代　号 10701　　学　号 0706321497
分类号 TP391　　密　级 公开

西安电子科技大学

硕士学位论文

题（中、英文）目 基于 OpenGL 三维分形地形的可视化研究

Research on Visualization of 3D Fractal Terrain Based on OpenGL

作者姓名 吴桐　　指导教师姓名、职务 璩柏青　教授

学科门类 工学　　学科、专业 机械制造及其自动化

提交论文日期 二〇一〇年一月

西安电子科技大学

学位论文独创性（或创新性）声明

秉承学校严谨的学分和优良的科学道德，本人声明所呈交的论文是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢中所罗列的内容以外，论文中不包含其他人已经发表或撰写过的研究成果；也不包含为获得西安电子科技大学或其它教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中做了明确的说明并表示了谢意。

申请学位论文与资料若有不实之处，本人承担一切的法律责任。

本人签名：吴桐　　日期 2010.3.10

西安电子科技大学

关于论文使用授权的说明

本人完全了解西安电子科技大学有关保留和使用学位论文的规定，即：研究生在校攻读学位期间论文工作的知识产权单位属西安电子科技大学。学校有权保留送交论文的复印件，允许查阅和借阅论文；学校可以公布论文的全部或部分内容，可以允许采用影印、缩印或其它复制手段保存论文。同时本人保证，毕业后结合学位论文研究课题再攥写的文章一律署名单位为西安电子科技大学。

（保密的论文在解密后遵守此规定）

本人签名：吴桐　　日期 2010.3.10

导师签名：　　日期 2010.3.13

摘要

三维真实感地形是可视化系统及虚拟仿真的基本组成部分，随着未来可视化和仿真技术快速发展，具有真实自然视觉效果的三维分形地形生成和建模技术显得越来越重要。本文着重研究三维分形地形模型的构建及其可视化系统的实现。主要工作如下：

针对当前三维真实感地形及其实时绘制技术的研究现状，在吸取三维计算机图形学、计算几何、科学计算可视化、虚拟现实等先进理论和技术成果的基础上，对三维地形真实感及其实时绘制技术中的分形地形模拟、三维真实感地形生成等核心技术内容展开讨论与研究。

在地形模型的建立过程中，对采用分形布朗运动模拟地形的方法进行了系统研究，给出了采用随机中点位移法中的“Diamond-Square”算法实现分形地形模拟的具体细节，并对模型进行了分析和优化。

利用OpenGL函数库在VC++平台上实现了本文的三维分形地形可视化系统，完成了场景中的地形、天空的构建与纹理渲染。

关键词：分形地形　分形布朗运动　中点位移法　可视化

Abstract

Realistic 3D terrain is one of the important components of virtual reality system and real time emulation. With the rapid development of virtual reality and emulation, the generation and modeling of high realistic 3D fractal terrain becomes more and more important. This thesis focuses on the establishment of 3D fractal terrain model and the achievement of the visualization system .The principal contents are as follows:

This paper focuses on the theoretical and methodological studies in the several subtopics：fractal terrain simulation，3D photo-realistic terrain rendering．These studies are based on some interisplines such as 3D computer graphics，computer geometry, scientific computation visualization, and virtual reality．

During the establishment of the model of terrain, it is systematically studied and carried out using fractal Brown motion method．For the fractal terrain simulation, the ‘diamond-square’ algorithm based on random midpoint displacement is given in detail，and the model is analyzed and optimized．

A 3D fractal terrain visualization system was achieved on the OpenGL and the VC++, with the construction and rendering of the terrain and the sky in the scene.

Keyword: **Fractal terrain** **fractal Brown motion** **midpoint displacement** **Visualization**

目录

第一章 绪论

1.1 三维地形可视化概述

地形与人类的生产生活息息相关，在城市规划、路径选取、资源调查与分配、工程勘查与设计、项目选址、环境监测、灾害预测与预报、军事、游戏娱乐等领域均有广泛的应用，因此人们一直关心如何真实的表达自然界的地形，以满足人类生活的需要。目前，地形的表达方式大致分为如下四种类[1]：

1．用符号、线条在介质上绘制地形，这是最原始的形式。标绘地形用的介质有绢丝、兽皮、布匹、纸张等，在上面用抽象的符号表示地物信息，用点、线条等几何形状表达基本的地形特征；用数字标注高程等信息，地图、高程图都属于这种类型。这种表达形式的缺点是缺乏真实感，不直观。

2．用沙盘把地形按比例尺缩小，人工建筑物、道路、山峰等都进行所谓处理。沙盘是真实地形的所谓景象，形象直观。沙盘至今仍有较广泛的应用，是表达地形的一种重要形式。

3．随着计算机、地图测绘、数据库、计算机图形学等理论和技术的发展，人们用计算机表达地形成为可能，电子地图应运而生，这是第三种地形的表达方式。电子地图是用矢量数据表达地理信息，比普通介质上符号线条地图有着非常大的进步，不但提供了更为强大的应用功能，如分层显示、定点、图层叠加、查找等等，而且更便于管理维护，例如数据查找，地图更新等等。尽管这种表达形式有了较大进步，但本质上还是用符号线条来表达地形，缺乏真实感。

4．随着计算机技术的进一步发展，计算能力的不断提高，使用计算机进行地形的三维表达成为目前研究的热点，这种地形的表达方式，不但感觉直观、真实性好，而且具有二维电子地图的其它优点，例如分层显示、位置顶点查找等。

可视化(Visualization)是为了支持用户的判断和理解，对人脑印象构造过程的一种仿真，具体的说，就是运用计算机图形学和图像处理技术，将计算机过程中及计算机结果的数据转换为图形以及图像在屏幕上显示出来，进行观察和模拟，并进行交互处理的理论、方法、和技术[2][3]。它涉及到计算机图形学、图像处理、计算机辅助设计、计算机视觉及人机交互技术等多个领域。

地形可视化的概念是在20世纪60年代以后随着地理信息系统(Geographical Information System，GIS)的出现而逐渐形成的。地理信息系统能够以电子地图的形式实现地理信息的可视化，并具有动态化和可交互的特点，大大方便了人们对地图的使用。

三维地形可视化技术是指在计算机上对数字地形模型(Digital Terrain Model，

DTM)中的地形数据进行逼真的三维显示、模拟仿真、简化、多分辨率表达和网络传输等内容的一种技术，它涉及到测绘学、现代数学、计算机三维图形学、计算几何、地理信息系统、虚拟现实、科学计算可视化、计算机网络等众多学科领域，在战场环境仿真、娱乐与游戏。地形漫游、道路选择、土地规划、网络GIS等众多领域有广泛应用[4] [5]。该技术在“数字地球”概念的大背景衬托下，显示出了强大的生命力和蓬勃生机，并随着与之相关的学科迅速发展而不断更新。因此，对其深入研究十分必要。

1.2 国内外研究现状

在计算机图形学领域，三维可视化是一个重要的研究方向，许多研究人员已经进行了大量卓有成效的研究，并有许多成熟的技术已经应用到实际中，出现了大量的优秀的可视化软件产品，如3D MAX、MAYA、EVS、AVS等。这些产品主要应用于游戏、电影动画、工业设计以及其它专业领域的研究，而与GIS联系较少。

可视化理论与技术用于地图学与GIS始于90年代初。1993年，国际地图学协会(ICA)在德国科隆召开的第16届学术讨论会上宣告成立可视化委员会 (Commission On Visualization),其主要任务是定期交流可视化技术在地图学领域中的发展状况和研究热点，并加强与计算机领域的协作。1996年该委员会与美国计算机协会图形学专业组(ACMSIGGAPH)进行了跨学科的协作，制订了一项称为“Carto Proiect"的行动计划，旨在探索计算机图形学领域的理论和技术如何有效地应用于空间数据可视化中，同时也探讨怎样从地图学的观点和方法来促进计算机图形学的发展[6]。1998年2月由B．H．Mccormick等根据美国国家科学基金会召开的“科学计算可视化研讨会"的内容撰写的一份报告中正式提出了“科学计算可视化(Visualization in Scientific Computing,简VISC)”的概念，从此标志着一门新的可视化学科的问世。

三维GIS研究主要集中在地形表面的重构、房屋建筑几何模型建立等方面。特别是在地形表达方面尤为突出。长期以来，人们针对不同的应用目的，依据各种数据模型、算法和数学理论，在现有的计算机发展水平上建立了许多地形可视化模型。目前，常见的地形可视化有两种类型：一是根据地学图形数据的精确描述，来进行真实地形的仿真；二是模拟自然场景中的地形，常用于具有真实自然视觉效果的虚拟环境中[7]。

在地形可视化建模方面大致可以分为如下三类：

(1)数据拟合生成三维地形

这是一种传统的地形生成方法，是利用常用的一些参数曲面，如Bezier曲面、Coons曲面、有理B样条曲面，通过插值、曲面拟合来生成所需要的三维地形。这种方法采用计算几何学建模，是早期三维地形生成的方法。由于其数学计算的复

杂性，对于复杂场景来说，计算量大而且要采用较复杂的曲面拼接技术。只适合中小规模的数据处理。另外，这种方法实际上是采用了欧式几何方法，而欧式几何所描述的物体具有光滑的表面和规则形状，物体的形状可由方程来描述。利用常用的参数曲面，通过插值、拟合来生成三维地形，也是采用方程来对地形建模。但由于地形的不规则和复杂性，用这种方法得到的地形真实感效果常不能令人满意。

(2)利用分形技术生成三维地形

1973年，曼德勃罗(B.B.Mandelbrot)在法兰西学院讲课时，首次提出了分维和分形几何的设想。分形几何学是一门以非规则几何形态为研究对象的几何学。由于不规则现象在自然界是普遍存在的，因此分形几何又称为描述大自然的几何学。欧式方法不能真实地描述这些物体，但可以用分形几何来真实地描述，是使用过程而不是方程来对物体建模。分形几何具有无限以及统计自相似性的规律，用递归算法使复杂的景物可用简单的规则来生成，可以生成任意水平的细节，为我们提供了一个很好的描述一般地面形状的数学模型。由于分形显示自然景物具有非常逼真的特点，自从分形技术产生以来，人们就开始探讨用分形技术来生成三维地形，地景生成技术也达到了一个新的阶段。采用分形技术来生成三维地形是目前地景生成的主要方法。

(3)基于数字地形模型的地形可视化

这种方法就是运用数字高程数据构造多边形面，用多边形网格逼近。数字高程模型是针对地球表面实际地形地貌的数字建模的结果。MilIer C.L于20世纪50年代中期提出了数字地形模型(Digital Terrain Model，DTM)的概念，后来把基于高程或海拔分布的数字地形模型称为数字高程模型(Digital Elevation Model,DEM)，DEM自20世纪50年代后期开始被采用以来，受到了极大的关注，在测绘、地质、景观建筑、农业、规划、军事工程、飞行器与战场仿真等诸多领域得到了广泛的应用。随着科学技术特别是计算机技术的迅速发展，在DEM的数据获取方法、数据存储和数据处理速度等方面取得了一些突破性的进展。现在，随着各种精度级别的DEM的普遍获取，过去许多潜在的应用领域现在已变成十分重要的方面。

在三维空间数据结构算法方面，杨必胜、李清泉、史文中提出了一种用于多分辨率三维模型快速生成和传输的稳健算法[8]；龚健雅提出了面向对象的矢量栅格集成数据模型[9]：还有邓念东，侯恩科提出了一种顾及维数的三维空间拓扑关系描述框架[10]；齐安文，吴立新等重点研究了基于三棱柱体体元在三维地质建模中的应用[11]；曹彤，李颖研究用于三维GIS的八叉树和四叉树算法[12][13]等；Klein采用一种与视点相关的TIN数据结构来表示交互中的集合信息，当视点改变时，采用Delaunav三角剖分法重构侧TIN；Luebke等提出了一种基于顶点数的简化算法，它可以对任意几何模型进行简化；Hoppe将他提出的渐进式网格模型也应用到地形当

