

# 基于 ROAM 算法的实时地形可视化研究

陈希亮<sup>1</sup>, 曹 雷<sup>1</sup>, 崔 平<sup>2</sup>

(1. 解放军理工大学 指挥自动化学院, 江苏 南京 210007;

2. 中国人民解放军第 68242 部队, 宁夏 吴忠 751600)

**摘 要:**地形可视化是战场环境仿真的重要组成部分,文中在分析 ROAM 算法原理,运用网格简化和剔除算法提高渲染效率的基础上,对 ROAM 算法进行改进。首先对其高程数据源进行泛化,消除了 ROAM 算法对高程数据源的严格要求。然后根据 ROAM 算法自身的特点,提出了基于强制分割传播的地形块裂缝处理方法,通过在构造地形网格之前进行地形块拼接达到自动避免块间裂缝产生的目的。最后,通过实例对算法进行验证,实验结果表明改进后的算法能够实现算法的数据源无约束性,并且有效消除地形块间的裂缝,实现各地形块细节层次的一致性。

**关键词:**地形可视化;ROAM 算法;数据源泛化;裂缝消除

**中图分类号:**TP31

**文献标识码:**A

**文章编号:**1673-629X(2013)01-0237-05

doi:10.3969/j.issn.1673-629X.2013.01.058

## Research on Real-time Terrain Visualization Based on ROAM Algorithm

CHEN Xi-liang<sup>1</sup>, CAO Lei<sup>1</sup>, CUI Ping<sup>2</sup>

(1. Dept of Command Automation, PLA Univ. of Sci. & Tech., Nanjing 210007, China;

2. 68242 PLA Troops, Wuzhong 751600, China)

**Abstract:** Terrain visualization is important constituent of the battlefield environment simulation, based on the analysis of the ROAM algorithm, using mesh to simplify and eliminate algorithm improve the rendering efficiency, the ROAM algorithm was improved. First the elevation data source for generalization, eliminate the strict requirements of the ROAM algorithm to elevation data. And then according to ROAM algorithm with its own characteristics, proposed the terrain block crack processing method based on forced division propagation, achieve the purpose of avoiding the crack between blocks automatically through joining terrain blocks together before construction of terrain grid. Finally, an example was given to validate the algorithm, the experimental results show that the improved algorithm can realize the source of data for the algorithm without binding force, and effectively eliminate the crack between terrain blocks, realizing detail consistency of the terrain block level.

**Key words:** terrain visualization; ROAM arithmetic; generalizing the data source; seamless connection

## 0 引 言

地形可视化是在计算机中构建虚拟战场环境的重要研究内容之一,在构建虚拟战场环境的方法中,根据实际地形数据生成地形的应用最为广泛<sup>[1]</sup>。由于实际场景本身的复杂性,实现实时渲染的难度较大。因此,对实时地形渲染方法的研究主要从实际场景的几何特性入手,采用合适的方法以降低场景的复杂性。根据生成的网格是否规则,动态地形渲染算法可以分为两种:基于不规则三角形网格 TIN (Triangulated Ir-

regular Network) 和基于规则网格 RSG (Regular Square Grid) 的算法。基于 RSG 的算法依据高程数据源生成规整的三角形网格模型,代表算法有 Lindstrom 在 1996 年提出的连续 LOD 算法<sup>[2]</sup>、Duchaineau 在 1997 年提出的 ROAM 算法<sup>[3]</sup>和 Röttger 在 1998 年提出的受限二叉树算法<sup>[4]</sup>等;基于 TIN 的算法则从高程数据源顶点中取出满足重要性条件的顶点生成网格,以参考文献[5]为代表。基于 RSG 的算法虽然生成的网格中顶点和三角形数目比基于 TIN 的算法多,但是由于其易于裁剪和优化,因此效率更高。在众多基于 RSG 的地形可视化算法中,ROAM 算法能够实时、动态、高效地对地形模型进行划分,在地形可视化领域受到广泛关注。

ROAM (Real-time Optimally Adapting Meshes) 算

收稿日期:2012-05-07;修回日期:2012-08-10

基金项目:江苏省自然科学基金 (BK2011120)

