算冗余,进一步保证了大规模地形的实时绘制。

4.2 算法步骤

本文采用自上而下的方式,在预处理阶段对地形块数据进行分块编号,并进一步计算其包围球结构以及相关参数;在实时绘制阶段,根据视点在每一帧的位置建立基于四叉树和包围球结构的屏幕误差的动态 LOD 模型,具体的建立 LOD 的过程可以描述如下:

(1) 对满足 RSG 空间划分的地形数据进行分块编号,并根据其端点坐标构造地形块对应的包围球,存储其球心、半径以及中心节点的几何误差,转到第(2)步;

(2) 依据基于投影和包围球结构的视锥裁剪方法,判断当前视点的可见地形块,转到第(3)步;

(3) 查找可见地形块是否在内存中,若不在内存,则将其调入内存环境;若地形块已在内存,转到第(4)步;

(4) 对可见地形块分别进行孤立分割,并对其建立四叉树结构(将可见地形块整块作为四叉树的根节点),读取第(1)步中的计算数据,转到第(5)步;

(5) 从地形块的四叉树根节点,依次判断是否需要细分;具体的细分操作为:判断视点在该帧的位置与节点包围球的位置关系,若视点在节点包围球中,则该节点需要进一步细分;若视点不在节点包围球中,则根据式(13)判断节点是否需要细分。若满足式(13),则节点需要进一步细分;否则,对该节点的子节点进行第(5)步操作。若该节点没有子孩子,则转到第(6)步;

(6) 根据当前视点的位置,重新判断地形数据,转到第(2)步。至此,实现动态 LOD 模型的生成,进而完成大规模地形的实时绘制。

5 实验设计与分析

本文的实验程序是在 Windows 平台下,使用 VS2008 和 OpenGL 完成的。采用的硬件环境为 CPU (Intel(R) Core(TM) i7-3770 CPU @ 3.40 GHz), 显卡(AMD Radeon HD 7470), 实验中使用的数据是 Large Geometric Models Archive at Georgia Tech 网站提供的 The Puget Sound Area 的地形数据(包括 $1\,025 \times 1\,025$ 、 $4\,097 \times 4\,097$ 的高程数据和 $1\,024 \times 1\,024$ 、 $4\,096 \times 4\,096$ 的纹理数据)。设计以下两个实验来验证本文中所描述的算法。

实验 1: 不同屏幕误差阈值控制下的地形实时

绘制效果.

在基于屏幕误差的 LOD 算法中,屏幕误差直接影响了地形渲染的精确度和实时性,屏幕误差的阈值与渲染的精度成正比.因此,确定一个最优的屏幕误差对算法的效率有直接影响.实验选用 $1\,025\times 1\,025$ 的 RSG 数据,通过调整阈值 τ ,得到不同精度下的三维地形的显示,表 1 所示为不同阈值下的相关参数.由表 1 可见,随着阈值的减小,绘制的三角形数目比例会增大,显示效果较好;但单帧绘制时间增长使得运行效率有所下降.综合以上因素,选择适当的阈值对算法至关重要.实验表明,当阈值设定为 2~3 时,不会引起视觉上较大的跳跃感.

表 1 不同屏幕误差阈值下地形绘制的相关参数

地形范围	预处理时间/s	τ	绘制三角形比例/%	单帧绘制时间/s
$1\,025\times 1\,025$	9.251	4	0.41	0.017
		3	0.53	0.020
		2	0.80	0.024

实验 2:基于地形块包围球和屏幕误差的 LOD 算法的实现.

实验采用 $4\,097\times 4\,097$ 的 RSG 数据,通过不断的实验参数调整 and 比较,选择 257×257 为地形数据分块大小,选取屏幕误差阈值 τ 取值为 3 的情况下,使用四叉树的数据结构,并结合计算机内外存调度策略以及实时绘制过程中裂缝处理等经典算法完成连续 LOD 的生成,实现地形的实时绘制效果.图 5 给出了不同视点位置的同一地形的地形绘制效果,其中(a)、(b)中白色框内是同一区域,可见(a)中要比(b)的三角形数目多,LOD 层较细,实现了地形实时自适应表达.