中，并且提供了与视点相关的支持，为了避免三角剖分给全局带来影响，他在算法中将地形预先分成大小相等的若干块，在块内进行渐进式网格剖分。由于不能解决拼接问题，块与块没有简化，这在一定程度上影响了模型简化的效率[14]。

近年来，国内外在空间信息三维可视化方面的研究工作主要集中在以下两个方面：

1.运用动画技术制作动态地图，可用于涉及时空变化的现象或概念的可视性分析；

2.运用虚拟现实技术进行地形环境仿真，真实再现地景，进行交互观察和分析[15]。

1.3 本文研究内容

本文共分六章：绪论、OpenGL三维显示技术、DEM的获取、三维分形地形模型的建立、三维分形地形可视化系统设计与实现，最后为结论与展望。全文围绕三维分形地形可视化技术中的上述几个方面的内容，从理论到实践的流程进行表述。

第一章 绪论。介绍了三维地形可视化的概念和目前三维可视化技术的国内外研究现状，以及论文的结构与章节安排。

第二章 OpenGL三维显示技术。论述了三维图形变换，可见面识别，光照模型，纹理贴图几类三维图形现实的关键技术。

第三章 DEM的获取。介绍了DEM的表示方法，数据采集方法，以及数据采集原则。

第四章 三维分形地形模型的建立。综述了自然景物模拟的不同方法，随后详细论述了地形的基本特征，以及在此基础上基于分形布朗运动的三维分形地形模型的建立。

第五章 三维分形地形可视化系统设计与实现。介绍了本文利用 VC++与 OpenGL 实现的三维分形地形可视化系统，首先介绍其软件工具 OpenGL 与 Windows 编程，然后介绍系统的功能结构、系统流程，详细介绍了本系统各个功能模块的构建过程，最后说明效果图。

最后总结全文工作，并对未来的研究提出展望。

第二章 OpenGL 三维显示技术

三维图形就是为了使图形看起来更加真实。就像每个人眼睛平时所看到的一样或至少接近人眼所看到的内容，比如说自然光照射在物体上，物体反射光到人眼睛里，就产生图像。现实世界的物体都是三维的，通常用高度、宽度和深度来表示。而计算机的显示屏幕是二维的，在二维的屏幕上模拟真实物体的高度、宽度和深度所成的象，称之为三维图像。本章就让我们认识三维图像基本概念。

一般说来，用计算机在图形设备上生成真实感图形必须完成以下四个步骤：

(一)建模，即用一定的数学方法建立所需三维场景的几何描述，场景的几何描述直接影响图形的复杂性和图形绘制的计算耗费；

(二)将三维几何模型经过一定变换转为二维平面透视投影图；

(三)确定场景中所有可见面，运用隐藏面消隐算法将视域外或被遮挡住的不可见面消去；

(四)计算场景中可见面的颜色，即根据基于光学物理的光照模型计算可见面投射到观察者眼中的光亮度大小和颜色分量，并将它转换成适合图形设备的颜色值，从而确定投影画面上每一像素的颜色，最终生成图形。

要将一个物体三维地显示在屏幕上，需要经过三个步骤：

首先，要确定物体的世界坐标(x,y,z),建立物体的世界坐标数据库。

其次，模型被旋转或平移到新的位置。旋转和平移的结果产生了一系列观察坐标(x,y,z)。旋转是基于球坐标。

最后，被旋转和平移了的三维模型被投影(通过投影公式)到显示屏幕上，产生(x,y)平面坐标[16]。

2.1 空间位置和坐标

世界上的物体处在一个三维空间里，每个物体都有上、下、左、右、前、后。而世界上这样的两个物体就有了相对的位置，人眼与物体之间的相对位置，就是通常所说物体的空间位置。

在数学上，可以用3个有刻度的坐标来描述这个空间位置关系，这就是通常读者最熟悉的笛卡尔直角坐标系，如图2.1所示。

OpenGL也采用这样的坐标系进行工作，只是z方向相反。整个处理三维数据的过程都是在这样的坐标系下完成的。三维场景所处的空间称为世界，而代表这个空间的坐标系称为世界坐标系。观察者(视点)的位置、物体的位置都将通过该坐

标系进行描述，描述它们的坐标称为世界坐标或图形坐标。

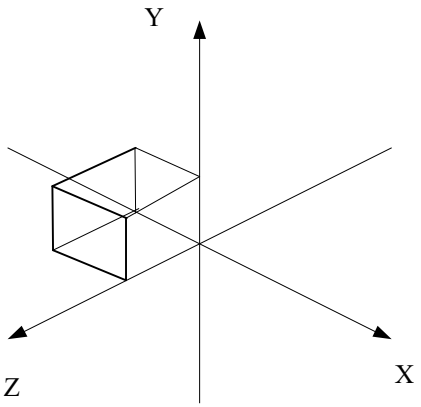


图 2.1 笛卡尔直角坐标系

2.2 三维图形变换

三维模型是在世界坐标系中建立的，但在计算机屏幕上所显示的景观画面是在给定的视点和视线方向下，三维地形场景投影到垂直于视线方向的二维成像平面(屏幕)上而形成的。将几何对象的三维坐标转换到其在屏幕上对应的像素位置，需要进行一系列坐标变换，一般统称为三维图形变换。计算机图形学中最基本的三维变换为几何变换、投影变换、视口变换。OpenGL不仅提供了三维变换函数，同时也提供了一系列矩阵操作函数。

2.2.1 几何变换

几何变换是指三维场景中的物体运动姿态的变化，包括物体的平移、旋转和缩放如图2.2所示。OpenGL提供实现平移、旋转和缩放的函数。

(1)平移变换

物体是相对原点定义与绘制的，所以移动坐标原点就相当于物体的平移运动。

OpenGL平移变换函数：

void glTranslate{d,f}(x,y,z)

功能是将物体从原点移动到(x,y,z)位置。即物体向左移动x，向上移动y，向前移动z。

(2)旋转变换

物体所在的坐标系发生旋转变换，因为物体相对于坐标系位置固定——由坐标定义的顶点位置固定，所以坐标系的变换将直接导致物体的变化。

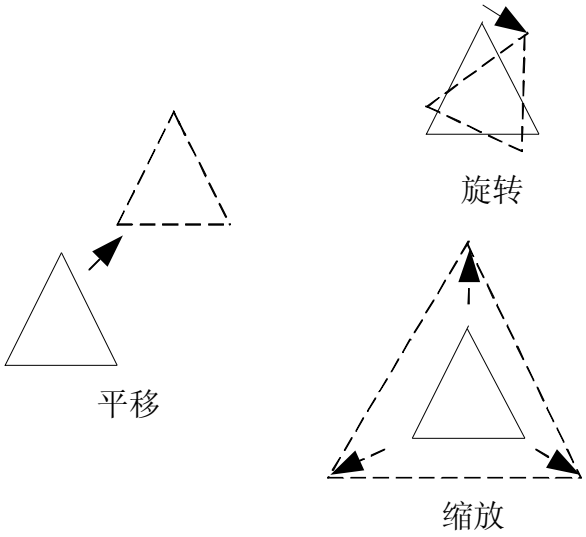


图2.2 几何变换

OpenGL旋转变换函数：

void glRotate{d,f}(angle,x,y,z)

旋转轴是局部坐标系的(0,0,0)指向(x,y,z)的射线，逆时针旋转angle(单位：度)

(3)缩放变换

物体缩放也是几何变换的重要组成部分，由于物体是相对原点和坐标系定义和绘制的，缩放坐标系即是坐标系中的物体缩放，在进行缩放时，你可以不同的轴线上指定不同的数值，这样就可以不规则地放大和缩小物体。

OpenGL缩放变换函数：

Void glScale{d,f}(a,b,c)

功能是将坐标系的x轴、y轴、z轴分别缩放a、b、c倍。坐标系中的物体将同步缩放。

2.2.2 投影变换

人眼观察事物是人眼的晶状体折射物体的光线，光线在视网膜上投影形成的影像。人眼有一定的视角，俗称视野，而视网膜是在晶状体后倒置的一个小的“显示屏”。在人眼之前放置一个显示屏，照射物体的光线投射到屏幕上成像的过程叫做投影。

从视点坐标系到屏幕显示器的二维平面坐标系的变换是投影变换。在计算机图形处理领域较多的采用透视投影。在OpenGL中，该变换过程是通过投影变换和视区变换来实现的。

一般有两种投影方式：平行投影(Orthographic Projection)和透视投影(Perspective Projection)。

(1)平行投影

OpenGL首先假定屏幕存在，由于透过屏幕观察物体与透过生活中的窗口观察物体不同，屏幕中的物体不可以相对观察者变化，所以OpenGL又为屏幕设定了一个观察位置，即距离屏幕一定距离的一个观察点A，改变视点A的位置，屏幕中的影像不发生变化，图2.3所示是一种平行投影的方式，B处为显示屏，“光线"由C处平行照射到屏幕上。

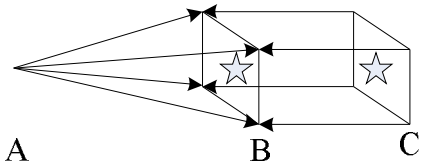


图 2.3 平行投影

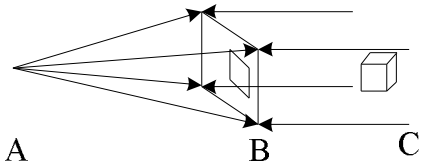


图 2.4 3D 空间到投影平面的平行投影

观察图2.3中的投影，屏幕只能显示屏幕所限制的四边形区域向后平行延展的四棱柱区域。再考虑到人眼的观察能力或计算机的计算能力，这个四棱柱不能无限延展，平面C就起到了限制作用。因此，OpenGL在定义投影方式时，同时也定义了视点A；一个在平行投影中作用不大的视距——A-B的距离；视场的左、右、上、下、前、后六个平面，并且六个平面围成一个矩形区域———视场。当然空间上的显示投影关系也定义出来了。

视场中的物体可以显示在屏幕上，而视场外的物体也可以绘制，但OpenGL并不将其显示出来，或者说，OpenGL将剪裁掉视场外的部分物体。

平行投影在CAD中广泛应用，其主视、俯视、侧视都是平行投影，但是此投影不能生成真实感效果，因为它丢失了深度信息，如图2.4，视场中一个立方体的平行投影结果，可以看出立方体显示成一个平行四边形，这是因为平行投影的原因。在平行投影中显示一个立方图形，需要将该图形转动一定的角度，使物体的各个边不重合投影在屏幕中。正是因为这个原因，在3D程序中主要使用透视投影。

OpenGL平行投影函数共有两种：

void glOrtho(

GLdouble left,:面向屏幕，相对读者，左边界。

GLdouble right,:面向屏幕，相对读者，右边界。

GLdouble bottom,:面向屏幕，相对读者，下边界。

GLdouble top,:面向屏幕，相对读者，上边界。

GLdouble near,:面向屏幕，相对读者，近边界。

GLdouble far,:面向屏幕，相对读者，远边界。

);

Void gluOrtho2D(

GLdouble left,:面向屏幕，相对读者，左边界。

GLdouble right,:面向屏幕，相对读者，右边界。

GLdouble bottom,:面向屏幕，相对读者，下边界。

GLdouble far,:面向屏幕，相对读者，远边界。

);

其中第二个函数为实用库函数，用于转化二维图形，因而缺省Gldouble top和Gldouble near。

(2)透视投影

OpenGL首先要假定屏幕存在，并为屏幕设定一个观察位置，即设一个观察点A。图2.5所示是一种透视投影的方式，B处为显示屏，“光线”由C处照射到屏幕上，即将影像显示在屏幕上。