作者简介:陈希亮 (1985-),男,湖北十堰人,助教,硕士,研究方向为计算机仿真、人工智能。

法是 Duchaineau 等人在 1997 年提出的,该算法通过对地形表面进行递归二分,构造出三角形二叉树,并以此为基础,对构造出的三角形进行分裂和合并操作以保证增加或删除顶点时所划分的网格的连续性。同 Lindstrom 的连续 LOD 算法相比,ROAM 算法在解决地形块之间存在的裂缝问题时更方便,效果更好。Lowa 大学仿真中心的 He 对基本的 ROAM 算法进行改进和扩展<sup>[6]</sup>,改进后的 ROAM 算法满足动态地形实时可视化的要求,基于此,提出了 DEXTER-ROAM 动态地形可视化算法,在对车辆驾驶模拟的可视化仿真中,DEXTER-ROAM 算法得到了比较好的应用。文中针对 ROAM 算法进行了深入的研究,针对 ROAM 算法在对数据源要求严格、存在裂缝等缺点的基础上提出了改进的办法。

## 1 ROAM 算法概述

ROAM 算法的基本思想是:以等腰直角三角形对地形网格进行逼近;在渲染地形时,根据视点的位置和地形的起伏程度,制定一定的误差度量标准评价地形的粗糙程度,而后根据制定的标准,对地形网格执行连续的基于三角形二叉分割的分割与合并操作,最终生成与实际地形相符合的无缝、连续的多分辨率地形表面。该算法以尽可能粗糙的网格表示地形,从而达到减少三角形数量、提高渲染效率的目的。作为一种动态 LOD,该算法能以高帧速提供具有数千三角形的高质量三角网,其简单和可扩展性使其成为目前地形渲染中广泛研究的算法<sup>[7-9]</sup>。

但 ROAM 算法渲染地形也存在一些不足:

1、ROAM 算法对地形数据要求非常严格,其地形高程数据必须表示的是一块正方形地形,且边长的高程数据点数必须为  $2^n + 1$  ( $n \in N$ )。由于 ROAM 算法是以等腰直角三角形作为地形网格的基本单元,因此地形网格中的三角形按照三角形二叉树合并后得到的也是等腰直角三角形,两个位于 0 层级的根等腰直角三角形构成的地形必然是正方形;同时,由于三角形在递归分割时始终进行二等分,因此地形正方形的边长的高程数据点数必须是  $2^n + 1$  (其中  $n$  为正整数)。在实际应用中,很多高程数据源并不能达到这样的要求。

2、ROAM 算法生成的地形中会出现明显的地形裂缝。

## 2 对 ROAM 算法性能的改进

针对 ROAM 算法两个方面的不足,一方面,提出覆盖采样的方法消除 ROAM 算法对高程数据源的严格限制,增强其通用性;另一方面,提出强制分割传递

的方法用于地形块无缝拼接,实现 ROAM 算法对大规模地形的支持。

### 2.1 数据源泛化

ROAM 算法要求地形高程数据必须表示一块正方形地形,且边长的高程数据点数必须为  $2^n + 1$  ( $n \in N$ )。实际应用中,很多高程数据源并不能达到这样的要求。为实现 ROAM 算法对数据源的无约束性,可以采用覆盖采样方法对算法进行改进。

为了更清楚地描述覆盖采样方法,做如下定义:

定义 1 标准高程数据源:数据源长宽方向的高程数据点数目相等,且为  $2^n + 1$  (其中  $n$  为正整数)的数据源。

定义 2 非标准高程数据源:数据源长宽方向的高程数据点数目不相等,或者一个或两个方向上的高程数据点数目不为  $2^n + 1$  ( $n \in N$ )的数据源。

定义 3 数据源坐标系:俯视数据源时,以数据源左上角高程数据点为坐标原点,以数据源横向向右为  $z$  轴正向,以数据源纵向向下为  $x$  轴正向, $y$  轴正向由  $x$  轴和  $z$  轴依据左手准则确定。

定义 4 采样坐标系:俯视采样网格时,以网格左上角采样点为坐标原点,以网格横向向右为  $z$  轴正向,以网格纵向向下为  $x$  轴正向, $y$  轴正向由  $x$  轴和  $z$  轴依据左手准则确定。

