

文章编号 1004-924X(2014)12-3419-08

## 基于注入式仿真平台的自适应真实感地形生成算法

王莹<sup>1,2,3\*</sup>, 王延杰<sup>1</sup>, 周渝人<sup>1</sup>, 李静宇<sup>1</sup>, 丁南南<sup>1</sup>, 姚志军<sup>1</sup>

(1. 中国科学院 长春光学精密机械与物理研究所, 吉林 长春 130033;

2. 中国科学院大学, 北京 100039; 3. 长春工业大学 电气与电子工程学院, 吉林 长春 130012)

**摘要:** 为了满足注入式仿真系统对实时性的要求, 提出了一种基于实时优化自适应网格 (ROAM) 算法的自适应真实感地形生成算法。将基本 ROAM 算法生成的地形作为基网格, 提出了一种新的评价参数, 通过比较给定阈值与参数来判断是否进行局部细分。对满足细分条件的顶点进一步提取地形的尖锐特征, 采用 Loop 算法细分局部曲面。在实时漫游过程中利用旋转映射技术生成一个视频流实时输出到注入式仿真平台并显示在数字视频接口 (DVI) 显示器上。实验结果表明: 该算法在实时漫游仿真时, 平均帧频达到 45.130 3 frame/s, 峰值帧频接近 50 frame/s, 高于原始算法的 36 frame/s。得到的仿真地形场景能够在注入式仿真平台上流畅输出到 DVI 显示器, 平均帧频为 45 frame/s 左右, 满足注入式仿真系统实时性的要求。

**关键词:** 计算机视觉; 自适应真实感地形生成算法; ROAM 算法; 地形提取; 注入式仿真平台

**中图分类号:** TP391.9; TP75 **文献标识码:** A **doi:** 10.3788/OPE.20142212.3419

### Real adaptive topography optimization algorithm based on injection simulation platform

WANG Ying<sup>1,2,3\*</sup>, WANG Yan-jie<sup>1</sup>, ZHOU Yu-ren<sup>1</sup>, LI Jing-yu<sup>1</sup>, DING Nan-nan<sup>1</sup>, YAO Zhi-jun<sup>1</sup>

(1. *Changchun Institute of Optics, Fine Mechanics and Physics,  
Chinese Academy of Sciences, Changchun 130033, China;*

2. *University of Chinese Academy of Sciences, Beijing 100039, China;*

3. *College of Electrical and Electronic Engineering,*

*Changchun University of Technology, Changchun 130012, China)*

\* *Corresponding author, E-mail: wangying2801@126.com*

**Abstract:** To meet the requirements of some injection simulation platforms for reality, a real adaptive topography optimization algorithm based on Real Time Optimally Adapting Mesh (ROAM) algorithm was proposed. By taking the terrain generated by ROAM as a base grid, a new integrated parameter  $Y$  was proposed, and a given threshold was compared with the  $Y$  parameter to determine whether the local splitting was implemented or not. The sharp features of the terrain were extracted from a vertex that satisfy the splitting conditions, and the local curve surface was splitted by Loop algorithm. Finally, a video stream was generated by rotation mapping algorithm in real-time roaming process and it was sent to the injection simulation system to

收稿日期: 2014-01-17; 修订日期: 2014-03-20.

基金项目: 国家 863 高技术研究发展计划资助项目 (No. 2012AA7031010B)

be displayed by a Digital Visual Interface(DVI) display. Experimental results show that the average of roaming frame rate of proposed algorithm is 45.1303 frame/s and the peak frame rate is close to 50 frame/s, higher than that of traditional algorithm. Moreover, the obtained terrain could display on the DVI display on the injection simulation platform with an average of roaming frame rate of 45 frame/s, which satisfies the requirements of the injection simulation systems for the reality.

**Key words:** computer vision; real adaptive topographic optimization algorithm; ROAM algorithm; terrain extraction; injection simulation platform

## 1 引言

仿真实验主要分为两种类型:半实物仿真和注入式仿真。其中半实物仿真花费较大,灵活性也差,注入式仿真则没有上述问题。目前注入式仿真主要应用于对红外图像的仿真,专门针对虚拟场景的注入式仿真的可查阅文献还比较少。

本文将上位机的地形场景生成一个视频流输入到注入式仿真平台,再通过 DVI(Digital Visual Interface)显示,形成了一套注入式仿真系统。

注入式仿真实验的实时性对虚拟场景的生成速度提出了很高的要求。虚拟场景生成效果和图像产生速度是衡量虚拟场景沉浸效果和程度的重要指标。对于数据量较小的虚拟仿真场景,可以同时兼顾上述 2 个指标,但是对于大规模虚拟仿真场景,其逼真度越高图像产生的速度就会越慢,因此要权衡考虑 2 个指标。图形的生成速度主要取决于图形处理的软硬件体系结构,特别是硬件加速器的图形处理能力以及图形生成所采用的各种加速技术。针对已知注入式仿真平台,本文通过改进图像处理算法来满足仿真平台的要求。