进行透视投影时，空间中的物体所放射的所有光线都会聚到观察点。

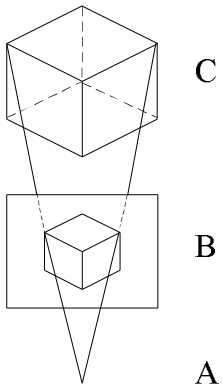


图2.5 透视投影

OpenGL透视投影函数共有两种：

void glFrustm(

GLdouble left,:面向屏幕，相对读者，左边界。

GLdouble right,:面向屏幕，相对读者，右边界。

GLdouble bottom,:面向屏幕，相对读者，下边界。

GLdouble top,:面向屏幕，相对读者，上边界。

GLdouble near,:面向屏幕，相对读者，近边界。

GLdouble far,:面向屏幕，相对读者，远边界。

);

void gluPerspective(

GLdouble left,:面向屏幕，相对读者，左边界。

GLdouble right,:面向屏幕，相对读者，右边界。

GLdouble bottom,:面向屏幕，相对读者，下边界。

GLdouble far,:面向屏幕，相对读者，远边界。

);

其中第二个函数为实用库函数。

2.2.3 视图变换

在计算机图形学中，视区变换的定义是将经过几何变换、投影变换后的物体显示于屏幕窗口内指定的区域内，这个区域通常为矩形，称为视图，如图2.6所示。

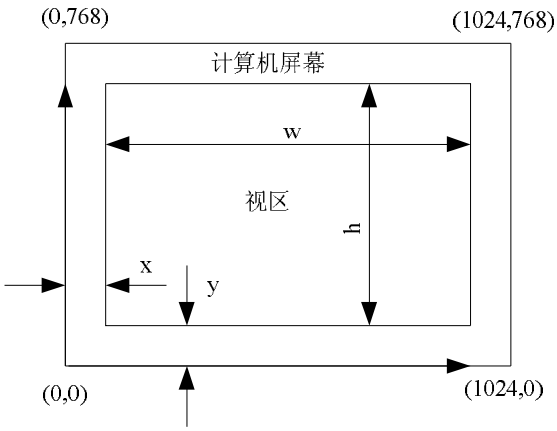


图 2.6 视图到窗口的映射

OpenGL视图变换函数：

void glViewPort(

Glint x,:面向屏幕，相对读者，左边距。

Glint y,:面向屏幕，相对读者，下边距。

GLsizei w,:面向屏幕，相对读者，视区宽度。

GLsizei h,:面向屏幕，相对读者，视区高度。

);

缺省时，参数值即(0,0,winWidth,winHeight)指的是屏幕窗口的实际尺寸大小。所有这些值都是以像素为单位，全为整型数。

2.2.4 OpenGL 矩阵操作

OpenGL中的变换都是依靠矩阵来完成所有的数学计算的，几何变换要用到模型观察矩阵，投影变换要用到投影矩阵。

在调用任何变换命令之前，必须指定是否想要去修改模型观察矩阵或者投影矩阵。对这两个矩阵的修改是通过OpenGL的glMatrixMode()功能函数来完成的。

OpenGL矩阵操作函数：

Void glMatrixMode(GLenum mode)

glMatrixMode参数包括GL_MODEVIEW、GL_PROJECTION、GL_COLOR、

GL_TEXTURE。进行变换时，需要调用glLoadIdentity()功能函数，它将单位矩阵转载作为当前的矩阵。实现代码如下：

```
glMatrixMode(GL_MODEVIEW);
glLoadIdentiy();      //重置模型观察矩阵
……                    //进行几何变换
glMatrixMode(GL_PROJECTION);
glLoadIdentiy();     //重置模型观察矩阵
……                    //进行投影变换
```

2.3 可见面识别

在三维空间中，一些物体遮挡另一些物体是常见的，而且这种遮挡关系随视点的不同而不同。为了保证物体显示的真实感，必须在显示立体视图时消去由于物体自身遮挡或相互遮挡而无法看见的线条和表面。清除一个物体被其它物体挡住的部分的操作称为消隐。OpenGL中消隐操作是由深度缓冲方法buffer(z-buffer)来实现的，深度buffer为窗口的每个点保留一个深度值，这个深度值记录了视点到占有该像素的目标的垂直距离，然后根据组成物体像素点的不同深度值，决定该点是否需要显示到屏幕上。

OpenGL深度检验函数：

Void glDepthFunc(GLenum fuunc);

该函数设置深度检验的比较函数，如果新片元的深度值与存储在深度缓存中的深度值满足指定的关系，则通过检验。表2.1列出了func参数的值。

实际操作中，需先调用glClear(GL_DEPTH_BUFFER)清除深度缓存，再启动深度测试，即调用函数glEnable(GL_DEPTH_TEST)，这样才能自动实现三维场景的消隐。取消自动实现三维场景的消隐要调用函数glDisable(GL_DEPTH_TEST)。

表2.1 深度检验func参数

参数	含义
GL_NEVER	无论结果如何，从不接受片元
GL_ALWAYS	无论结果如何，一直接受片元
GL_LESS	if depthnew<bepthold 接受新值
GL_LEQUAL	if depthnew<=bepthold 接受新值
GL_EQUAL	if depthnew=bepthold 接受新值
GL_GEQUAL	if depthnew>=bepthold 接受新值
GL_GREATER	if depthnew>bepthold 接受新值
GL_NOTEQUAL	if depthnew!=bepthold 接受新值

2.4 光照模型

三维景物图形的逼真性取决于能否成功地模拟浓淡或明暗效果，即用光照模型来计算可视面的亮度或色彩。浓淡处理并不能精确的模拟真实世界中光线和表面的性质，而只能逼近实际条件。一般逼真性越强，采用的光照模型就越复杂，计算量就越大。因此在涉及模型时，要兼顾精度和代价两方面的要求。

与现实世界相似，OpenGL通过将光近似地分解成红、绿和蓝色分量来计算光和光照。这就是说，一个光的颜色由此光中的红、绿和蓝色分量的数量所决定。当光照射到一个表面时，OpenGL根据其表面的材质来确定此表面应该反射的光的红、绿和蓝色分量的百分比。尽管这些都是近似的，但是OpenGL所使用的方程可以以相当快的速度计算出来。

OpenGL试图通过四种光的组合来模拟真实世界的光照：

(一)环境光。环境光看上去并不是来自任何特定的方向。即使存在一个光源，由于它强烈的散射性不可能确定其方向。被环境光所照射的表面将其向各个方向均匀地反射。

(二)散射光。散射光来自于一个确定的方向，但是它一旦遇到一个表面，就会被向各个方向均匀地反射。无论视点处于什么位置，此表面都显示出同样的亮度。

(三)镜面反射光。镜面反射光具有方向性，在表面的反射也有特定的方向。镜面反射光经常被称为亮光。

(四)反射光。带有反射光的物体看起来就好象其自身会发光，只不过这样的光不会对场景中的其他物体产生影响。在OpenGL中，反射光增加了物体的亮度，但是任何光源都不会影响反射光。

OpenGL允许在场景中最多同时使用八个光。我们真正需要的光不会超过八个，即使3D世界中存在多于八个光的数据，也应该使用某种数据操作程序将当前不可见的光剪裁掉。向场景中添加光照需要四个步骤[17]：

1)为每个物体的每个顶点计算法向量。法线确定了物体相对于光源的指向。

2)创建、选择并定位所有的光源。

3)创建并选择一种光照模型。光照模型定义了环境光，并设置用于光照计算的视点位置。

4)为场景中的物体定义材质属性。

2.4.1 光源创建

OpenGL光源创建函数：

void glLightfv(Glenum light,G1enum pname,TYPE*param)

此函数三个参数:1ight指定光的光源，pname指定光源的属性参数，param参数为pname指定了光源的属性值，在函数的向量形式下，它是一个指向数组的指针，非向量形式下是一个数值。只有在设定单值光源属性时才可以用到非向量形式。

表2.2 函数glLight*()参数pname说明

参数名	缺省值	意义
GL_AMBIENT	(0.0,0.0,0.0,1.0)	光源的环境光亮度
GL _DIFFUSE	(1.0,1.0,1.0,1.0)	光源的散射光亮度
GL_SPECULAR	(1.0,1.0,1.0,1.0)	光源的镜面反射光亮度
GL_POSITON	(0.0,0.0,1.0,0.0)	光源的位置(x,y,z,w)
GL_SPOT_DIRECTION	(0.0,0.0,-1.0)	聚光方向(x,y,z)
GL_SPOT_EXPONENT	0.0	聚光指数
GL_SPOT_CUTGFF	180.0	聚光终止角度
GL_CONSTANT_ATTENUATION	1.0	恒定衰减因子
GL_LINEAR_ATTENUATION	0.0	线性衰减因子
GL_QUADRATIC_ATTENUATION	0.0	二次衰减因子

注意：上以列出的GL_DIFFUSE和GL_SPECULAR的缺省值只能用于GL_LlGHT0，其他几个光源的GL_DIFFUSE和GL_SPECULAR缺省值为(0.0,0.0,0.0,1.0)。

光源的创建：

GLfloat light_position[]={1.0,1.0,1.0,0.0};

glLightfv(GL_LIGHTO,GLPOSITION,light_position);

其中light position是一个指针，指向定义的光源位置齐次坐标数组。其它几个光源特性都为缺省值。同样，我们也可用类似的方式定义光源的其他几个特性值，例如：

GLfloat light_ambient[]={0.0,0.0,0.0,1.0};//默认情况没有环境光

GLfloat light_diffuse[]={1.0,1.0,1.0,1.0};//可认为光的颜色

GLfloat light_specular[]={1.0,1.0,1.0,1.0);//物体的高光区的颜色基本上与物体

所反射的光线的颜色一致：

glLightfv(GL_LIGHT0,GL_AMBIENT,light_ambient);

glLightfv(GL_LIGHTO,GL_DIFFUSE,light_diffuse);

glLightfv(GL_LIGHT0,GL_SPECULAR,light_specular);

在OpenGL中，必须明确指出光照是否有效或无效。如果光照无效，则只是简单地将当前颜色映射到当前顶点上去，不进行法向、光源、材质等复杂计算，显示的图形就没有真实感。要使光照有效，首先得启动光照，即：

glEnable(GL_LIGHTx);

若使光照无效，则调用gDisable(GL_LIGHTING)关闭当前光照。然后，必须使所定义的每个光源有效，如glEnable(GL_LIGHT0)，其它光源类似，只是光源号不同而已。

2.4.2 法线的计算

法向量在OpenGL光照模型中是一个重要的概念，它是一个特殊类型的单位向量，它严密地代表了多边形所朝的方向。利用法向量，OpenGL能够计算出光线照射到物体上的角度。因此，若想产生立体感的物体，必须定义物体的法向量。

选取多边形上相邻的三点p1,p2和p3(三点不能在一条直线上)，求取向量v2(p3-p1)和v1(p2-p1)，并将其进行差积，便得到垂直于该三点所组成的多边形的法向量，如图2.7所示。

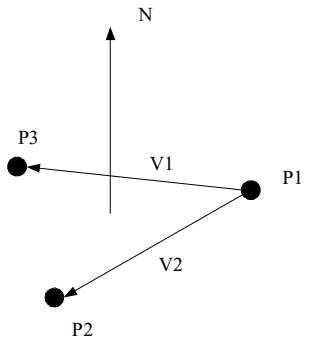


图 2.7 三角形平面法向计算

OpenGL法线函数：

void glNomal3f(GLfloat nx,GLfloat ny,GLfloat nz);

