

文章编号:1001-893X(2011)06-0087-04

毫米波防撞雷达三维显示界面的设计与实现^{*}

靳碧鹏

(中国西南电子技术研究所,成都 610036)

摘要:提出了一种地形分块加速绘制算法。算法递归地将地形划分为多个大小相等的地形块,通过统计地形块中目标分布的稀疏度以及无目标地形块所占的比例,寻找最优的地形分块方式。此外,提出了一种利用 Diamond-Squares 算法生成随机纹理,并融合顶点颜色为地形着色的方法,渲染的地形真实感强、层次分明。最后,介绍了防撞雷达三维显示中的场景漫游方式。该方法已成功应用于实际工程中,显示效果良好,并且具有通用性,可以应用于一般的三维地形绘制。

关键词:防撞雷达;三维显示;地形绘制;纹理映射;环境漫游

中图分类号:TN957.7;TP317.4 **文献标识码:**A doi:10.3969/j.issn.1001-893x.2011.06.019

Design and Implementation of 3D Terminal Displays for Millimeter Wave Anti-Collision Radar

JIN Bi-peng

(Southwest China Institute of Electronic Technology, Chengdu 610036, China)

Abstract: This paper presents a terrain partition algorithm for quick rendering. The algorithm recursively divides the whole terrain into several equal blocks. By analysing the amount of target in a terrain block and the ratio of the blocks which contain no targets, it can find out an optimized way. Furthermore, a terrain coloring method is presented, which blends texture image generated by Diamond-Squares, and the vertex color. The terrain rendering by this method has a realistic effect and clearly color layers. Finally, the cruising method in 3D scene used for 3D displays of anti-collision radar is introduced. The method has been successfully deployed in engineering and has achieved a good effect. Otherwise, the method also adapts to general terrain rendering.

Key words: anti-collision radar; 3D terminal displays; terrain rendering; texture mapping; cruising in environment

1 引言

毫米波防撞雷达是针对直升机在恶劣气候条件下低空飞行时,无法清晰判断前方危险障碍物,进而影响飞机飞行安全的情况而设计的。这种障碍物防撞功能,要求雷达终端显示界面能够非常直观、准确地显示障碍物目标,以方便飞行员及时作出反应。目前,国内雷达的终端显示都是以二维方式为主,主要有 PPI 显示、B 型显示、C 型显示 3 种,但都有各自的缺陷:PPI 显示方式以极坐标的形式显示目标的

距离和方位,B 型显示以直角坐标的形式显示目标的距离和方位,两种方式都无法直观地表示目标高度;C 型显示方式能够清晰地显示目标的高度,却无法直观地显示目标的距离。在毫米波防撞雷达中,我们提出采用 3D(三维)方式显示雷达目标,这种方式不仅可以直观地显示目标的三维位置,还能真实地再现扫描空域内的实际场景。然而,这种方式在具体实现时,却面临一些问题:由于嵌入式图像处理板的硬件限制,图像绘制速度慢,无法达到系统的实时性要求;由于绘制效率的原因,无法考虑地形真实感方面的问题,形成的地形场景单一,无法反映不同

^{*} 收稿日期:2011-03-10;修回日期:2011-06-15

高度的地貌特征,或者没有添加阴影效果,绘制的地形立体感不强。在这种背景下,本文提出了地形分块加速绘制算法,以及随机纹理与顶点颜色相融合的地形着色方法。

2 地形分块加速算法

本文算法是一种规则网格地形绘制算法。目前基于规则网格的地形绘制算法在实际中应用较多的包括实时连续细节层次绘制(基于四叉树结构 LOD)^[1]、实时的最优自适应网格(ROAM)^[2]、基于 GeoMipMap 的绘制^[3]、嵌套网格算法^[4]。

本文设计的地形分块加速绘制算法源自基于四叉树结构的 LOD 算法思想,将地形分块,不同块采用不同的细节描绘。不同的是,LOD 算法是自顶向下递归细分地形,从而建立多分辨率的地形层次模型;本文算法仅仅是建立两层分辨率的地形模型,其过程是:首先将地形分成若干个大小相同的方块(除边界方块外),然后利用一定的准则判断本次分块是否合理,如果不合理,将地形进一步细分为更多的方块(地形块个数是上一次的 2 倍),如此递归,直到找出满足算法准则的分块方式。这种分块算法相比较四叉树结构的 LOD 算法实现更为简单,而且每次地形数据发生变化时,地形分块和地形渲染完全分离,大大提高了绘制速度。另外,本算法不需要满足四叉树结构 LOD 算法中顶点个数必须为 $2^n \times 2^n$ 的条件,适应于任意顶点个数的地形绘制。