Duchaineau 于 1997 年提出了一种实时优化自适应网格<sup>[1]</sup>(Real-Time Optimally Adapting Meshes, ROAM)算法。它能够根据视点的位置和模型的起伏形状动态地计算模型的细节层次,减少每帧渲染多边形的数量,并具有较高的视觉质量。对于大规模场景,该算法由于评价体系太过精确使其渲染速度慢。陈希亮<sup>[12]</sup>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。许少华<sup>[13]</sup>提出了改进的分块算法,并通过细化边界增加地形的真实感。He<sup>[14]</sup>对 ROAM 算法进行了修改和扩展,使之适用于动态地形可视化的要求。该算法在 NADS (National Advanced Driving Simulator) 车辆驾驶

模拟器系统中得到了较好的应用,但不能针对大规模动态场景。

本文在先前工作的基础上,提出了一套针对大规模地形场景实时漫游系统下切实可行的地形生成算法。该算法利用原始 ROAM 算法低分裂次数生成基本地形框架作为虚拟仿真系统的初始地形网格;将综合地形参数与给定阈值进行比较,对满足一定条件的顶点进行 Loop 局部曲面细分;最后将实时漫游仿真场景生成一个视频流输出到注入式仿真平台并进行显示。

## 2 自适应真实感地形生成方法

### 2.1 基于 ROAM 算法的基础地形

ROAM 算法分为分裂和合并两个过程。基础理论为:记录三角划分中每个三角形的优先级,从基础三角划分开始,对最高优先级的三角形重复进行强制分裂。这个过程能够建立一个三角划分的次序,同时在每一步都能够最小化最大优先级队列。唯一需要的就是确保优先级是单调的,即孩子的优先级不大于它父亲的。图 1 为利用 ROAM 算法产生的地形网格框架。 $L_{\max}$  为三角形节点的最大分裂次数。

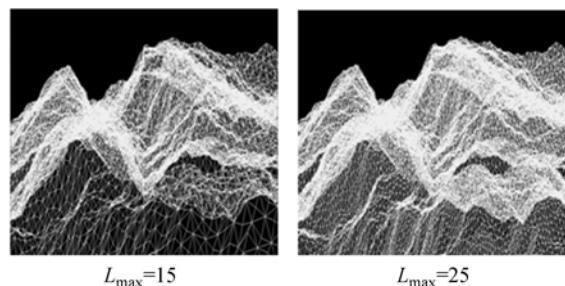


图 1 不同最大分裂层次的地形网格  
Fig. 1 Grids with different maximum division levels

ROAM 算法采用基于屏幕空间失真的方法作为节点的评价体系。它把三角形的几何误差转

换为图像空间上的像素误差:对于近处的物体,即使很小的几何误差转换到图像空间也会变成很大的像素误差。这个方法虽然很精确,但是渲染速度比较慢。针对大规模场景仿真,所需要的精确度没有那么多高,一般在米级左右。因此本文将 ROAM 算法的生成地形作为初始化基础地形,在实时漫游过程中通过计算  $Y$  值的大小来判定地形顶点是否进行 Loop 细分。实验表明本文算法漫游帧频高于原始算法。

## 2.2 地形总体特征的提取

引入地形参数  $Y$  来表示地形的总体特征属性,通过量化不同地形区域  $Y$  参数的大小来表示该地形区域的特征对地形生成算法的不同影响。对于  $Y$  参数的计算,这里主要考虑 2 个因素:视距和地形粗糙度。

### 2.2.1 视距

对于视点的相关判断,就像蜘蛛处于蜘蛛网中心往四周观测一样,视距越小,地形的细节越多,反之就越少。显然这种地形视觉特性和光的衰减的情况极为相似,因此可以借助光的衰减因子来表示 LOD 地形的视点相关因子。即:

$$L_{\beta} = \frac{1}{k_a d^2 + k_b d + k_c}, \quad (1)$$

其中: $d$  为视点与观察点之间的距离; $k_a$  为二次衰减因子,且  $k_a \geq 0$ ;  $k_b$  为一次衰减因子,且  $k_b \geq 0$ ;  $k_c$  为常数衰减因子。视点相关因子的取值范围介于  $(0, 1)$  之间。视距越小,视点相关因子越大,看到的细节就越多,反之亦然。但是本文因为考虑到算法的复杂度以及场景的实时漫游,故将视点相关因子简化为:

$$L_{\beta} = \frac{1}{k_b d + k_c}, \quad (2)$$

从而降低了计算的复杂度。

### 2.2.2 粗糙度

地形本身的粗糙度与地形的起伏有关,平坦地形的粗糙度较小,崎岖地形的粗糙度较大。图 2 中三角形分裂后中点  $v_c$  的粗糙度显然和该三角形节点的 3 个顶点的高度值有关。