传递给此函数的三个参数是该表面法向量的x,y和z坐标分量。调用完此函数之后，紧接着就要调用相应的顶点函数。

2.5 纹理贴图

纹理映射技术也叫纹理贴图技术。在三维图形中，纹理映射的方法运用很广，尤其描述具有真实感的物体，如飞行仿真中常把一大片植被的图像映射到一些多边形上用以表示地面，或用大理石、木材、布匹等自然物质的图像作为纹理映射到多边形上表示相应的物体。

纹理分为两种，通过颜色色彩或明暗的变化体现出来的表面细节称为颜色纹理；另一类纹理则是由于不规则的细小凹凸造成的，例如，桔子皮的皱纹和未磨光的凹痕等。生成颜色纹理的一般方法是在一平面区域(即纹理空间)上预先定义纹理图案，然后建立物体表面的点与纹理空间的点之间的对应(即映射)。当物体表面的可见点确定之后，以纹理空间的对应点的值乘以亮度值，就可以把纹理图案附到物体的表面上。也可以用类似的方法给物体表面产生凸凹不平的凸包纹理。不过这时纹理值作用在法线向量上，而不是作用于颜色亮度。

纹理的定义有连续法和离散法两种。连续法把纹理定义为一个二元函数，函数的定义域就是纹理空间。离散法把纹理函数定义在一个二维数组中，代表纹理空间中行间隔和列间隔固定的一组网格点上的纹理值。网格点之间的其他点的纹理值则通过两格点的值插值获得。通过纹理空间与物体空间之间的坐标变换，把纹理贴图到物体表面。

纹理是数据的简单矩阵阵列。这些数据包括颜色数据、亮度(流明)数据或者颜色和alpha数据。矩形纹理还可以粘贴到非矩形区域上，这使得纹理映射的应用显得更加灵活，但纹理贴图的数学过程十分复杂。而OpenGL已对其进行了处理，使编程者可以利用几个函数就可以完成纹理贴图。执行纹理贴图的步骤可以概括为：定义纹理贴图，纹理过滤，说明纹理贴图方式，定义纹理坐标等。

2.5.1 定义纹理

在此主要采用离散法定义纹理。纹理可以是一维的，也可以是二维的。在图像处理中，通常采用的是二维纹理贴图。定义二维纹理贴图的函数是glTexImage2D()。

OpenGL二维纹理函数：

void TexImage2D(GLenum target,GLint level,GLint components，

GLsizei width,glsizei height,GLint border，

GLenum format,GLenum type,constGLvoid*pixels);

参数target是常数GL_TEXTURE_2D。参数level表示多级分辨率的纹理图像的

级数，若只有一种分辨率，则level设为0。

参数componems是一个从l到4的整数，指出选择了R、G、B、A中的哪些分量用于调整和混合，1表示选择了R分量，2表示选择了R和A两个分量， 3表示选择了R、G、B三个分量，4表示选择了R、G、B、A四个分量。

参数width和height 给出了纹理图像的长度和宽度，参数border为纹理边界宽度，它通常为0,width和height必须是2m+2b,这里m是整数，长和宽可以有不同的值，b是border的值。纹理映射的最大尺寸依赖于OpenGL,但它至少必须是使用64x64(若带边界为66x66),若width和height设置为0,则纹理映射有效地关闭。

参数format和type描述了纹理映射的格式和数据类型，参数format可以是GL_COLOR_INDEX、GL_RGB、GL_RGBA、GL_RED、GL_GREEN、GL_BLUE、GL_ALPHA 、 GL_LUMINANCE 或 GL_LUMINANCE_ALPHA(注 意 ： 不 能 用 GL_STENCIL_INDEX 和 GL_DEPTH_ COMPONENT) 。 类 似 地 ， 参 数 TYPE 是 GL_BYPE、GL_UNSIGNED_BYTE、GL_SHORT、GL_INT、GL_UNSIGNED_INT、GL_FLOAT或GL_BITMAP。

参数pixels包含了纹理图像数据，这个数据描述了纹理图像本身和它的边界。

2.5.2 纹理过滤

纹理映射到一个多边形上的过程就是从纹理图像空间映射到帧缓冲图像空间的一个过程。这种映射需要重新构造纹理图像，会造成应用到多边形上的图像失真。当一个纹理贴图被应用到一个变形的多边形上以后，如果视点接近纹理，一个单独的屏幕像素就可以表现纹素的一部分：如果视点远离纹理，一个像素可以表现一个纹素集合。纹理过滤就是用来在计算最终图像时告诉OpenGL如何将纹素映射成像素。

在纹理过滤中，放大就是指一个屏幕像素表现一个纹素的一小部分；而缩小就是指一个像素包含了一定数量的纹素。使用glTexparameter*()功能函数告诉OpenGL在适当的时候使用适当的过滤方式。

OpenGL纹理过滤函数：

void glTexParameterf(GLenum target,GLenum pname,TYPE param):

第一个参数可以是GL_TEXTURE_1D或GL_TEXTURE_2D，即表明所用的纹理是一维的还是二维的；第二个参数指定滤波方法，其中参数值GL_TEXTURE_MAG_FILTER 指 定 为 放 大 滤 波 方 法 ，GL_TEXTURE_MIN_FILTER指定为缩小滤波方法；第三个参数说明滤波方式，若选择GL_NEAREST,则采用坐标最靠近像素中心的纹素，这有可能使图像走样；若选择GL_LINEAR,则采用最靠近像素中心的四个像素的加权平均值。GL NEAREST

所需计算比GL_LINEAR要少，因而执行得更快，但GL_LINEAR提供了比较光滑的效果。

2.5.3 映射方式

在进行纹理贴图时，纹理图像可以直接贴到物体上从而覆盖物体已有的颜色，也可以和物体的原有颜色进行融合。OpenGL用于纹理贴图方式的函数为glTexEnv()。

OpenGL纹理映射方式函数：

void glTexEnv{if}[v](GLenum target,GLenum pname,TYPE param);设置纹理映射方式。参数target必须是GL_TEXTURE_ENV; 若参数pname是GL_TEXTURE_ENV_MODE,则参数param可以是GL_DECAL、GL_MODULATE或GL_BLEND,以说明纹理值怎样与原来表面颜色的处理方式：若参数pname是GL_TEXTURE_ENV _COLOR,则参数param是包含四个浮点数(分别是R、G、B、A分量)的数组，这些值只在采用GL_BLEND纹理函数时才有用。

2.5.4 纹理坐标

在OpenGL中，许多信息是通过顶点来传递的。顶点信息有几何坐标、顶点法向量、顶点颜色等等。与顶点有关的另一类重要的图像控制信息是顶点的纹理坐标。纹理坐标控制纹理图像中的纹素怎样映射到物体。纹理坐标可以是1、2、3、4维的，通常用齐次坐标来表示，即(s,t,r,q),如图2.8[18]所示。OpenGL定义纹理坐标的函数为gltexCoord()。

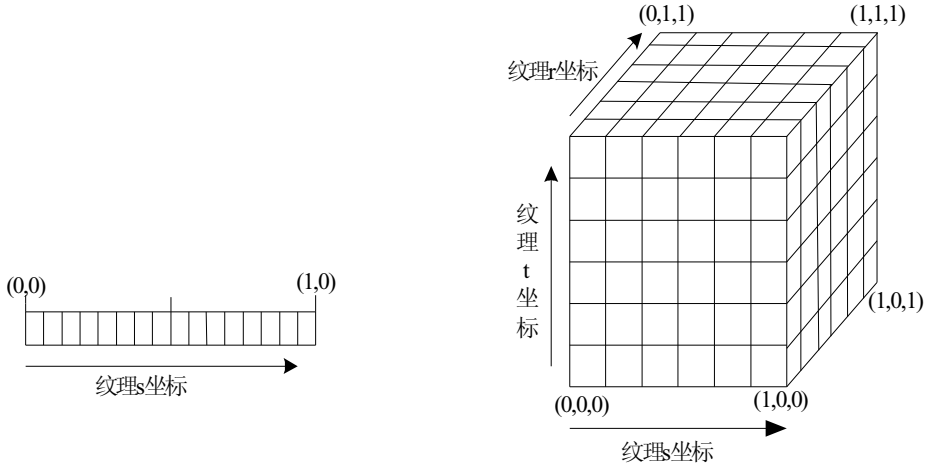


图 2.8 一、二、三维纹理坐标

OpenGL纹理坐标函数：

Void glteXCoord{1234}{sifd}[V](TYPE coords);

设置当前纹理坐标，然后调用glVertex*()所产生的顶点都赋予当前的纹理坐标。对于gltexCoord(),s坐标被设置成给定值，t和r设置为0，q设置为1；用gltexCoord2*()可以设置s和t坐标值，r设置为0，q设置为1；对于gltexCoord3*()，q设置为1，其它坐标按给定值设置；用gltexCoord4*()可以给定所有的坐标。使用适当的后缀(s,I,f或d)和TYPE的相应值(GLshort、GLint、glfloat或GLdouble)来说明坐标的类型。

将纹理贴到矩形上时，计算纹理坐标较简单，但是如果要将纹理贴到曲面上。就必须精确地计算纹理坐标。纹理坐标、纹理矩形大小以及被贴物体的表面形状将直接影响纹理贴图的效果。

最后，在绘制场景之前要激活纹理映射，激活或者取消纹理映射的函数是：glEnable() 和 glDisable(), 其参数可以是GL_TEXTURE_1D,GL_TEXTURE_2D，GL_TEXTURE_3D,分别代表一维、二维、三维的纹理图。

第三章 DEM 的获取

要在计算机上生成三维地形，首先须考虑的是地形数据的来源、结构和组织问题。自从1958年由美国麻省理工学院的Miller.C.L及其同事首次提出“数字地形模型(Digital Terrain Model,DTM)”概念以来，其理论和算法都已进行了很深入的研究，并得到广泛的应用。

3.1 地形高程的表示

数字地形模型(DTM)，是空间地形数据集合的统称，是带有空间位置特征和地形属性特征的数字描述。DTM中属性为高程的要素称为数字高程模型 (Digital Elevation Model,DEM),DEM是地表单元上的高程集合，通常用高程矩阵表示。DEM的应用非常广泛，其应用包括农、林、牧、水利、交通、军事以及测绘、制图、遥感等诸多领域，如公路、铁路、输电电线的选线、水利工程的选址、军事制高点的地形选择、土壤侵蚀、土地类型的分析等。广义的DEM可包括等高线、三角网等。从DEM可以派生出其它一些诸如平均高度、坡度。坡向等数据，以配合多种资源与环境的分析。地形表面高程变化的表示有很多种方法，可以归结为两类：基于数学定义的表面近似方法和基于点、线图形的表示方法，如图3.1所示。DEM的数学表达方法主要是借助连续的三维函数，利用函数生成的高度平滑的复杂曲面来近似整个地形表面；或者将复杂地形表面分解成正方形像元或面积大致相同的不规则小块，用这些局部拟合地形。DEM的图形表示则可分为如下两种模式：

1．线模式

表示地形的最普通的线模式是一系列描述高程曲线的等高线。由于现有的地图大多数都绘制有等高线，这些地图便是数字地形模型的现成数据源，通过扫描数字化，可得到数字等高线地图(Digital Contour Map,DCM)。根据数字等高线地图通过插值公式计算各点的高程值，得到DEM。

2．点模式

在点模式下，根据DEM生成方式的不同，主要有人工网格法、立体像对分析法和不规则三角网法。

(1)人工网格法

将地形图蒙上格网，逐格读取中心点或角点的高程值，构成数字高程模型。由于计算机中矩阵的处理比较方便，高程矩阵已成为DEM的最通用的形式。英国和美国都用较粗略的矩阵(美国用63.5m像元格网)从1：25万地形图上产生了全国的