2.1 分块算法描述

假设地形顶点的个数为 $M \times N$,分块后每个地形方块包含 K 行、 K 列个地形顶点。第一次分块时,设置 K 等于 M 、 N 的最大公约数,那么地形被等分为 $m \times n$ 个方块,其中, $m = \text{Round}(M/K)$, $n = \text{Round}(N/K)$ 。

(1) 设置 $m \times n$ 个方块的信息,存储到数组 $\text{Quad}[m][n]$ 中。地形方块的信息结构体定义如下:

```
Struct Quad
{
    Int    nRowStartIndex; //起始顶点的行索引号
    Int    nRowEndIndex;   //终止顶点的行索引号
    Int    nColStartIndex; //起始顶点的列索引号
    Int    nColEndIndex;   //终止顶点的列索引号
    Float  fRatio;          //障碍物顶点所占的比例
}
```

其中, $f\text{Ratio} = C/K \times K$, C 为地形方块中表示障碍物的顶点(顶点高度大于地面高度)个数, $f\text{Ratio}$ 表

示了地形方块中障碍物顶点所占的比例。

(2) 遍历数组 $\text{Quad}[m][n]$ 中的每个 $\text{Quad}[i][j]$,统计属性 $f\text{Ratio} = 0$ 的方块个数 Q ,计算其所占的比例 a :

$$a = Q / (m \times n)$$

式中, a 代表所有方块中无障碍物目标的方块所占的比例,如果 a 较大,说明划分后无障碍物目标的方块占的比例比较多,绘制时可以采用较少的细节描绘;如果 a 较小,说明无障碍物目标的方块数目占的比例较少,需要继续细分方块,降低细节层次。

计算属性 $f\text{Ratio} \neq 0$ 时, $f\text{Ratio}$ 的平均值 s :

$$s = \sum_{i=0}^{m \times n} f\text{Ratio}_i / (m \times n - Q)$$

式中, s 代表障碍物顶点在方块中所占的平均比例,如果 s 较大,说明方块中分布了较密集的障碍物顶点,无法再细分;如果 s 较小,说明方块中障碍物顶点分布非常稀疏,还可以继续细分,分离出无障碍物目标的方块。

(3) 依据 a 、 s 的值判断此次分块是否为最优分块。在试验中,我们寻找的最佳经验值为:如果 $a < 0.4$,或者 $a \geq 0.4$ 并且 $s < 0.2$ 时,需要继续细分地形,此时设置 $K = K/2$,依据顶点个数 K 重新划分地形,生成 $m \times n$ 个地形方块,转到步骤 1 进行新一轮的判断。

(4) 如果不满足步骤 3 的条件或者 $K = 1$ 时,说明此次分块是最优分块或者无法再细分,则停止分块。

2.2 网格渲染与裂缝修补

地形合理分块后,以块为单位绘制所有地形块。绘制时,判断地形方块的属性 $f\text{Ratio}$ 的值,如果 $f\text{Ratio} = 0$,说明地形块中没有障碍物目标,只需要采用较少的细节描绘,只需绘制方块的 4 个顶点构成的两个三角形面,如图 1 中的 C2、C3,图中圆形顶点为地形顶点;如果 $f\text{Ratio} \neq 0$,说明地形块中有障碍物,需要保持地形的细节,因此需要绘制方块内所有顶点构成的三角形面,如图 2 中的 C1,三角形个数较 C2、C3 的多。

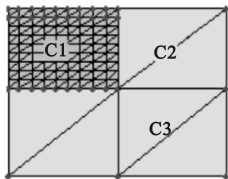


图 1 地形分块渲染示意图
Fig.1 Quad rendering in terrain

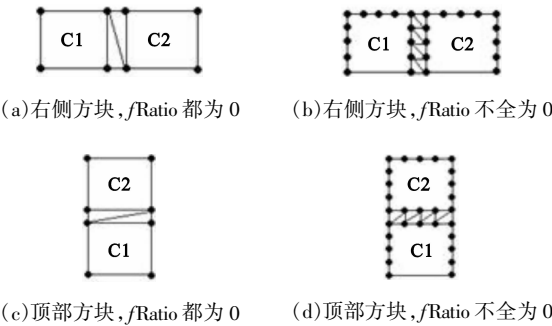


图 2 补画裂缝示意图

Fig.2 Four cases of mending a crack between quads

可以看出,分块加速绘制算法的核心是尽量分离出不包含障碍物目标的地形方块,以减少绘制的三角形个数,提高绘制速度。

但是,由于一个顶点只属于一个地形块,以块为单位绘制地形,相邻两个方块之间会出现裂缝。为了消除裂缝,每个方块绘制结束后,必须补画它与右边相邻方块、上边相邻方块之间的裂缝。补画裂缝时,判断相邻两个方块的属性 $fRatio$ 是否都为 0,如果是,绘制两个方块之间相邻的 4 个顶点构成的两个三角形,如图 2(a)演示了 C1 与右边相邻方块 C2 补画裂缝的三角形面,图 2(c)演示了 C1 与上边相邻方块 C2 补画裂缝的三角形面;如果有一个方块的属性 $fRatio \neq 0$,那么绘制相邻两行或者两列上所有顶点构成的三角形,如图 2(b)、(d)所示,此时的三角形个数较多。