设 3 个顶点和斜边中点  $v_c$  的高程值分别为  $h_i (i=0, 1, 2, 3)$ , 4 个顶点的最小高程值为  $m$ , 则它们的高程值分布  $H_i$  为:

$$m = \text{Min}(h_i) - 1 \quad i=0, 1, 2, 3$$

$$H_i = \frac{h_i - m}{\sum_i (h_i - m)} \quad i=0, 1, 2, 3 \quad (3)$$

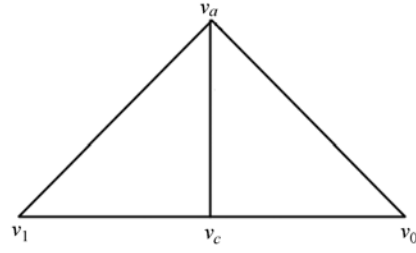


图 2 三角形顶点示意图

Fig. 2 Schematic diagram of triangle vertices

取最小值  $m$  时要减去 1, 这是为了防止 4 个点在同一个平面时公式(3)的分母为零。假设当前的位置为  $(x_0, y_0)$ , 引入一个熵值  $T(x_0, y_0)$  表示该点的粗糙度:

$$T(x_0, y_0) = - \sum_{i=0}^3 H_i \log_4 H_i \quad i=0, 1, 2, 3. \quad (4)$$

该地形点的熵值越小则地形起伏越大, 反之熵值越大地形就相对平坦。由于粗糙度的计算涉及到对数运算, 而  $H_i < 1$ , 通过泰勒级数展开, 舍去高次项, 则粗糙度可简化为:

$$T(x_0, y_0) \approx - \sum_{i=0}^3 H_i (H_i - 1) = 1 - \sum_{i=0}^3 H_i^2 \quad i=0, 1, 2, 3, \quad (5)$$

给出一个综合因子  $Y$ , 令

$$Y = \frac{T(x_0, y_0)}{L_{\beta}}, \quad (6)$$

用来衡量距视点距离  $d$  的顶点的综合细节程度。代入公式(2)和公式(5)得到:

$$Y = (k_b d + k_c) \left( 1 - \sum_{i=0}^3 H_i^2 \right). \quad (7)$$

由公式(7)可知,  $Y$  越小时得到的细节越多, 曲面越细腻。

一般来说, 视距是计算视点与所要观察的三角形平面的距离。由于三角形节点众多, 计算垂直于每个节点所在平面的单位法矢的计算量也比较大, 因此本文将视点与三角形分裂下一级斜边中点的距离作为视距。

## 2.3 基于参数阈值的局部细分

对于大规模地形场景的渲染除了常用的视锥体裁剪技术之外, 只需令距离视点较近的地形显示更多的细节即可。本文给定一个阈值  $\xi (\xi > 0)$ , 当  $Y < \xi$  时采用 Loop 算法细分曲面, 否则不做处

理。这里主要考虑 Loop 算法的细腻性。

Loop 细分是四倍三角形分裂细分模式<sup>[5]</sup>。细分曲面从控制网格  $M^r$  开始, 细分一次后由  $M^r$  变为  $M^{r+1}$ 。设  $v^0$  为控制网格的一个顶点, 其 1-邻域内的顶点(与顶点  $v^0$  紧密相邻的顶点  $v_1^0, v_2^0, v_3^0, v_4^0, \dots, v_k^0$ ) 的个数  $k$  成为该顶点的度。当  $k=6$  时为正则顶点, 否则为奇异顶点。 $M^{r+1}$  中的顶点  $v_0^{r+1}$  是由  $M^r$  中顶点  $v_1^r, v_2^r, v_3^r, v_4^r, \dots, v_k^r$  经过局部仿射而成。通过几次细分得到除了奇异点处  $C^1$  连续其余点处  $C^2$  连续的 4 次箱样条曲面。

首先对控制网格的边进行分类。在文献[10]中 Hoppe 将边分为需要保持尖锐特征的折痕边和非折痕边。这里将这两种边分别定义为 TypeEdge 取 1 和 0。令与顶点  $v^0$  相连的所有边的类型值的总和为  $n$ 。