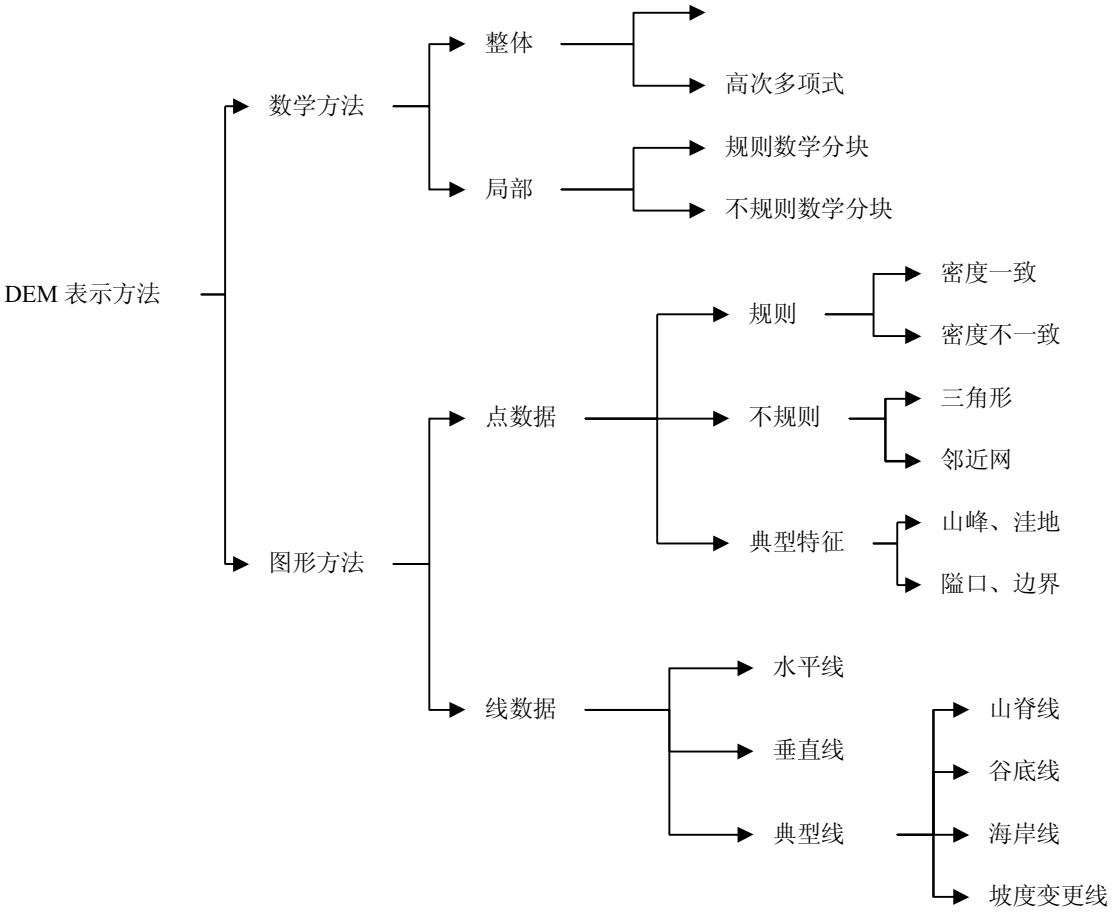


图3.1 DEM的表示方法

高程矩阵，以1:5万或1:2.5万比例尺地图和航片为基础的高分辨率高程矩阵正在英、美等国家扩大其使用范围。高程矩阵的优点在于便于计算等高线、坡度、坡向、山地阴影以及描绘流域轮廓等，然而由于采用的是规则格网表达，因而在地形简单的区域存在大量的冗余数据，而且为适应地形复杂程度不同的区域需要改变网格大小等。高程矩阵的另外一个缺点是由于栅格过于粗糙而不能精确表示地形的关键特征，如山峰、洼坑、隘口、山脊、山谷线等。

(2)立体像对分析

改进采样方法的实际应用很大程度上解决了采样过程中产生的数据冗余问题。改进采样方法通过对遥感立体像对，根据视差模型，自动选配左右影像的同名点，建立数字高程模型。在产生DEM数据时，地形变化复杂的地区，增加网格数量(提高分辨率)，而在地形起伏不大的地区，则减少网格数量(降低分辨率)。

(3)不规则三角网法(Triangulated Irregular Network,TIN)

对有限个离散点进行三角划分，得到的每一个三角形确定一个局部平面，而后插值计算出各网格点的高程，从而生成DEM。TIN与高程矩阵的不同之处是能随地形起伏变化的复杂性而改变采样点的位置和密度，因为能够克服高程矩阵中的数据冗余问题，同时还能按山脊、山谷线、地形变化线等重要的地形特征进行数

字化得到满足精度要求的DEM。

3.2 DEM 的数据采集

如前所述，建立DEM，首先必须测量一些点的三维坐标，这就是DEM数据采集或DEM数据获取，这些具有三维坐标的点称为数据点或参考点。

3.2.1 DEM 数据采集方法

DEM的数据采集的方法按采集的方式可分为选点采集、随机采集、沿等高线采集、沿断面采集等：按数据采集的方法分，有人工、半自动、自动采集等；按数据的来源分，有野外实地直接测量获取DEM数据、利用摄影测量方法获取DEM数据、以地形图为数据源的DEM数据获取方法等。

1．野外实地直接测量获取DEM数据

这种方法适用于大比例尺、精度要求高、采集面积范围较小的DEM数据获取。主要仪器是全站仪以及具有相应接口的便携机或微机。其基本过程是根据测量学原理，利用上述野外测量仪器和设备测定控制点和采样点的空间位置。为了确保地形数据的精度，总是选择地形特征点、线进行采样，以数字形式将其记录并存储在计算机中。该方法的优点是可以获取高精度的DEM数据，其缺点是劳动强度较大、效率较低，仅适用于在小范围内作业。

2．利用摄影测量方法获取DEM数据

该方法以航空、航天摄影所获得的立体像对作为数据源，根据摄影测量的基本原理，在解析测图仪或数字摄影测量系统上经过内定向、相对定向和绝对定向等过程，采用自动或半自动方式，按一定的间距，采样出DEM数据。

该方法的缺点是数据源(立体像对)获取的成本较高、采集作业要求具备专业的仪器设备(主要是解析测图仪或数字摄影测量系统)和训练有素的摄影测量专业技术人员。

3．以地形图为数据源的DEM数据获取方法

从地形图上获取DEM是目前应用最广泛的一种方法。这是因为采用这种方法所需的原始数据源(地图)容易获取，对采集作业所需的仪器设备和作业人员的要求不太高，采集速度也比较快，易于进行大批量作业。对于测绘部门，还可以利用分版图，分版采集各类要素，提高作业效率。从地形图上采集DEM通常有两种方法：手扶跟踪方法和地形图扫描矢量化方法。

(1)手扶跟踪方法

该方法利用手扶跟踪数字化仪在图上沿等高线逐点采集图形的直角坐标 (X，

Y)，并赋予属性编码(Z或ID)。其作业步骤如下：

1)地图扫描；

2)地图数字化。包括：地图定位、菜单定位、设置坐标原点、求方向角、图幅四个图廓角点的数字化、图幅内容数字化；

3)地图数字化数据预处理(误差改正)。

该方法虽然具有简便易行、对作业条件要求不高的优点。但是，作业劳动强度极大、效率低，所采用的数据精度也难以保证，特别是遇到线划稠密地区，几乎无法进行作业。显然用该方法完成大面积DEM数据的采集任务是不现实的。

(2)地形图扫描矢量化方法

地形图扫描矢量化是采用扫描仪或影像的灰度或颜色，得到栅格数据以每单位距离内采集多种幅面的地图再经过去噪声、细化、弧度跟踪、拓扑生成等处理，转换成矢量数据结构。地形图扫描矢量化的原理和主要过程如图3.2所示。

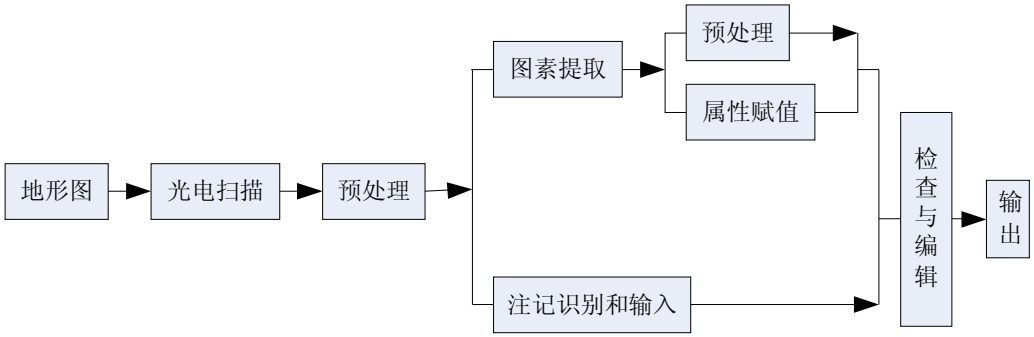


图3.2 地形图扫描矢量化的原理和主要过程

扫描矢量化方法具有速度快、人工干预少、精度高等优点，还可以减少手扶跟踪不准确造成的误差。随着高性能计算机的发展、扫描技术的成熟以及数字图像处理技术不断发展和应用，目前利用地形图扫描后的数字图像，进行数据采集的技术已基本成熟，相应的软件也不断涌现。然而这种方法仍然存在不少技术难点，例如，扫描数据量大，增加计算机的存储负担；后续处理比较复杂和费时；扫描有时造成灰度不均、漏线、污点等噪音，消除这些噪音需要特殊技术。

3.2.2 DEM 数据采集原则

数据采集是DEM的关键。在地形图上的高程数据主要以等高线的形式表示，此外，独立高程注记点数据更具有重要的意义，如山脊线、谷底线、渐崖线、山坡转折线等分布在地形线上的高程数据，是表示地形转折的控制数据，是数据采集的重要目标。研究结果表明，由于实际地形无一定数学规律可以遵循，因此影响数学模型精度的主要因素是原始数据的获取。在一定的地形条件下，DEM的精度与原始数据的精度呈线性从属关系。任何一种内插方法，均不能弥补由取样点

不当所造成的信息损失。因此，决定DEM精度的是数据的采集密度和采点的选择。但数据点太密，会增大数据获取和处理的工作量，数据的冗余也会增加，比如在单调坡面上过多的采点无助于地形特征的表示。因此，与采集点密度相关的就是选点问题，一个点对构成地貌形态中贡献的大小，由该点的不可置换程度(是否能够从周围的点派生)决定，该点不可置换程度越大，表示它在构造地貌形态中贡献越大。

综上所述，数据采集一般遵循以下原则：

1．在DEM数据采集之前，根据DEM的精度要求确定合理的采样精度：

2．在DEM数据采集过程中，根据DEM的精度要求确定合理的取样密度，单调地形应均匀采点，密度不必过大，对变化明显的地形应密集采点，尽量采集地形转折处的数据点；

3．不应出现大的空白区，如对于大片平坦地区应保证最低的采点密度。

第四章 三维分形地形模型的建立

4.1 自然景物模拟方法概述

运用计算机来进行自然景物的模拟逐渐成为计算机图形学领域中的一个重要分支，为了模拟丰富多彩、千变万化的自然景物，一系列的模型曾被提出，如过程模型，生长模型、粒子模型、分形模型等等[19][20]。

用于自然景物模拟的建模方法可分为三类：模拟规则景物外形的建模；基于物理特性的建模；模拟非规则景物外形的建模。

1.用于模拟规则景物外形的方法以几何造型技术为主(包括实体造型和曲面造型)。该方法可以完美地模拟建筑物、汽车等物体。而在自然界中某些物体只能被认为大体上是规则的，其局部则是不规则的，这时将以几何造型为基础采用法向扰动法或分层样条法来产生更为真实的效果。法向扰动法或分层样条法的基本原理是不影响原景物的大致形状，但改变了景物表面的局部细节。