3 快速地形着色方法

在显示界面中,不但要表现地形的真实感,而且需要突出表现地形的高度层次,以提示障碍物的危险级别。多纹理混合绘制算法^[5]能够较好地反映地形的层次感和立体感,但是,这种算法在每次地形数据变化时,都需要重新将多种纹理混合成一张与地形匹配的纹理图像,CPU 占用时间长,系统实时性差。本文提出了利用 Diamond - Square 算法^[6]生成带有随机阴影的纹理图模拟地形表面凹凸不平的细节,再混合多种颜色表现地形的高度层次,这种混合着色的方法占用内存少,处理速度快,能够很好地满足系统的实时性要求。

3.1 利用 Diamond - Square 算法生成纹理

(1)首先利用 Diamond - Square 算法生成地形的高程数据,算法描述如下:

1) 初始化二维数组 $n \times n$ ($n = 2^{m+1}$),并将 4 个

角设为相同高度。图 3 以一个 5×5 的数组为例,其中 4 个角放入初始高度值,用圆点表示,此为递归细化过程的起点;

2) Diamond 步:取 5×5 正方形的 4 个点构成的正方形,用二维随机中点位移法为其对角线交点生成一个随机高度值,作为此点的高度值,如图 3(b)所示;

3) 将 5×5 正方形细分成 4 个 3×3 正方形,如图 3(c)所示;

4) Square 步:取 3×3 正方形的棱锥中心和 4 个顶点的平均值,再加上一个与 Diamond 步相同的随机位移,可以计算出每条边的中心值,在计算位于边界上的点时,选用正方形另外一条边附近的对称点作为求平均值的点;

5)对上述过程进行迭代,直至达到所有的细分网格点被填充为止。

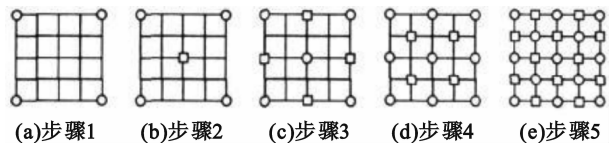


图 3 Diamond-Square 算法

Fig.3 The step of Diamond - Square algorithm

(2)将算法生成的高程数据绘制为三维地形,打开光照,增加阴影效果。

(3)读取帧缓存中的图像数据,此时,图像为真彩色图像,将其转化为灰度图,转换公式为

$$n_{Color} = 0.299 \times n_{Red} + 0.587 \times n_{Green} + 0.114 \times n_{Blue}$$

其中, n_{Red} 、 n_{Green} 、 n_{Blue} 分别代表红色分量、绿色分量、蓝色分量。

(4)将灰度图载入地形纹理内存,供纹理映射使用。

3.2 生成地形的颜色信息

对于某一个地形顶点 V_i ,设置颜色的过程如下。

(1)找出地形网格中与顶点 V_i 有边连接的 6 个顶点,比较 7 个顶点的高度,找出其中的最大高度值 H ,如图 4 所示,中间的方形顶点是顶点 V_i ,圆形顶点是与顶点 V_i 有边连接的 6 个顶点。

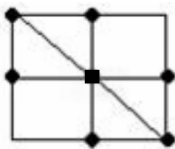


图 4 相邻顶点示意图

Fig.4 The neighbor vertex connect with a certain vertex

(2)依据高度 H 的大小,设置顶点 V_i 的颜色,分以下几种情况:如果 H 等于地面高度,顶点 V_i 的颜色为绿色,代表草地;如果 H 低于载机高度,顶点 V_i 的颜色为灰白色,代表一般障碍物;如果 H 高于载机高度,顶点 V_i 的颜色为黄色,代表危险地形。

渲染地形时,设置纹理映射方式为混合模式,为每个地形顶点指定纹理坐标和 RGB 颜色,混合渲染后的地形不但立体感强,而且地形层次非常清晰。由于着色方法使用的分形纹理是在初始化时生成的,因此程序在运行过程中只需要维护一组与高程数据相匹配的颜色数据,占用的内存少,计算量小,处理速度快,很好地满足了系统的速度要求。图 5 显示了利用雷达数据生成的三维地形效果图。