(a) 当  $n=0$  时称为光滑点; 当  $n=1$  时称为尖刺点。光滑点和尖刺点的细分规则相同, 细分后新的内部顶点的位置为<sup>[9]</sup>:

$$v_0' = \frac{1}{8}((8 - k_b)v_0 + b v_1 + \dots + b v_k), \quad (8)$$

其中:  $b = \frac{1}{k}(5 - \frac{1}{8}(3 + 2\cos\frac{2\pi}{k})^2)$ 。

细分后新边点的位置为:

$$v_i' = \frac{1}{8}(3v_0 + v_{i-1} + 3v_i + v_{i+1}) \quad (i=1, 2, \dots, k). \quad (9)$$

(b) 当  $n=2$  时该点为边界点, 又分为如图 3 所示的边界顶点和边界边点, 当边界点的度  $k=4$  时为正则边界点, 其余的为奇异边界点。

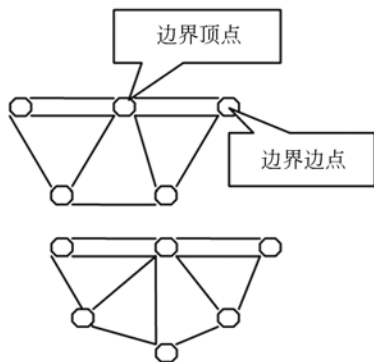


图 3 正则边界点和奇异边界点

Fig. 3 Regular boundary points and singular boundary points

(1) 边界顶点的细分规则:

$$v_0' = \frac{1}{8}(6v_0 + v_1 + v_k),$$

其中:  $v_1$  和  $v_k$  是和顶点  $v_0$  相连的 2 个边界顶点。

(2) 边界边点的细分规则。

当 2 个边界点都为正则边界点或都为奇异边界点时:

$$v_i' = \frac{1}{2}(v_0 + v_i),$$

当其中一个为正则边界点时:

$$v_i' = \frac{1}{8}(3v_0 + 5v_i).$$

权 5 给正则边界点。

### 3 基于注入式仿真平台的地形快速绘制

#### 3.1 注入式仿真硬件平台

注入式仿真系统主要包括三部分: 上位机, DVI 显示器, 注入式仿真平台。整体构架如图 4 所示。上位机主要用于产生虚拟地形场景。DVI 显示器显示注入式仿真平台输出的视频图像。注入式仿真平台主要包括 PCIe(Peripheral Component Interconnect Express)接口、CameraLink 输入和输出接口、DVI 输出接口、DDR3 SODIMM (Double Data Rate 3 DRAM Small Outline Dual In-line Memory Module) 接口以及这些接口的 FPGA(Field Programmable Gate Array)控制逻辑。仿真图像经过 PCIe 传输给 FPGA 缓存, 然后传送给 DVI 输出显示。如图 5 所示为注入式仿真平台的硬件实物图。

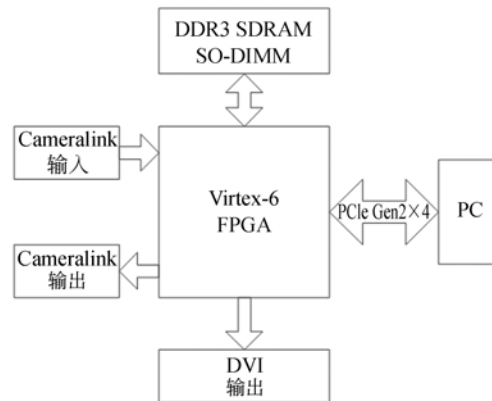


图 4 注入式仿真系统结构图

Fig. 4 Structure of injection simulation system

注入式仿真平台通过 FPGA 实现 PCIe 接

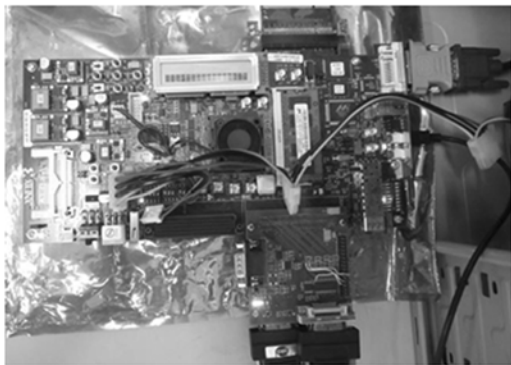


图 5 注入式仿真平台实物图

Fig. 5 Physical image of injection simulation platform

口,传输带宽高,独立测试条件下,端口最大速率约为 3.8 GB/s,最高带宽利用率约为 59.7%。因此对于这样一个高性能的仿真平台图像传输速率不是要考虑的主要问题,其瓶颈在于上位机发送图像的帧频。本文通过改进上位机地形场景生成算法提高实时漫游帧频来满足注入式仿真平台高实时性的要求。