2.由于许多景物的外形及运动方式都是由其整体的物理特性决定的，因此可以利用相关的物理定理来建立模型。该方法比较典型的有：

(1) 约束模型(包括动力约束模型和能量约束模型). 先给出物体受到的约束集合，进而用数学等式、不等式、微分方程式和函数将这些约束表示出来，通过方程的解析解或数字解来模拟物体的外形或运动形式，例如，用动力约束模型来模拟人体关节的运动就非常成功。

(2) 用于模拟柔软物体或柔软表面的模型。该模型将物体划分为二维或三维的栅格，每个栅格点由弹性单元、阻尼单元、塑性滑移单元连接，应用弹塑性理论的有关定律来模拟不同材料的柔软物体或柔软表面在受到不同力作用时的形状或运动情况。

3.用于模拟不规则景物外形的方法，包括粒子系统、分形模型、分形布朗运动等等。

(1)粒子系统(Particle Systeus)是迄今为止被认为模拟不规则模糊物体最为成功的一种图形生成算法。它采用一种完全不同于以往造型、绘制系统的方法来构造、绘制景物。景物被定义为由成千上万个不规则的、随机分布的粒子所组成，而每个粒子均有一定的生命周期，它们不断改变形状，不断运动。粒子系统的这一特征，使得它充分体现了不规则模糊物体的动表性和随机性，很好地模拟了火、云、水、森林和原野等自然景观。

(2)分形模型。分形是其组成部分以某种方式与整体相似的形。也就是说分形

是指一类无规则，混乱而复杂，但其局部与整体有相似性的体系(自相似体系)，并且体系形成过程具有随机性，从而体现了自然景物的一般特性——自相似性和随机性。分形方法在模拟多种自然现象时已证明是有用的，因此分形模型在模拟自然景物如云彩、山脉、湖泊、火焰等方面取得了令人惊叹的效果。

(3)分形布朗运动(Fractional Bromaian Motion,FBM)。Mandelbrot和Van Ness提出的一类一维高斯随机过程，它可用来作为自然界中许多自然时间序列的模型[21]。由于FBM很好地描绘了许多自然现象(如地域、星球表面、山脉、丛林等等)，故在计算机图形学中将它作为表示不规则物体的模型是最理想的。和法向扰动法相比，FBM的“不规则性”容易通过自相似参数及方差来控制，且由于FBM的自相似性，它的“不规则性”从景物轮廓到细节都是一致的，从而更深刻地模拟了不规则物体。用FBM来模拟不规则物体的关键是FBM的计算。通常，计算FBM的时间复杂度是O(nlog(n)),这种复杂度使图形生成变得十分昂贵。于是Fournier,Fussell几乎与Carpenter同时提出了应用递归点细分技术来合成FBM。随机曲面法已成为计算机地形模拟中一个有代表性的方法，被称为中点位移法。如今，在许多电影和动画片中都使用了中点位移法。

4.2 基于分形技术的地景仿真技术

4.2.1 分形地形模拟原理

自然界里许多现象都具有与观测尺度无关的标度特征，这种无标度特征的根源就是自相似。Mandelbrot指出一个分形曲面的主要特征是其自相似和增量的平稳性。自然地形是复杂系统物理作用在不同空间尺度上的最终结果，因此它极可能是一个典型的分形曲面。早在分形理论诞生之前，地球物理学家已对地形模型的表达问题作了大量的研究。Sayes和Thomas从跨越近80年空间尺度的地形数据分析中得出，自然地形是非平稳的。接着Berry和Hannayca进一步建立了地形描述模型，给出了著名的变异差(variograms)函数[22]和地形剖面的功率谱密度：

$$\mathrm{E}\left[\mathrm{X(x)\text{-}X（x+d）}\right]^2 = k(|d|)^{2H} \tag{4-1}$$

$$G(w) = 2p\,kw^{-\partial} \tag{4-2}$$

以上两式中，X(x)为地形高程．E[]为统计期望，H和∂为描述地形变化的统计参数。式4-1和4-2已在地学领域得到了广泛的应用，是地形模型表达的一个经典模型[23]。

Mandelbrot和Ness将布朗运动(BM)拓展到分形布朗运动(FBM)。并由此建立了

随机分形理论。该理论表明，任何一个表面，在局部条件作用下作形状随机修改，经数次反复以后，将形成一个满足FBM特征的分形表面。式2.1与2.2就是FBM的两个最重要的特性。因此，Mandelbrot将Berry和tiannay的模型纳入了其分形理论的框架之下，并作了进一步的证实工作。

真实地形是否是一个FBM表面? Gilbert和Scholz等地球物理学者又作了许多的统计验证工作，研究表明，许多地形能够很好地由FBM表达，但真实地形的自相似性仅存在于一定限度的无标度区间之中。此外，有部分地形其自相似性随尺度而变化。人们认为，至今为止FBM最好地描述了真实地形的随机过程[24]。

在计算机图形学中，基于FBM的地形模拟之所以取得了巨大的成功，其主要原因就在于上述的理论背景。目前FBM方法已成为地形模拟的主流技术，它比传统的纹理映射法和法向扰动法具有更多的优越性。

4.2.2 FBM 的特性及其合成技术

分形地形模拟技术的实质就是合成一个满足FBM特性的空间曲面。关于FBM随机过程的统计性质及其数学证明可见相关文献[25][26]。下面总结FBM最主要的几个特性。设X(t)为分形布朗随机过程：

(1) X(t)具有统计自相似性

$$p(X(t)<x)=p(X(lt)<l^{H}x) \tag{4-3}$$

(2) X(t)是非平稳过程

$$E[X(t)X(s)]=C[|t|^{2H}+|s|^{2H}-|t-s|^{2H}] \tag{4-4}$$

(3) ΔX(t,Δt) 具有统计自相似性

$$\Delta X(t,l\Delta t) \quad |\Delta t|^{KH}\,E[|\Delta X(t,\Delta t)|] \tag{4-5}$$

(4) 表示概率分布相同，ΔX(t,Δt)绝对矩满足幂律关系

$$E[|\Delta X(t,\Delta t)|^{K}]=|\Delta t|^{KH}\,E[|\Delta X(t,\Delta t)|] \tag{4-6}$$

(5) ΔX(t,Δt) 是平稳过程

$$E[\Delta X(t+s,\Delta t)\Delta X(t,\Delta t)]=C[|\Delta t+s|^{2H}+|\Delta t-s|^{2H}-2|s|^{2H}] \tag{4-7}$$

(6) X(t)的平均功率谱密度为幂型

$$S_X(w)=C|w|^{-2H-1} \tag{4-8}$$

(7) ΔX(t,Δt) 导数 ΔX' 的功率谱密度为幂型

$$S_{\Delta X'}(w)=C|w|^{-2H+1} \tag{4-9}$$

FBM是现代非线性时序分析中的重要随机过程，它能有效地表达自然界中许多非线性现象，目前已发展了多种FBM合成技术，下面作一概略介绍。

(1)切变位移法。将FBM视为分形泊松分布随机声的极限，n维泊松场中的每点是无限阶跃随机变量的累积和。

(2)分形高斯噪声法。分形高斯噪声(FGN)定义为FBM增量的导数，即式4.9中 $\Delta X'$ 。它表示为高斯白噪声与一个幂型函数加权的滑动平均。

(3)逆Fourier变换法。对高斯噪声进行Fourier变换，除以 $f^{b/2}$ 后再反变换为一个满足FBM谱(式4.8)的随机函数。

(4)Weierstrass-MandeIbrot随机函数法。W-M函数亦是一类处处不可微的连续函数，在满足一定条件的变换系数下，可产生FBM曲线。

(5)小波变换法。以定义的相邻倍频变化率对高斯白噪声序列进行小波正交合成，可得FBM。

(6)协方差矩阵变换法。通过Cholesky分解技术，可以将一个高斯自噪声序列变换为具有一定协方差矩阵的相关序列，以此产生一个FBM。

上述方法均能合成严密精确的fBln函数。然而这些方法并不是都可以直接用于计算机地形模拟，因为在图形学中，一个好的地形模拟方法必需具有高效的时间和空间效率，同时模拟结果还需具备较好的视觉效果，甚至为了算法的简洁性和适应性，牺牲部分FBM数学特性也是可行的。

4.3 分形地形建模方法

分形几何是Mandelbrot于1975年提出的一个名词，分形几何关注的是物体的随机性、奇异性和复杂性，他试图透过混乱现象和不规则构型揭示隐藏在背后的局部与整体的本质联系和运动规律。

分形几何具有细节无限以及统计自相似性的典型特性，它用递归算法使复杂的景物可用简单的规则来生成，在现代的计算机图形学中，分形几何在对自然现象的真实绘制和建模方面起着重要作用。

地形建模是自然景物建模中很重要的一类，许多分形地形建模都与分数维布朗运动(FBM)这一数学模型有关，FBM是现代非线性时序分析中的重要随机过程，它能有效地表达自然界中许多非线性现象，也是迄今为止能够描述真实地形的最好的随机过程[27][28]。

分形地形建模大致可归纳为泊松阶跃法，逆Fourier变换法、中点位移法、逐次随机增加法和带限噪声累积法等几类。

4.3.1 泊松阶跃法(Poisson Faulting)

泊松阶跃法[22][25]是最早由Mandelbrot研究的地景生成算法。它是将泊松分布用于FBM的产物。这是变位移法在二维空域上的扩展。对于直线上的随机行走，t时刻BM的位置向量X(t)可以看作为粒子独立跳跃序列的累积位移：

$$X(t)=\sum_{t=-\infty}^{+\infty}A_tP(t-t_1) \tag{4-10}$$

式中，A_t 是一个高斯分布的随机变量，t_1 为服从泊松分布的时间，一维情形，阶跃函数p(t)定义为:

$$P(t)=\begin{cases}t^{H-\frac{1}{2}},t>0\\-|t|^{H-\frac{1}{2}},t\le 0\end{cases} \tag{4-11}$$

泊松阶跃的实质就是在服从泊松分布的间隔上，将高斯随机分布（或称步长函数）加到一个平面或球面上，其结果具有FBM特征。这种方法很适合于用球面生成类似星球的物体，这时只需将X(t)构造为半径为r的函数X(r)。其主要缺点是算法的时间复杂度达到了 $O(n^3)$ 。

4.3.2 逆 Fourier 变换法(Inverse fourier Transformation)

对高斯白噪声作Fourier变换 $T(w)$ ，其功率谱密度 $S(w)$ 应为式4.8幂函数型，故 $T(w)$ 应满足条件 $T(w)\propto w^{\frac{b}{2}}$ 。对满足条件的 $T(w)$ 作逆Fourier变换即可得到FBM序列X(t)。二维的逆Fourier变换为：

$$X(x,y)=\sum_{k=0}^{N-1}\sum_{l=0}^{N-1}a_{kl}e^{2p\,(kx+ly)} \tag{4-12}$$

其中变换满足幂型条件：

$$E(|a_{kl}|^2)\propto(k^2+l^2)^{-H-1} \tag{4-13}$$

a_{kl} 是通过对二维白噪声作Fourier变换，然后除以 $f^{b/2}$ 得到。

逆Fourier变换法[22][25]可以通过高频成份的调节来任意逼近FBM，生成地形很好的FBM特性；此外，模拟视觉效果可通过尺度变化即间隙度(1acunarity)改变。“lacuna”是表示“空隙”的拉丁语，Mandelbrot引入“Lacunarity”一词，作为量化分形地表纹理的一个参数。其取值范围为larcunarity>1。由于分形地表的纹理可以看成是由地表上微小扰动引起的凹凸起伏形成的，且随着lacunarity的增大，分形地表的凹凸起伏程度变缓，峰与峰或谷与谷之间的间距变大，因此产生不同的纹理图