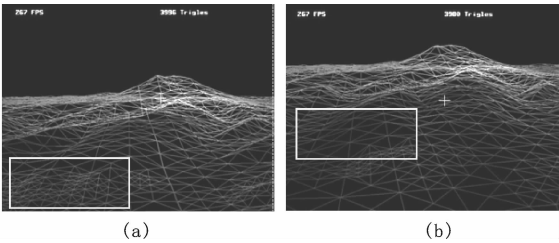


图 5 大规模地形实时三维表达

实验进一步对文献[5]、文献[7]的方法和本文设计的基于四叉树的包围球构造的算法进行了比较,当屏幕误差阈值取 3 时,漫游时每帧中地形场景包含的三角形面片的数目如图 6 所示.

图 7 为本文方法与文献[5]方法对同一区域的地形绘制比较,从图中可见虽然本文算法三角形数目减少,但依然较好的保持了地形的基本地貌特征.

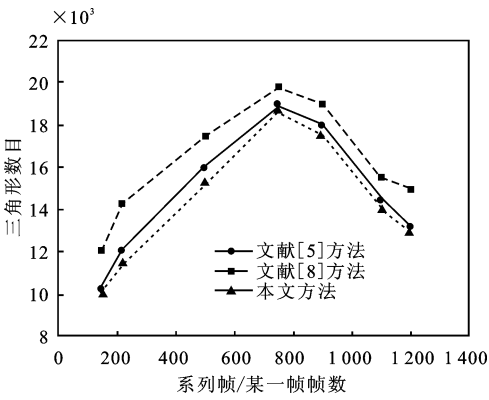
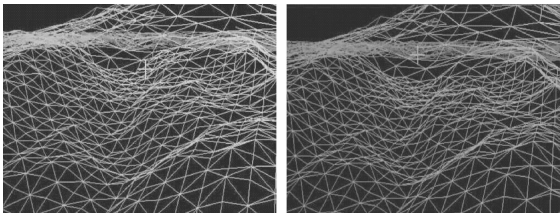


图 6 三角形数目对比



(a)本文方法
(三角形个数 4 532)

(b)文献[5]方法
(三角形个数 5 185)

图 7 阈值为 3 时某一区域显示效果

进一步对算法帧速进行了测试比较,本文算法在三角形数目不断增大时,平均帧速率并没有大幅度下降,在绘制三角形个数为 20 000 左右时平均帧速为 53 f/s,相比文献[5]中的 46 f/s 增加了 0.15%,说明本文的算法更好地满足了大规模地形绘制的实时性,并且可以较好地保持地形的基本地貌特征.

6 结束语

在大规模地形实时绘制时,动态生成多分辨率的 LOD 模型无疑是关键技术之一. 准确地实现地形的多分辨率 LOD,不仅能够提高计算机 GPU 渲染地形的速度,而且能够保证视觉上的进一步优化,减少层次切换带来的跳跃感. 本文提出的基于包围球和屏幕误差的 LOD 算法,在以下几方面做了改进:

- (1)根据 RSG 划分规则,对地形数据进行分块编号,在预处理阶段对地形块构造相应的包围球,完成相关参数的计算,不仅减少了空间上的存储量,还减少了实时绘制阶段的计算量;
- (2)建立基于地形分块和四叉树结构的屏幕误差较文献[5]中的基于三角形二叉树的屏幕误差,减少了实时渲染的计算量,保证了地形绘制的实时性和流畅性;

(3)视锥裁剪和屏幕误差计算两个过程都使用包围球结构,减少了地形绘制整个过程中的计算冗余,为渲染的实时性提供了进一步的可能。

实验结果表明,本文提出的 LOD 算法在屏幕阈值为 3 的情况下,实时性能良好,同时具有较高的帧速率,适用于大规模地形的实时绘制。在下一步的工作中,将结合视点运动的预测问题,进一步提高渲染的速度与地形的逼真度。

参考文献:

[1] Pfeifer N. A subdivision algorithm for smooth 3D terrain models [J]. ISPRS Journal of Photogrammetry and Remote Sensing, 2005, 59(3):115-127.

[2] 印桂生,陈怀友,张菁,等. 基于九宫格的累进 LOD 地形绘制算法[J]. 西南交通大学学报, 2010, 45(3): 411-417.