### 3.2 算法步骤

根据上述真实感地形的生成方法,本文算法步骤设计如下:

1) 建立虚拟三维世界坐标,令  $z$  轴正方向为垂直于屏幕向里。取合适顶点坐标作为 ROAM 算法分裂起始顶点;

2) ROAM 算法细分网格,采用  $v_c[2] = 0.5f * (v_0[2] + v_2[2]) + \text{RAND}()$  来计算新顶点的高度,其中  $v_c$ 、 $v_0$ 、 $v_2$  分别为斜边中点以及三角形斜边的 2 个顶点,  $\text{level}2dsize[1] * \text{RAND}()$  为方差在一定范围内的随机数。连续分裂  $L_{\max}$  次,停止再分;

3) 在漫游过程中,计算距离视点  $d$  的地形区域顶点的总体特征,得到  $Y$  参数大小;

4) 通过比较  $Y$  参数与给定阈值  $\xi$  的大小,确定顶点是否细分;

5) 对满足细分条件的顶点进行 Loop 细分;

6) 绘制地形,通过旋转映射技术将仿真场景输出到注入式仿真平台显示。

在注入式仿真平台内,图像数据流走向的选择主要是通过将 DDR3 存储区划分为分别对应 CameraLink 输入和 PCIe 输入的两块来实现的。在 PCIe DMA 中定义一个数据流向选择寄存器,通过判断此寄存器的值来决定输出设备选择哪个

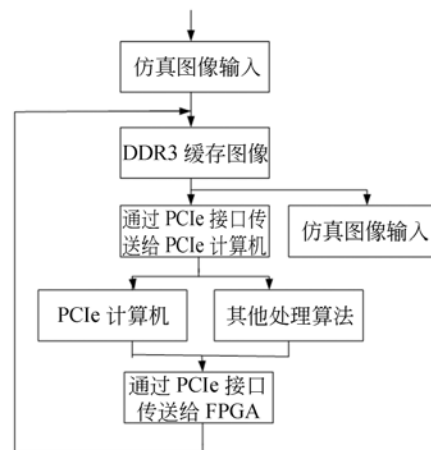


图 6 图像硬件处理流程

Fig. 6 Flowchart of image operating on injection simulation platform

输入源的图像数据输出。而数据流向选择寄存器的值可以在上位机中更改,这样就实现了上位机控制图像数据的流向,上位机可以控制图像输出的分辨率和帧频。图 6 所示为注入式仿真平台图像处理流程图。

## 4 虚拟场景的注入式仿真实验及结果分析

上位机实验环境为 Pentium(R) dual-Core CPU E5300 2.6G,内存为 2G,操作系统为 Microsoft Windows XP,编程环境为 VC++ 2010 调用 OpenGL。

对于  $Y$  值的选取本文做了多次实验,离散选取了 39 组  $Y$  值计算漫游时的平均帧频,绘制的曲线如图 7 所示,其中  $L_{\max} = 25$ 。发现当  $Y < 160$  时,局部细分顶点过多,计算时间过长;当  $Y > 165$  时,满足注入式仿真平台实时性要求;当  $Y > 175$  时,与原始算法相比,不仅漫游帧频高,而且得到的细节更加细腻;当  $Y > 210$  时,虽然实时漫游帧频高,但是细分的结果不是很明显。因此在本文实验中  $Y$  取 175~210 之间的值时,实时仿真效果好。本文实验中阈值  $\xi$  取 200。

图 8 为本文算法在  $L_{\max}$  分别为 15、25、45 时 Loop 细分一次前后对比图。由图 8 可以看出,本文算法在保持尖锐特征的同时细节更加平缓、细腻一些。

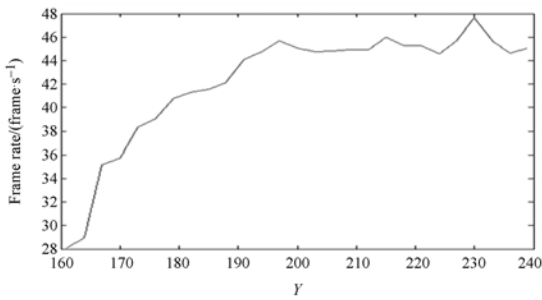
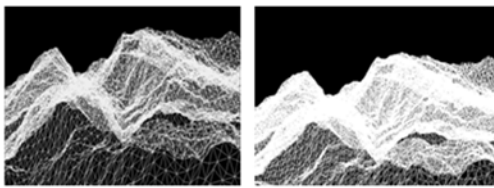
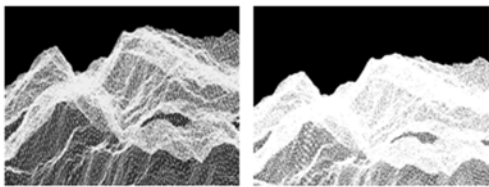