对非标准高程数据源,构造边长方向采样点数目为  $(2^n + 1)$  的正方形采样网格对高程数据源进行覆盖,使得采样网格的四个顶点与高程数据源的顶点重合。对于不能重合的采样点,通过插值方式求得对应的高程。这样就完成了对非标准高程数据源的覆盖采样。

设高程数据源沿  $x$ 、 $z$  方向顶点个数分别为  $p$ 、 $q$  ( $p, q \in N, p \neq q$ ),采样网格沿  $x$ 、 $z$  方向采样点数均为  $2^n + 1$  ( $n \in N$ )。构造采样网格的过程就是通过高程数据源求得各采样点高程值的过程,方法如下:

将顶点  $V_s(x_s, y_s, z_s)$  由采样坐标系映射为数据源坐标系中的顶点  $V_i(x_i, y_i, z_i)$ 。即将采样网格的  $2^n + 1$  个采样点等间距分布到高程数据源的  $p$ 、 $q$  个高程数据点中。

$$x_i = x_s \times \frac{p-1}{2^n}, y_i = y_s, z_i = z_s \times \frac{q-1}{2^n}$$

通过线性插值求出数据源坐标系中顶点  $V_i(x_i, y_i, z_i)$  对应的高程  $y_i$ <sup>[10]</sup>,如图 1 所示。

插值过程如下:

首先求出  $V_i$  在数据源中所处的正方形网格  $Square_i$  的四个顶点的高程,

$$\begin{aligned} x_u &= \lfloor x_i \rfloor, z_u = \lfloor z_i \rfloor \\ y_u &= \text{SrcVerts}[x_u \times \text{iZVertNum} + z_u] \end{aligned}$$

$$y_{rt} = \text{SrcVerts}[x_{lt} \times \text{iZVertNum} + z_{lt} + 1]$$

$$y_{lb} = \text{SrcVerts}[(x_{lt} + 1) \times \text{iZVertNum} + z_{lt}]$$

$$y_{rb} = \text{SrcVerts}[(x_{lt} + 1) \times \text{iZVertNum} + z_{lt} + 1]$$

其中,  $y_{lt}$ 、 $y_{rt}$ 、 $y_{lb}$ 、 $y_{rb}$  分别表示  $\text{Square}_i$  的左上、右上、左下、右下四个顶点的高程值;SrcVerts 采用行优先的方式存储了高程数据源中所有的高程点数据;iZVertNum 为高程数据源在  $z$  方向的数据点数目。

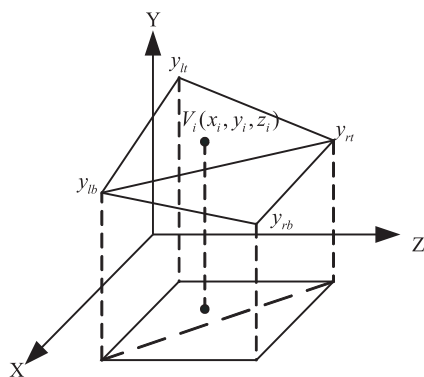


图1 插值方法求高程

然后根据顶点  $V_i$  在  $\text{Square}_i$  中的位置,对  $y_{lt}$ 、 $y_{rt}$ 、 $y_{lb}$ 、 $y_{rb}$  进行插值求出顶点  $V_i$  的高程  $y_i$ ,当  $V_i$  位于左上三角形时通过  $y_{lt}$ 、 $y_{rt}$ 、 $y_{lb}$  插值求解;当  $V_i$  位于右下三角形时通过  $y_{rt}$ 、 $y_{lb}$ 、 $y_{rb}$  插值求解。由于数据源网格间距为 1,因此定义  $\Delta x = x_i - x_{lt}$ ,  $\Delta z = z_i - z_{lt}$ ,如果  $\Delta x + \Delta z < 1$ ,则位于左上三角形:

$$y_i = y_{lt} + \text{Lerp}(0, y_{lb} - y_{lt}, \Delta x) + \text{Lerp}(0, y_{rt} - y_{lt}, \Delta z)$$