[3] 李钦,戴树岭,赵永嘉,等. 分块 LOD 大规模地形实时渲染算法[J]. 计算机辅助设计与图形学学报, 2013, 25(5).

[4] 张俊峰,姚志宏. 基于四叉树孤立分割和屏幕误差的地形 LOD 算法[J]. 西南交通大学学报, 2013, 48(4): 666-674.

[5] 杨莹,冯立颖,赵静,等. 基于分块和包围球误差函数的地形绘制方法[J]. 计算机工程, 2010,36(15): 199-201.

[6] 王超一,丁莹,张鑫,等. 一种基于投影的视锥体裁剪算法(英文)[J]. 系统仿真学报, 2014, 26(10): 2412-2417.

[7] Lindstorm P, Pascucci V. Terrain simplification simplified: A general framework for view-dependent out-of-core visualization [J]. IEEE Transactions on Visualization and Computer Graphics, 2002, 8(3): 239-2.

[8] Strugar F. Continuous distance-dependent level of detail for rendering heightmaps [J]. Journal of Graphics, GPU, & Game Tools, 2009,14(4):57-74.

作者简介:

王倩女, (1990-), 硕士研究生. 研究方向为虚拟现实.

高保禄 男, (1971-), 博士, 讲师. 研究方向为智能信息处理.

高锐军 男, (1990-), 硕士研究生. 研究方向为智能信息处理.

阴桂梅 女, (1975-), 硕士, 副教授. 研究方向为数据挖掘与机器学习.

高保禄(通讯作者) 男, (1971-), 博士, 讲师. 研究方向为智能信息处理. E-mail:85389301@qq.com.

(上接第 126 页)

从以上自修复流程还可以清晰的看出,相比于传统的双机热备系统,在与 EHW 技术相融合之后,电路模块出现故障之后,还能通过动态配置,实现电路功能的修复.从而提高了整个系统的可靠性.

5 结束语

本文在对常用的冗余容错技术分深入析的基础之上,依据系统可靠性和硬件资源消耗指标,提出了基于 EHW 和双机热备技术的故障自修复电路系统.设计了对应的电路系统,研究了对应的工作原理及故障自修复流程.对提高电路系统的可靠性具有重要的工程应用价值.

参考文献:

[1] 张峻宾,蔡金燕. 故障诊断与硬件演化的一体化设计[J] 微电子学与计算机, 2013, 31(2): 94-97.

[2] 张峻宾,万让鑫,蔡金燕,等. 基于硬件演化的电子电路故障自修复研究综述[J]. 计算机应用研究, 2013, 30(12): 3521-3524.

[3] Zhang JB, Cai JY, Meng YF, et al. Fault self-repair strategy based on evolvable hardware and reparation balance technology[J]. Chinese Journal of Aeronau-

tics, 2014, 27(5): 1211-1222.

[4] Gregory SH, Jason DL, Derek SL. Computer-Automated Evolution of an X-Band Antenna for NASA's Space Technology 5 Mission[J]. Evolutionary Computation, 2011, 19(1): 1-23.

[5] 姚睿,王友仁,于盛林,等. 基于进化硬件的自修复 TMR 系统设计及其可靠性分析[J]. 传感器与微系统, 2007, 26(8):72-76.

[6] 巨政权,郑见灵,满梦华,等. 三模冗余演化自修复系统可靠性及状态规律分析[J]. 电讯技术, 2012, 52(2): 245-249.

[7] 褚杰,原亮,赵强等. 基于 TMR-EHW 的高可靠性电机控制系统[J]. 计算机工程, 2009, 35(23): 10-12.

[8] 杨仕元. 数字系统的故障诊断与可靠性设计[M]. 北京:清华大学出版社, 1999.

作者简介:

张峻宾 男, (1987-), 博士研究生. 研究方向为电子电路故障自修复与硬件演化技术. E-mail:zjbsec@qq.com.

蔡金燕 女, (1961-), 教授, 博士生导师. 研究方向为检测理论、故障诊断、可靠性共性技术.

孟亚峰(通讯作者) 男, (1970-), 副教授、硕士生导师. 研究方向为电子装备状态检测、故障诊断.