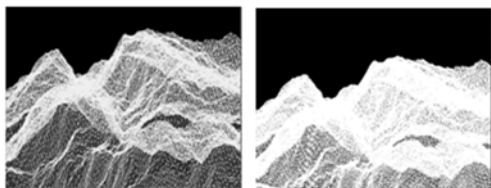
图 7 Y 值对帧频的影响曲线  
Fig. 7 Influence of Y on frame rate



(a)ROAM 分裂  $L_{max} = 15$  (b)本文算法  $L_{max} = 15$   
(a)ROAM algorithm with  $L_{max}$  of 15 (b)Proposed method with  $L_{max}$  of 15



(c)ROAM 分裂  $L_{max} = 25$  (d)本文算法  $L_{max} = 25$   
(c)ROAM algorithm with  $L_{max}$  of 25 (d)Proposed method with  $L_{max}$  of 25



(e)ROAM 分裂  $L_{max} = 45$  (f)本文算法  $L_{max} = 45$   
(e)ROAM algorithm with  $L_{max}$  of 45 (f)Proposed method with  $L_{max}$  of 45

图 8 Loop 细分对比图

Fig. 8 Comparison charts of Loop subdivision

由于地形漫游过程中随着视点的移动看到的地形细节不同,生成地形的基本三角形网格的个数也不同,为了计算数据的可靠性,让全部地形网格在 1 400 frame 内漫游完。每 100 frame 图

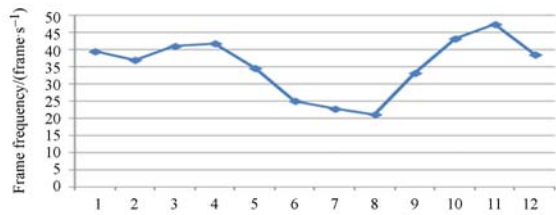


图 9 ROAM 算法  $L_{max} = 45$  时的帧频变化曲线  
Fig. 9 Change of frame frequency of ROAM algorithm ( $L_{max} = 45$ )

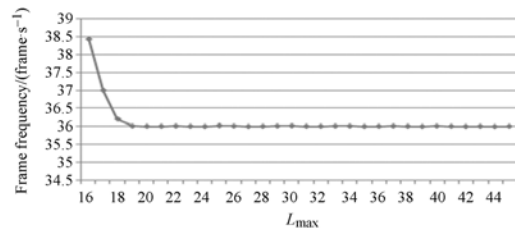


图 10 ROAM 算法随  $L_{max}$  的变化规律  
Fig. 10 Frame frequency of ROAM algorithm varies with different  $L_{max}$

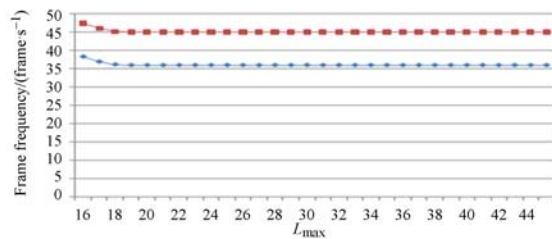


图 11 本文算法和 ROAM 算法随  $L_{max}$  不同帧频的变化规律  
Fig. 11 Frame frequency of the proposed algorithm and ROAM algorithm varies with different  $L_{max}$

像计算一次平均帧频,得到 12 组不同阶段的平均帧频数据,再将这 12 组数据的平均值作为当前  $L_{max}$  的帧频,图 9(彩图见期刊电子版)为  $L_{max} = 45$  时整个漫游过程中帧频的变化规律曲线。蓝色曲线表示第 1 组到第 12 组每 100 frame 图像对应的平均帧频的大小。图 10 为 ROAM 算法随  $L_{max}$  不同帧频的变化规律。计算 ROAM 算法当  $L_{max}$  为 15~45 的帧频的平均值为 36.187 frame/s。图 11(彩图见期刊电子版)为本文算法与 ROAM 算法帧频对比图,红色曲线代表本文算法  $\xi = 200$  时不同  $L_{max}$  对应的帧频变化曲线,计算 14 组帧频的平均值为 45.130 3 frame/s,蓝色为 ROAM 算