否则  $V_i$  位于右下三角形或对角线上:

$$y_i = y_{rb} + \text{Lerp}(0, y_{rt} - y_{rb}, 1 - \Delta x) + \text{Lerp}(0, y_{lb} - y_{rb}, \Delta z)$$

经过覆盖采样的采样点集合是符合 ROAM 算法要求的标准高程数据源,可以直接用于构建地形网格。

## 2.2 地形块裂缝消除

在 ROAM 算法中,不连续的分割在 patch 边界形成了裂缝。现有的地形块裂缝处理方法主要有三种:一是限制,即对相邻地形块的细节层次进行约束,要求其相差不能超过一个层次;二是网格变形,即对裂缝处的顶点进行移动,使裂缝处的顶点互相重合;三是修补,即在裂缝处添加新的三角形来填充裂缝<sup>[5]</sup>。这三种方式都是对网格模型进行后期处理,从而达到消除裂缝的目的<sup>[11]</sup>。

文中采用基于强制分割传播的地形块裂缝处理方法,通过预处理,自动避免地形块间裂缝的产生。该方法的基本思想是:在地形网格生成之前就将所有地形块相互拼接,绑定地形块中两个基三角形与相邻地形块中基三角形的邻居关系;在生成地形网格时,如果位于地形块边缘的三角形被分割,则通过其预先绑定的

邻居关系将强制分割操作传播到相邻地形块中,通过这种方式实现相邻地形块之间细节层次的一致性。

首先考虑的是地形块之间拼接关系的初始化。在 ROAM 算法中,一个地形块是由两个位于二叉树 0 层级的基三角形构成的,它们互为下邻居。在单块地形中,这两个三角形的左右邻居均为空,ROAM 算法的分割操作在地形块的边界处终止。为了在地形网格生成阶段能够将强制分割操作传播出去,必须在开始生成地形网格之前定义好地形块之间的关联关系,这种关联关系就是通过定义两个基三角形的左右邻居实现的。对由  $j * i$  块地形块构成的大规模地形,位于第  $i$  行第  $j$  列的地形块拼接关系初始化如图 2 所示。

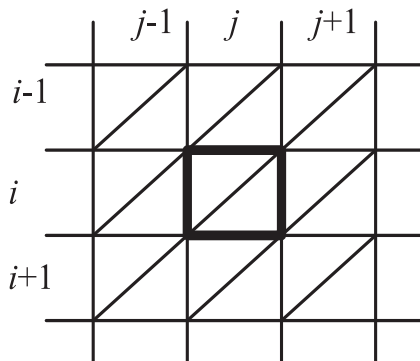


图2 地形块拼接关系初始化

地形块内部拼接关系定义:

$\text{patch}[i, j].\text{BaseLeft}.\text{BaseNeighbour} = \text{patch}[i, j].\text{BaseRight};$

$\text{patch}[i, j].\text{BaseRight}.\text{BaseNeighbour} = \text{patch}[i, j].\text{BaseLeft}$

地形块之间拼接关系定义:

$\text{patch}[i, j].\text{BaseLeft}.\text{LeftNeighbour} = \text{patch}[i, j - 1].\text{BaseRight};$

$\text{patch}[i, j].\text{BaseLeft}.\text{RightNeighbour} = \text{patch}[i - 1, j].\text{BaseRight};$

$\text{patch}[i, j].\text{BaseRight}.\text{LeftNeighbour} = \text{patch}[i, j + 1].\text{BaseLeft};$

$\text{patch}[i, j].\text{BaseRight}.\text{RightNeighbour} = \text{patch}[i + 1, j].\text{BaseLeft}$

其中 patch 是包含所有地形块的数组;地形块的 BaseLeft、BaseRight 属性分别定义了地形块的左、右基三角形;三角形的 LeftNeighbour、RightNeighbour、BaseNeighbour 属性分别定义了三角形的左邻居、右邻居和下邻居。对地形中所有地形块都执行以上拼接操作(位于整个地形边界处的地形块要进行适当取舍),就完成了强制分割传递的准备工作。