像效果。逆Fourier变换法生成的地形表面是一次性计算得到的，这样不利于继续细节或改变空间分辨率的操作。另一个问题是生成的图像是周期变化的，其视觉效果不佳，且算法的时间复杂度是O(nlogn)，一个解决的办法是只取一大幅生成图像中的某一块(1/4或1/2)。

4.3.3 中点位移法(midpoint displacement)

Wiener早在1920年就提出了运用递归中点技术来合成布朗运动过程，在此基础上，Fournier、Fussell几乎与Carpenter同时提出了采用递归点细分技术来合成FBM随机曲面的理论，此法已成为计算机地形模拟中一个有代表性的方法，即中点位移法[22][25][26]。

中点位移法的基本原理是基于ΔX方差幂律关系：

$$E\left[(X(t+\Delta t)-X(t))^2\right]=\Delta t^{2H}s^2 \tag{4-14}$$

式中X(t)是满足高斯分布$N(0,s^2)$的随机变量。中点位移是对线段中点处的高程进行位移，然后将分割的线段再细分出中点，并进行进一步的位移，以此过程递归进行，直到满足一定的空间分辨率。其中，线段中点处的高程位移量为：

$$X_n(\frac{t_1+t_2}{2})=\frac{1}{2}(X_{n-1}(t_1)+X_{n-1}(t_2))+\Delta_n \tag{4-15}$$

在此，Δ_n为0均值Δ_n^2方差的高斯随机变量，可推得n步细分后，

$$\Delta_n^2=\frac{s^2}{(2^n)^{2H}}\left[1-2^{2H-2}\right] \tag{4-16}$$

中点位移法是标准的分形几何法，可用作快速地景生成。它是利用细分过程中，在两个点或多个点之间进行插值的方法来进行地景建模，因此中点位移法产生了真正的分形地表．但是不同细分阶段产生的点在相邻区域中有不同的统计特性，这常会留下一道明显痕迹，即所谓的“折痕问题”(creasing probiera)，当添入更多细节时，该轨迹也不消失。虽然中点位移法有这样一个比较明显的缺陷，但是它的高速度以及具有为已有形状增加细节的能力，使它成为一个有用的面向应用的分形算法。

4.3.4 逐次随机增加法(Successive random additions)

根据采样定理。当空间分辨率提高2倍时，增加的高频分量将改变所有的原始点值。此法以中点细分递归技术进行插值，但需对所有的点增加高程位移。对该

法作进一步扩展，可使中点细分扩展到任意尺度细分，即 $r=\frac{1}{2}\to 0<r<1$，有（$r=\frac{t}{T}$）：

$$X_n(t)=\frac{t}{T}(X_{n-1}(T)-X_{n-1}(0))+\Delta_n \tag{4-17}$$

可导出 Δ_n 满足：

$$\Delta_n^2=\frac{1}{2}s^2(1-(\frac{t}{T})^{2-2H})t^{2H} \tag{4-18}$$

逐次随机增加法[22][25]是一个灵活的细分方案，如果需要利用上一级细分过程确定的点(不是所有的点都必须用)，则这些点需要首先增加一个服从某种分布的随机变量。一般新的点可通过在上一级细分水平基础上进行线性或非线性插值得到。此法合成的地形是平稳的，消除了皱折问题，可以任意的尺度进行细分，间隙度调节灵活。此法实现简便，但计算量稍大，算法的时间复杂度依赖于最终的分辨率和地表lacunarity化的程度[25]。

4.3.5 带限噪声累积法(Summing band limited noises)

带限噪声累积法[22][25]是将频率范围受到严格限制的信号反复叠加，其中每一个信号的同谋是随机的，即噪声。因此，这种方法也称噪声合成法。设 $N(\vec{P})$ 是一个噪声函数，$\vec{P}$ 是空间坐标向量，此方法可表示为:

$$X_0=(N(\vec{p_0})+C_t)C_S w^p+C_0 \tag{4-19}$$

X_0 是高程场中某点的初始高程，C_t 是一个常数平移量，C_S 是噪声函数的比例因子，C_0 是地形海平面的高程值，n次迭代后，高程场中点位值为：

$$X_n=X_{n-1}+X_{n-1}(N(\vec{p_n})+C_t)C_S w^p \tag{4-20}$$

其中 $w=r^{b/2}$。这是一种基于函数的建模方法，每个点的确定独立于它的所有邻接点，因此此法具有较强的控制地形局部频率成份、分数维以及其他统计性质的能力。实际上，山体在不同的高度具有不同的统计特征，如山脚一般比较平整，山顶凹凸变化较大。这些变化说明，地形存在着局部统计性，此法在这一点上较其他方法有大的不同，它能较好地模拟这类地形，也是此法区别于一般的随机分形算法的独特之处。

4.4 随机中点位移算法分类

以上对分形地景建模算法进行了简单的概述。由于现实自然景物的复杂性，

真实模拟自然景物的计算量还是很大的，因此特别需探索出一些高效的算法，尤其在虚拟仿真系统中，通常希望产生实时动画的效果，因此算法的效率更显重要。由于中点位移法易于实现，且运行速度快，所以是一种常用的随机分形算法。中点位移法又叫作随机中点位移法，是最简单和经典的方法，是对分形布朗运动的直接应用。

一维随机中点位移法的思想是：针对一已知端点高程(或属性)的线段，其线段中点的高程(或属性)为其两端点高程(或属性)的平均值再加一随机位移量，位移后的两线段再进行上述中点细分并位移，递归直到满足所需的分辨率为止。一般情形下，采用服从高斯分布的随机数，但由于实现中较为复杂，可利用减小随机数范围的方法，实现同样的效果如图4.1所示。

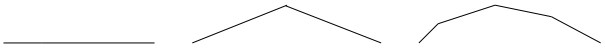


图4.1 一维表示下的中点位移

随机分形理论认为，对任一条曲线在局部上随机扰动，重复足够多次，便可形成一条满足布朗运动特性的分形曲线。将其推广到二维，依据表面构网方式的不同，其模拟可分为三角形细分法、方形细分法、菱形-方形法等方法，其实现思想与一维类似。

4.4.1 三角形边线细分法

一种典型的三角形边线细分法如图4.2所示。该细分法就是对三角形各边进行等分，再将各中点连接起来，这样一个三角形就变成了四个小三角形．然后在各中点上增加一个高斯随机变量，其标准偏差由下式给出：

$$S = K2^{-iH} \tag{4-21}$$

其中，i为迭代次数；k是一个比例因子；H是分形维数。

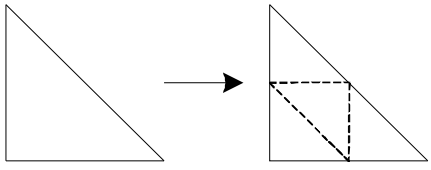


图4.2 三角形边线细分法

由于各个三角形在不同的时间，以不可预知的顺序进行细分操作，为了保证三角形之间有良好的衔接性，必须使相邻的两个三角形的公共边在高程上具有相同的位移量，另外由于各相邻三角形间没有信息传递，因此这是一种与细分操作前后无关的方法。这一点很重要，因为这导致了所谓的 “折痕问题”，即地景表面

留有不自然的线型轨迹，并且不能通过局部平滑消除。

4.4.2 方形-方形细分法

此法如图4.3所示，根据“新生正方形的面积是已存在正方形面积的一半”这一原则，产生插值点的位置坐标，其高程值由权重比9:3:3:1决定，距离最近点具有最大的权系数。

这种插值方法构成了一个双二次曲面的效果，它的一个重要优点是地景表面的法线是连续的，所以消除了三角形细分中的折痕现象。此外,由于双二次曲面的效果影响，用这个算法构造的轮廓比较柔和。但是如果需要，也可以通过减小H使表面更粗糙。另外，还可以通过放大Y值，使山峰更突出，从而可以得到陡峻的地景效果。

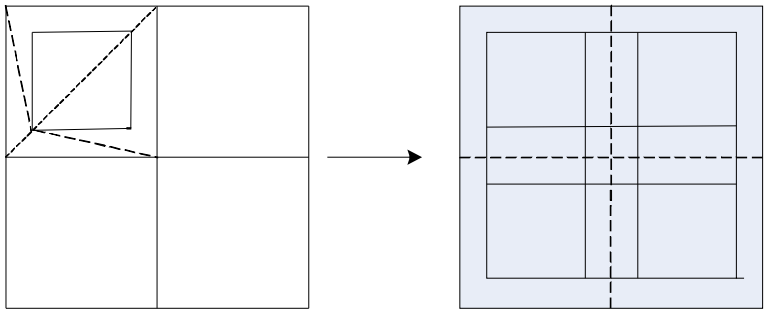


图4.3 方形-方形细分法

虽然中点位移法简单、快速，但通常难以预测生成结果的大致形状。针对这个问题，出现了一些改进的算法，如通过给定初始迭代次数和初始迭代数据来控制生成图形的基本形状和轮廓等。自70年代中期“分形”概念提出到80年代。由于分形几何迅速发展，从而在地形生成方面成功地绘制出了相当逼真的地景。但此后，对地形的生成与仿真的研究基本停留在“视觉上可以接受”这一标准上，没有与实际所需的真实地形、地貌联系，应用受到一定的限制。

4.4.3 菱形-方形细分法(Diamond—Square)

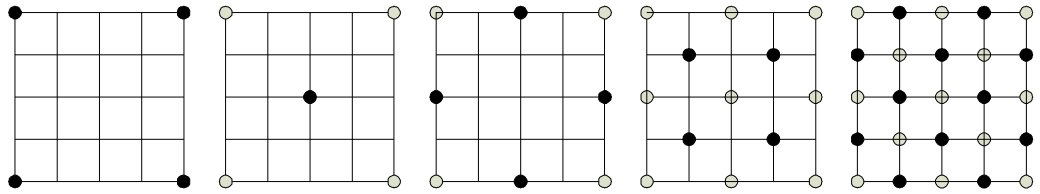
其基本思想如下：首先从一个尺寸为1的空二维数组开始，其中n为正整数，将四个角点的高程(或属性)初始化，以 $(2^n+1)(2^n+1)$ 网格为例，(如图4.4(a)中的黑点)，然后进行以下两步的递归细分：

(1)diamond步：取四个点的正方形，在正方形中点生成一个随机值，中点为两对角线交点。中点高程(或属性)值由四个角点高程(或属性)值的平均再加上一个随机位移量计算得到。这样就得到了一个棱形(钻石形)网格。如图4.4(b)中，新值显

示成黑色，已经存在的点显示为白色。

(2)square步：取每个四点形成的棱形，在棱形的中心生成一个随机位移值。平均角点高程(或属性)值再加上与diamond步相同的随机量，计算出每条边中点值。这样又得到一个正方形格网。如图4.4(c),其中黑色显示新值，现存值为白色。

再重复diamond步(如图4.4(d))和square步(如图4.4(e))n=2次，直到得到整个$(2^2+1)(2^2+1)$网格，即5×5网格。图4.5为生成的5X5格网对应的三维地形线框图。



(a) (b) (c) (d) (e)

图4.4 中点位移法两次迭代示意图

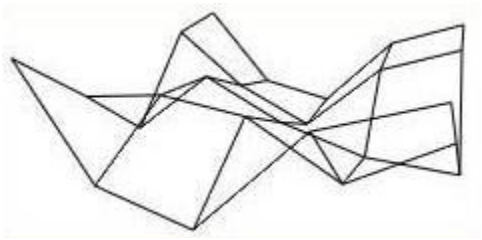


图4.5 5×5网格对应的三维地形线框图

这样，如果已经生成了一个种子正方形并经过单独一次细分过程，将得到4个正方形;第二次经过该过程得到16个方形如图4.4所示，第三次得到64个方形，增长得很快。正方形的数目为2^{2i}，其中i是递归经过细分过程的次数。经过多次迭代后，不仅细化了基平面，也增加了空间三维点元素的密度，当三维点元素的密度达到要求时,就可以得到具有分形布朗运动特性的地形高程模型。