法帧频变化曲线。由图可知本文算法在大规模地形实时漫游过程中优于原始 ROAM 算法。

在数字图像注入式仿真测试时,由于将虚拟场景的注入代替传感器图像的生成,因此需要将虚拟场景实时绘制并注入到仿真平台。在每一帧图像细分处理完之后,将三维仿真场景映射到二维平面。视点坐标系与三维仿真“大地”坐标系的关系如图 12 所示,视点位置即观察者眼睛的位置。计算经过坐标轴  $y$  并与视点坐标系  $x_s$  轴所在直线垂直的平面(A)的方程表达式。

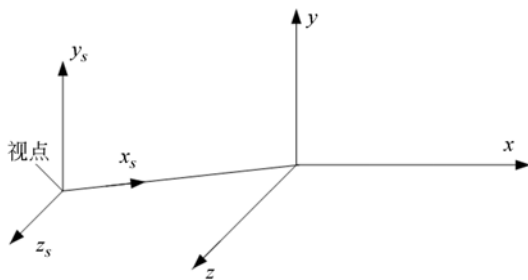


图 12 坐标系关系图

Fig. 12 Chart of coordinate relations

将视场内所有分裂完成并显示出来的地形顶点的坐标经过旋转映射<sup>[21]</sup>到平面 A 上,并快速检测顶点和三角形边是否存在重合现象,对于重合的三角形边界只需映射一次。将映射之后的地形顶点保存为分辨率  $256 \times 256$  的灰度 BMP 图像输出。这个过程耗时  $2 \sim 3$  ms,峰值达到 4.406 ms。

上位机经过 Camera Link 输出接口将图像输出到 PCIe 板卡,缓存到 DDR3 SDRAM 中,然后将缓存的虚拟仿真图像经过 PCIe 计算机,PCIe 计算机将虚拟仿真图像经过 PCIe 接口传送到 PCIe 板卡,PCIe 板卡将图像缓存到 DDR3 SDRAM 中,然后在将缓存的仿真图像经过 DVI 接口输出,最终显示到 DVI 显示器上。

如图 13 所示为本文算法仿真实验结果图,其中  $L_{\max} = 25$ 。右侧为上位机实时产生漫游仿真场景,左侧为 DVI 显示器显示注入式仿真平台输出的仿真场景。

#### 参考文献:

[1] DUCHAINEAU M, WOLINSKY M, SIGETI D, et al. . ROAMing terrain: real-time optimally adapting meshes[J]. *Proceedings of the 8th Conference on visualization*, 1997.



图 13 注入式仿真结果图

Fig. 13 Results of injection simulation experiment

在实时仿真过程中当  $L_{\max}$  取值为 15、25 时改变值,通过仿真平台在 DVI 显示器显示的仿真视频均为流畅输出,且上位机测试到平均帧频为  $43 \sim 46$  frame/s。由实验可知本文算法满足注入式仿真平台实时性仿真的要求。当  $L_{\max} = 45$ ,  $\xi$  取 200 时,漫游过程中偶尔会出现丢帧现象。初步判定为硬件仿真平台时序问题,具体原因需要进一步研究。这是课题组下一步要解决的主要问题。

## 5 结 论

为了满足注入式仿真平台对虚拟生成场景实时性的要求,提出了一种在基本 ROAM 算法生成网格的基础上对模型进行 Loop 细分的地形生成算法。对基网格的下一级节点进行 Y 参数的评估,将满足一定条件的顶点进行局部细分。实验表明本文算法不仅在生成图像的细节上更加细腻,而且在图像生成速度方面也高于原始算法。当  $\xi = 200$  时本文算法实时漫游帧频的平均值为 45.130 3 frame/s,高于原始算法的 36.187 frame/s。将实时仿真视频流输出到注入式仿真平台并显示,结果表明本文算法在  $L_{\max}$  分别为 15、25 时能够十分流畅地通过仿真平台输出到 DVI 显示器上,平均帧频在 45 frame/s 左右。实验结果表明本文算法优于原始算法,满足注入式仿真平台实时性仿真的要求。

[2] 王涛,汤浩,王金林,等. 一种人工数字地图的生成算法及其仿真研究[J]. *计算机仿真*, 2009, 26(6): 66-69.

WANG T, TANG H, WANG J L, et al. . An algorithm for artificial data map creation and its simulation [J]. *Computer Simulation*, 2009, 26(6): 66-69. (in Chinese)