接下来考察地形块之间拼接关系的传递。强制分割操作只有在相互关联的地形块之间才能执行,因此在分割过程中保持地形块之间边缘三角形拼接关系的



传递是进行强制分割传递的前提。为了说明这种传递的可行性,考察一次地形块边缘三角形的分割过程,如图 3 所示。

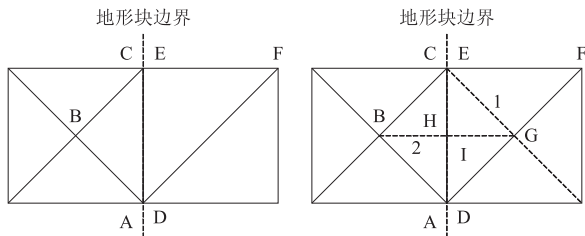


图 3 地形块边缘三角形分割

其中带有数字序号的虚线表示了分割操作进行的步骤;不同地形块中位于同一位置的顶点在逻辑上是重合的,但它们是由相互独立的顶点表示的,因为它们位于不同的地形块中。

假设左图中三角形  $ABC$  不满足误差范围需要进行分割,当前有:

$$\text{Triangle}_{ABC} \cdot \text{BaseNeighbour} = \text{Triangle}_{DEF}$$

分割过程分为两个阶段:

1 分割三角形  $DEF$ 。按照强制分割的要求,由于三角形  $ABC$  与三角形  $DEF$  不在同一层级,则先要对三角形  $DEF$  进行分割,如步骤 1 所示。分割后有:

$$\text{Triangle}_{ABC} \cdot \text{BaseNeighbour} = \text{Triangle}_{EGD}$$

2 对三角形  $ABC$  和三角形  $EGD$  构成的钻石结构进行分割,如步骤 2 所示。分割后有:

$$\text{Triangle}_{CHB} \cdot \text{LeftNeighbour} = \text{Triangle}_{GIE}$$

$$\text{Triangle}_{CHB} \cdot \text{RightNeighbour} = \text{Triangle}_{BHA}$$

$$\text{Triangle}_{BHA} \cdot \text{LeftNeighbour} = \text{Triangle}_{CHB}$$

$$\text{Triangle}_{BHA} \cdot \text{RightNeighbour} = \text{Triangle}_{DIG}$$

从分割结果可以看出,三角形  $ABC$  分割后生成的子三角形与相邻地形块中对应三角形分割生成的子三角形之间仍然保持正确的邻居关系,地形块边缘三角形之间的关联关系在三角形分割的过程中得到了继承。

综上所述,强制分割传递方法能够保证相邻地形块之间细节层次的一致性,从而避免产生地形块裂缝。与现有方法不同的是,该方法只需要在开始生成地形网格之前进行适当设置,在网格生成期间不需要进行额外处理,自动实现对相邻地形网格的动态切分。

### 3 实例分析

在 CPU 为 Intel(R) Core(TM) i5CPU 750,主频 2.67GHz,内存 4GB,显卡 ATI RADEON HD 5700 Pro,显存为 1024MB 的机器上,使用 C#语言,在 Managed DirectX 下实现文中改进的算法<sup>[12]</sup>。

采用任意像素的灰度图作为高程数据源,通过覆

盖采样方式每地形块取  $129 \times 129$  个顶点,构造  $4 \times 4$  块的地形,实现了地形块的无缝拼接和地形的正常显示,说明了覆盖采样方法及强制分割传递方法的可行性。不同的地形块以不同的纹理表示,渲染的地形如图 4 所示,图中可见的四块地形,从左到右,从远到近,数据源像素分别为  $54 \times 73$ 、 $126 \times 100$ 、 $120 \times 97$ 、 $137 \times 109$ 。