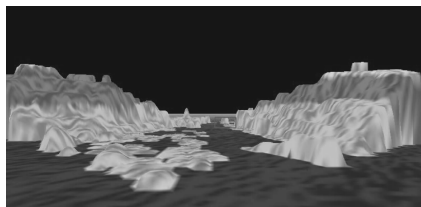


图 5 地形渲染效果图

Fig.5 The terrain rendered by colored method

4 场景漫游

系统中,每隔一帧产生一组新的地形数据,为了反映载机在真实场景中的飞行,场景视点的位置在一帧的时间间隔内需随载机速度移动,当新一帧的地形数据产生时,场景视点返回零点,然后重新开始随载机速度向前移动,直到新的地形数据再次产生。

视点在场景中移动的距离按如下公式计算:

$$D = V \times T / FACT$$

式中, D 是视点移动的距离, V 是载机速度, T 是一帧的时间间隔, $FACT$ 是三维场景与真实场景之间的缩小比例。

如果仅仅靠移动视点来显示载机的运动,每次帧切换、场景变化、视点返回零点时,视觉上有一种被强行拉回的感觉,这是因为在一帧时间间隔内,视点不断向前移动而地形纹理始终不变引起的。为了帧切换时也能有流畅的漫游效果,地形纹理需要随视点位置的变化而同步移动。软件中,纹理移动通过变换纹理坐标来实现,每次渲染时,累加视点移动的距离,计算其对应到纹理图上的偏移值,将该偏移值添加到每个纹理坐标上。这种纹理与视点同步移动的漫游方法,使得每次地形数据更新,场景交替时的画面非常流畅自然。

5 实验与结论

本文的三维终端显示软件在嵌入式图像处理板上运行,硬件配置为:CPU 为 Intel Core Duo 处理器 U2500,频率为 1.2 GHz,内存为 1 GB,显卡为 Intel 945 GM chipset,显存为 128 MB,采用 Visual C++ 6.0 和 OpenGL 编程。本系统中,三维地形的顶点个数固定为 300×400 ,对于图 5 所示的地形场景,未使用分块加速算法时,三角形个数为 238 602 个,绘制时间为 142 ms;采用分块加速算法后,绘制的三角形个数变为 72 736 个,绘制时间缩短为 31 ms。实验证明,本文采用的地形分块算法、着色方法、场景漫游完全满足了防撞雷达系统的性能需求,绘制速度快,场景交替自然流畅,能够真实显示载机飞行前方的地形场景,为飞行员提供直观可靠的信息。另外,本文提出的算法也同样适用于其它背景的三维地形绘制。

参考文献:

- [1] Lindstrom P, Koller D, Ribarsky W, et al. Real-time continuous level of detail rendering of height fields [C]//Proceedings of Annual Conference on Computer Graphics Series. New Orleans, LA: ACM, 1996: 109 - 118.
- [2] Bryan Turner. Real-Time Dynamic Level of Detail Terrain Rendering with ROAM [EB/OL]. (2000 - 04 - 03) [2006 - 10 - 12]. http://www.gamasutra.com/features/20000403/turner_01.htm.
- [3] de Boer Willem H. Fast Terrain Rendering Using Geometrical Mip-mapping [EB/OL]. (2001 - 02 - 12) [2006 - 10 - 12]. <http://www.flipcode.com/tutorials/geomipmaps.pdf>.
- [4] Losasso F, Hoppe H. Geometry Clipmaps: Terrain Rendering Using Nested Regular Grids [J]. ACM Transactions on Graphics, 2004, 23(3): 769 - 776.
- [5] 黄忠, 张佑生. 基于约束四叉树的真实感地形绘制 [J]. 合肥工业大学学报, 2008, 31(6): 854 - 857.
HUANG Zhong, ZHANG You - sheng. Realistic rendering of terrain based on restricted quadtree [J]. Journal of Hefei University of Technology (Natural Science), 2008, 31(6): 854 - 857. (in Chinese)
- [6] 王丽芳. 基于随机分形的地形场景的研究 [J]. 山西电子技术, 2007(5): 15 - 16.
WANG Li - fang. Research of Terrain Scenes Based on Random Fractal [J]. Shanxi Electronic Technology, 2007(5): 15 - 16. (in Chinese)

作者简介:

靳碧鹏(1981—),女,甘肃定西人,2007年获硕士学位,现为工程师,主要从事雷达终端显示领域的研究。

JIN Bi - peng was born in Dingxi, Gansu Province, in 1981. She received the M. S. degree in 2007. She is now an engineer. Her research concerns radar terminal display.

Email: ann08840907@163.com