- [3] HAGBI N, BERGIG O, EL-SANA A, *et al.*. Shape recognition and pose estimation for mobile augmented reality[J]. *8th IEEE International Symposium on Mixed and Augmented Reality*, 2009.
- [4] 周海, 周来水, 王占东, 等. 散乱数据点的细分曲面重建算法及实现[J]. *计算机辅助设计与图形学学报*, 2003, 15(10): 1287-1293.  
ZHOU H, ZHOU L SH, WANG ZH D, *et al.*. Surface reconstruction from unorganized points using subdivision techniques [J]. *Journal of Computer-aided Design & Computer Graphics*, 2003, 15(10): 1287-1293. (in Chinese)
- [5] LOOP C T. *Smooth subdivision surfaces based on triangles* [D]. M. S. Thesis, Department of Mathematics, University of Utah, 1987.
- [6] ENGEL K, KRAUS M, ERTL T. High-quality pre-integrated volume rendering using hardware-accelerated pixel shading [C]. *In Proceedings of the ACM SIGGRAPH/EUROGRAPHICS workshop on Graphics hardware*, 2001, 9-16.
- [7] 潘绍松, 左洪福. 工业增强现实中的相机跟踪[J]. *光学精密工程*, 2011, 19(6): 1353-1359.  
PAN SH S, ZUO H F. Camera tracking in industrial augmented reality[J]. *Opt. Precision Eng.*, 2011, 19(6): 1353-1359. (in Chinese)
- [8] HOPPE H, DEROSE T, DUCHAMPY T, *et al.*. Piecewise smooth surface reconstruction [C]. *Proceedings of the 21<sup>st</sup> annual Conference on Computer Graphics and Interactive Techniques*, 1994: 295-302.
- [9] 周海, 周来水, 王占东, 等. Loop 细分曲面的自适应等距面生成算法与实现[J]. *机械科学与技术*, 2003, 22(6): 1016-1020, 1029.  
ZHOU H, ZHOU L SH, WANG ZH D, *et al.*. An algorithm for generating offset Loop subdivision surface [J]. 2003, 22(6): 1016-1020, 1029. (in Chinese)
- [10] LIU Y G, PRABHAKARAN B B, GUO X H. Point-based manifold harmonics [J]. *IEEE Transactions on Visualization and Computer Graphics*, 2012, 18(10): 1693-1703.
- [11] YOO S W, SEONG J K, SEONG J K, SUNG M H. A triangulation-invariant method for anisotropic geodesic map computation on surface meshes [J]. *Visualization and Computer Graphics*, 2012, 18(10): 1664-1677.
- [12] 陈希亮, 曹雷, 崔平. 基于 ROAM 算法的实时地形可视化研究[J]. *计算机技术与发展*, 2013, 23(1): 237-241.  
CHEN X L, CAO L, CUI P. Research on real-time terrain visualization based on ROAM algorithm [J]. *Computer Technology and Development*, 2013, 23(1): 237-241. (in Chinese)
- [13] 许少华, 修春来. 改进的分块 ROAM 的地形简化生成算法[J]. *齐齐哈尔大学学报*, 2009, 25(3): 56-59.  
XU SH H, XIU CH L. Improved block ROAM terrain simplified algorithm [J]. *Journal of Qiqihar University*, 2009, 25(3): 56-59.
- [14] HE Y F. *Real-time visualization on dynamic terrain for ground vehicles simulation* [D]. Iowa: University of Iowa in America, 2000.
- [15] GIJSENIJ A, GEVERS T, VAN DE J. Computational color constancy: survey and experiments [J]. *IEEE Transactions on Image Processing*, 2011, 20(9): 2475-2489.
- [16] 卜雄洙, 李桂娟, 杨波, 等. 中心偏移的全景环形图像快速展开[J]. *光学精密工程*, 2012, 20(9): 2103-2109.  
BU X ZH, LI G J, YANG B, *et al.*. Fast unwrapping of panoramic annular image with center deviation [J]. *Opt. Precision Eng.*, 2012, 20(9): 2103-2109. (in Chinese)
- [17] HENNING BIERMANN, ADI LEVIN, DENIS ZORIN. Piecewise smooth subdivision surfaces with normal control [C]. *Proceedings of 27<sup>th</sup> Annual Conference on Computer Graphics and Interactive Techniques*, 2000: 113-120.
- [18] CHEN CH, ZHUANG Y T, NIE F, *et al.*. Learning a 3D human pose distance metric from geometric pose descriptor [J]. *Visualization and Computer Graphics*, 2012, 17(11): 1676-1689.
- [19] ELMQVIST N, DRAGICEVIC P, FEKETE J D. Rolling the dice: multidimensional visual exploration using scatterplot matrix navigation [J]. *IEEE Trans. Visualization and Computer Graphics*, 2008, 14(6): 1141-8.

#### 作者简介:



王莹(1985—),女,山东济宁人,博士研究生,2009年于哈尔滨理工大学获得学士学位,主要从事图像处理,视觉仿真方面的研究。E-mail: wangying2801@126.com



王延杰(1963—),男,吉林长春人,博士生导师,研究员,主要从事实时图像处理方面的研究。E-mail: wyjciomp@sohu.com