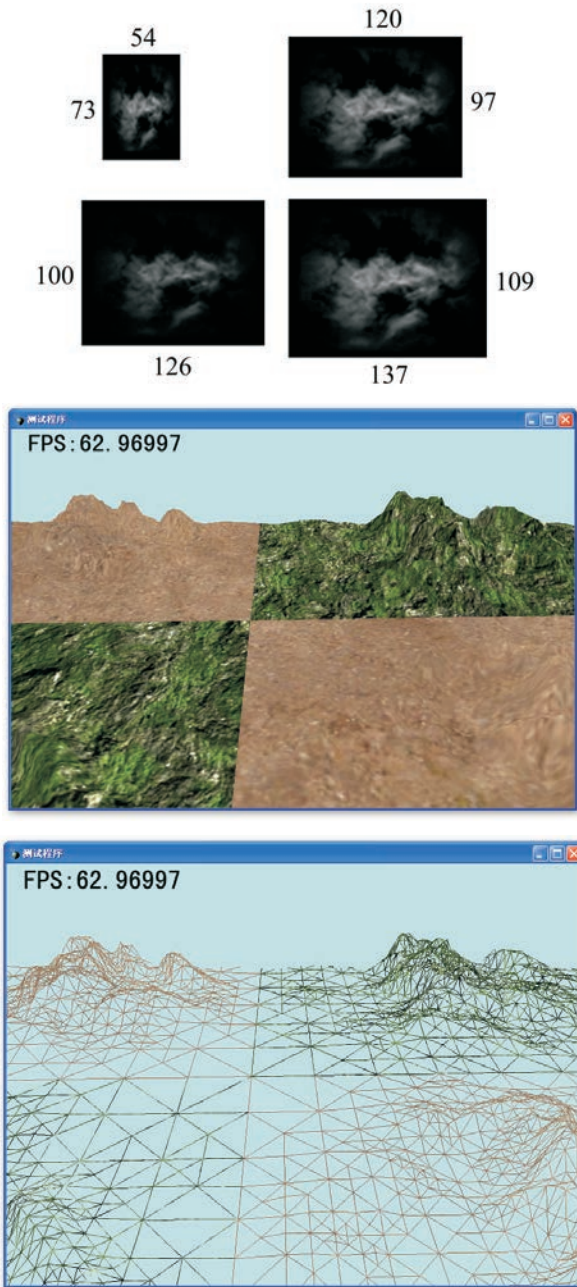


图 4 以任意分辨率灰度图为高程数据源实现的地形

由实验可见,覆盖采样方法能够对非标准数据源进行泛化,消除 ROAM 算法对高程数据源的限制,实现算法的数据源无约束性。

图 5、图 6 中生成的地形使用的是同一高程数据源,观察时摄像机位于相同的位置。从图 5 中可以看出,单独使用 ROAM 算法不进行地形块间裂缝消除处

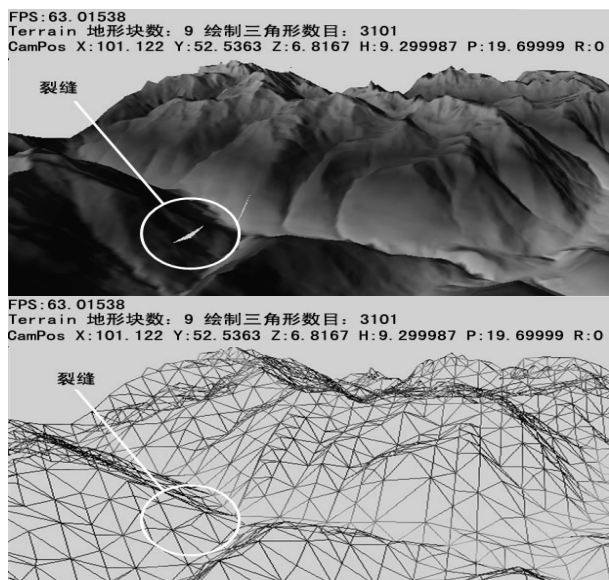


图 5 未经过裂缝消除的地形及其网格

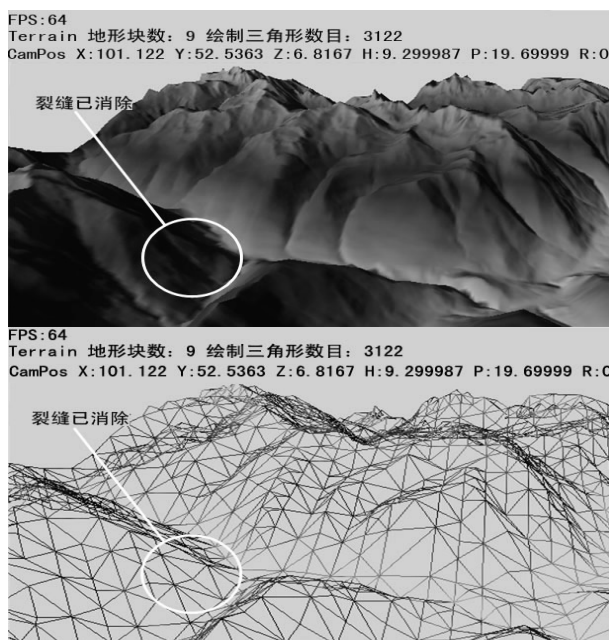


图 6 经过地形消除处理的地形及其网格

理,在生成的地形中会出现明显的地形裂缝。通过观察其网格发现,在裂缝产生处没有构成完整的钻石结构,相邻的两个三角形位于二叉树的不同层级上,从而导致了裂缝的产生。

图 6 显示了使用强制分割传播方法进行裂缝消除实现的地形。与图 5 进行对比可以发现,在图 5 中出现的裂缝已经消除。对裂缝产生处的两个相邻三角

形使用强制分割传播方法处理,实现了地形块的无缝拼接。因此,强制分割传播方法能够有效消除地形块间的裂缝,实现各地形块细节层次的一致性。

## 4 结束语

文中针对战场环境仿真中的三维地形可视化问题,从两个方面对 ROAM 算法进行了改进,提出了覆盖采样的方法实现数据源分辨率无约束性,使用基于强制分割传递的方法避免大规模地形不同分块之间的裂缝。最终,通过实例验证了算法改进的有效性。

### 参考文献:

- [1] 郭齐胜,董志明. 战场环境仿真[M]. 北京:国防工业出版社,2005.
- [2] Lindstrom P, Koller D, Ribarsky W, et al. Real-time, continuous level of detail rendering of height fields[C]//Proceedings of SIGGRAPH 96. New Orleans: [s. n. ], 1996:109-118.
- [3] Duchaineau M, Wolinsky M, Sigeti D E, et al. Roaming terrain: Real-time optimally adapting meshes[C]//IEEE Visualization 97. Phoenix: [s. n. ], 1997:81-88.
- [4] Röttger S, Heidrich W, Slasallek P, et al. Real-time generation of continuous levels of detail for height fields[C]//WSCG '98. [s. l. ]: [s. n. ], 1998.
- [5] Hopper H. Smooth view-dependant LOD control and its application to terrain rendering[C]//IEEE Visualization '98. [s. l. ]: [s. n. ], 1998.
- [6] He Y F. Real-time visualization of dynamic terrain for ground vehicle simulation[D]. Iowa: University of Iowa in America, 2000.
- [7] 涂 超. ROAM 算法原理及其应用研究[J]. 辽宁工程技术大学学报, 2003, 22(2): 176-179.
- [8] 魏 楠, 江 南. ROAM 算法及其在地形可视化中的应用[J]. 计算机工程与科学, 2007, 29(2): 66-68.
- [9] 徐 青. 地形三维可视化技术[M]. 北京:测绘出版社, 2000.
- [10] Luna F D. DirectX 9.0 3D 游戏开发编程基础[M]. 段 菲译. 北京:清华大学出版社, 2007.
- [11] 吴 晶, 徐晓刚, 王建国, 等. 大规模地形的快速漫游算法[J]. 计算机仿真, 2009, 26(2): 252-255.
- [12] 肖永亮, 耿卫东, 陈 为. 计算机游戏程序设计[M]. 北京:电子工业出版社, 2005:100-101.

# 基于 ROAM 算法的实时地形可视化研究

作者：

陈希亮，曹雷，崔平

作者单位：

陈希亮, 曹雷(解放军理工大学 指挥自动化学院, 江苏 南京 210007), 崔平(中国人民解放军第 68242 部队, 宁夏 吴忠 751600)

刊名：

计算机技术与发展

英文刊名：

Computer Technology and Development

年，卷(期)：

2013(1)

本文链接：[http://d.g.wanfangdata.com.cn/Periodical\\_wjtz201301060.aspx](http://d.g.wanfangdata.com.cn/Periodical_wjtz201301060.aspx)