4.5 三维分形地形模型建立

4.5.1 基于迭代策略的”Diamond-Square”算法

在二维空间中进行分形插值产生分形地形数据，二维FBM是一种很有效的分形模型，用分形FBM可以较好地给予描述。目前，构建分形FBM的方法很多，最为常用的还是基于网格的随机中点位移法和基于三角形的随机中点位移法，前一种方法适合于具有规则网格的地形数据，目前大多数地形模型都是使用规则网格地形数据。

二维随机中点位移法实现的算法方案有很多，难点是必须消除分形曲面成过程中可能产生的裂缝和皱折(creasing problem),经过比较分析，本文采用著名的“Diamond-Square”算法。

早期的Diamond-Square算法均使用递归策略完成，递归策略虽然编程简单，但存在棱形角点丢失问题，从而产生裂缝或皱折。原因是：一个递归实现可能采用如下形式：

执行diamond步：

执行square步：

减小随机数范围：

调用自己四次。

这是个很简洁的实现，但要求用不充足的数据生成某些点。原因是经过第一遍之后，将再次调用以执行square步，这时并没有一个棱锥四个角的全部数据，从而产生裂缝或皱折。

虽然基于迭代策略编程量较大，但可在数组网格中间区域避免上述问题，而数组网格边界的点可通过特殊处理来消除，所以近期的研究均为基于迭代策略的。

基于迭代策略的“diamond—square”算法伪码：

当正方形(square)边长大于K时

{

遍历数组，对每个正方形表达执行diamond步：

遍历数组，对每个棱形表达执行square步：

依随机位移量的计算公式，减小随机数范围：

}

K为常数，一般取2的幂，其大小决定了迭代次数的多少。

4.5.2 中点位移法中随机偏移量的确定

在上述算法中，每次求取新的高度值时，都需要附加一个随机偏移量，为保证分形布朗运动中的自相似性，该随机偏移量须符合高斯分布，其均值为0，方差为 Δ_n^2 。随着迭代过程的不断深入，基平面上的正方形网格会不断细化，求取中点的网格尺度也随之减小。相应地，在每次迭代过程中，相应随机偏移量的扰动幅度也要成比例减小，它是与随机变量的方差直接相关的。

根据自相似性的特点，取任意 t_0 和r>0,分形布朗运动中随机变量 $X(t_0+t)-X(t_0)$ 和 $\frac{1}{r^H}(X(t_0+rt)-X(t_0))$ 是自相似性的。不失一般性，令 $t_0=0$ 且X(0)=0,则 $\frac{1}{r^H}(X(rt))$ 为随机偏移量。式中H称为分形参数，且0<H<1也是自相似性的，它们具有相同的有限维数的联合分布函数，二者在统计意义上并无差别。

在三维的分形布朗运动中，上述的自相似性依然存在，只是此时函数的定义域是在二维平面上，高度坐标Z是x,y两坐标的函数，即z=Z(x,y)。在中点位移中，高度坐标z的生成过程总伴随着基平面上网格的不断细化，如图4.6所示。设初始的网格尺度为d，且其四个角点处的高度值已知。在求出方格中心点及四条边中点处的高度值的同时，网格也完成了第一次细化，新网格的网格尺度减小为原来的$\sqrt{2}/2$，即为$\sqrt{2}/2\,d$.按此方法对网格进行细化时，每个迭代周期中网格的缩小比例保持不变，恒为$\sqrt{2}/2$，我们称之为细化因子r。

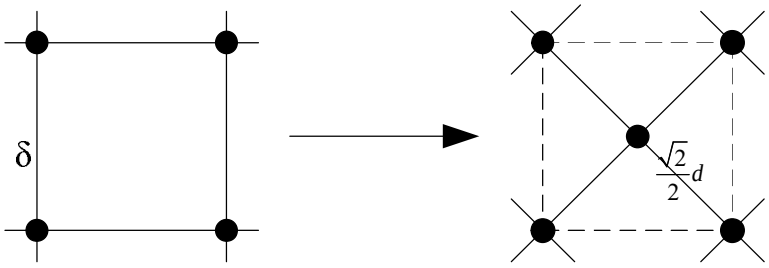


图4.6 网格细化过程

如前所述，为保证地形模型具有自相似性的特点，当基平面的网格尺度以细化因子。不断减小时，随机偏移量的方差也作相应的调整，由文献[21]可知，在中点位移法中，在第n次迭代所对应随机偏移量的方差值与初始方差值s^2之间的关系为：

$$\Delta_n^2 = r^{2nH} s^2 \tag{4-22}$$

式中，s^2为迭代开始前所设定的随机变量的初始方差，r为细化因子，H为分形参数。

如上所述，当基平面的细化因子$r=\frac{\sqrt{2}}{2}$时，代入式4.22，则可求得在第n次迭代时，中点偏移量的方差值Δ_n^2为：

$$\Delta_n^2 = (\frac{\sqrt{2}}{2})^{2nH} s^2 = (\frac{1}{2})^{nH} s^2 \tag{4-23}$$

在构建分形地形的数据模型时需要用到随机数发生器，以生成随机变量样本．在大多数开发系统(如Visual C++,Visual Basic等)中都具有实现伪随机数发生器的函数(如Visual C++中的rand())，这类函数返回在特定区间[0,A]上均匀分布的伪随机数序列，并以此随机数序列作为每次计算高度值时的随机数，常用的随机数序列的概率特征有以下几种[29]：

1．均匀分布

即所产生的随机数在特定范围内出现的概率相等。由于伪随机数发生器rand0返回在区间[0,A]上均匀分布的随机数，所以限定[0,A]的范围后，直接使用伪随机数发生器rand()就可以产生服从均匀分布的随机数序列。

2．正态分布

正态分布又称高斯分布，是一种自然界中广泛存在的概率分布形式。中心极限定理表明，如果Y_n是任意n个符合相同分布的随机变量的标准化总和，则Y_n的概率分布随着$n \to \infty$而趋于正态分布，实现过程如下：

对于：X=rand()则：

$$E(X) = A/2, D(X) = A^2/2 \tag{4-24}$$

对X进行标准化有：

$$Y = (X - E(X)) \Big/ \sqrt{D(X)} \tag{4-25}$$

对于n个Y的和：

$$Y_n = (\sum X_i - E(\sum X_i)) \Big/ \sqrt{D(\sum X_i)} = (\sum X_i - \sum E(X_i)) \Big/ \sqrt{\sum D(X_i)}$$

$$= (\sum X_i - (A \times n/2)) \Big/ \sqrt{A^2 \times n/12} \tag{4-26}$$

通常情况下，当n=3或n=4时就已经符合需要。

3．指数分布

以随机数发生器rand()为基础，可以构造服从指数分布的随机数序列。常用的基本形式为：

$$r = -\ln(rand() \times ex + 1.0) + add \tag{4-27}$$

式(4-27)中的ex、add为修正参数。这样，所得到的r值将服从指数分布。

以上所给出的随机数序列产生方式都是通用的生成方式，在具体的三维分形地形建模过程中，随机偏移量的计算公式均要用到高斯函数gauss()，而该函数在C语言库函数中并不存在，需要依据概率论中的中心极限定理对多个rand()随机数求和取平均来自行定义，因而降低了分形地形模拟生成的速度。目前，许多模拟分形地形的游戏和计算机仿真中，为了达到实时性的要求，普遍对随机偏移量出采用如下计算公式[30]：

$$\Delta i = (rand()\%(r+1) - r/D) \tag{4-28}$$

其中D为分形维数，rand()为库函数自带的伪随机函数，r取值与地形起伏特征有关。当然该数学模型并非严格基于分形布朗运动的。

4.6 实验分析

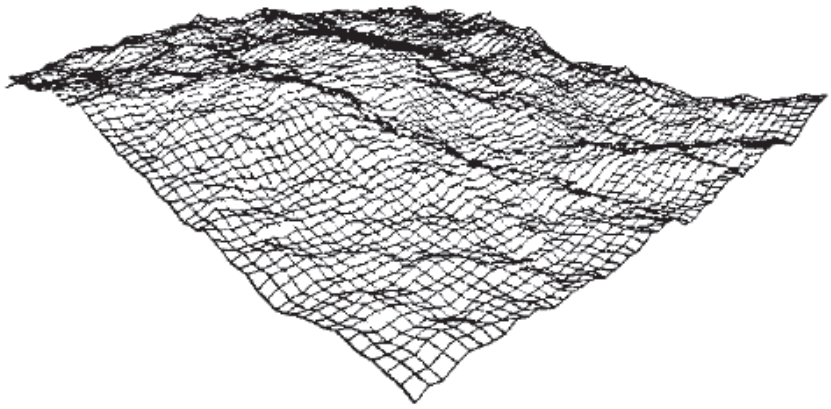
根据上述的研究思路，基于VC++6.0平台，依据Diamond-Square迭代算法编制了应用程序，实现了上述算法所描述的三维分形地形模型生成过程。实验证明，运用分形布朗运动中的中点位移法来作为地形的建模理论是可行的，该方法所形成的分形布朗运动曲面比较理想地吻合了真实地形的数学特征。

作为地形模型建模的理论基础，中点位移法还具有一些自身特有的优点，其中最主要的特点是合成速度快，时间复杂度仅是n的线性次项；另外，该方法实现非常容易，代码简单，并且可以合成任意分辨率的地形，甚至可以细分到子像素，降低图像显示的混淆度。由于是中点细分合成的图像，因此图像对于放大缩小具有不变性。

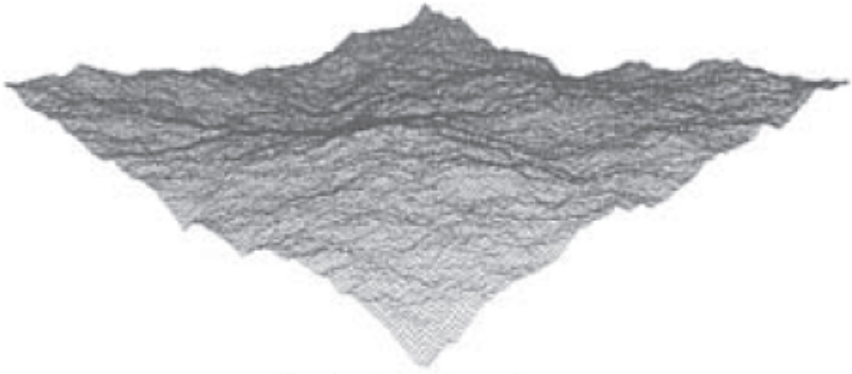
对影响地形形状的相关因素进行考查，可知，分形地形的形状决定于随机数据的迭代次数I和每次迭代过程中的中点偏移量。中点偏移量的大小受分形参数H和初始方差 s^2 影响。在一个参数区间内，三维地形与现实山地、平原、丘陵是相似的；而在另一些参数区间内，生成的三维地形则是自然界中不存在的，称之为病态的参数区间。

4.6.1 迭代次数 I 对地形形状的影响

从分形的角度来讲。随着迭代次数的增加，三维分形地形的精细程度应该有所增强，但实验表明，地形轮廓出现了一定程度的失真，山体、山谷分布细密。图4.7所示是在H值同为0.6的情况下。不同迭代次数所产生的不同效果。实验证明，迭代次数在4或5的情况下效果最佳。



(a)迭代次数I=3



(b)迭代次数I=4

(c)迭代次数I=6

图4.7 分形参数I对地形形状的影响

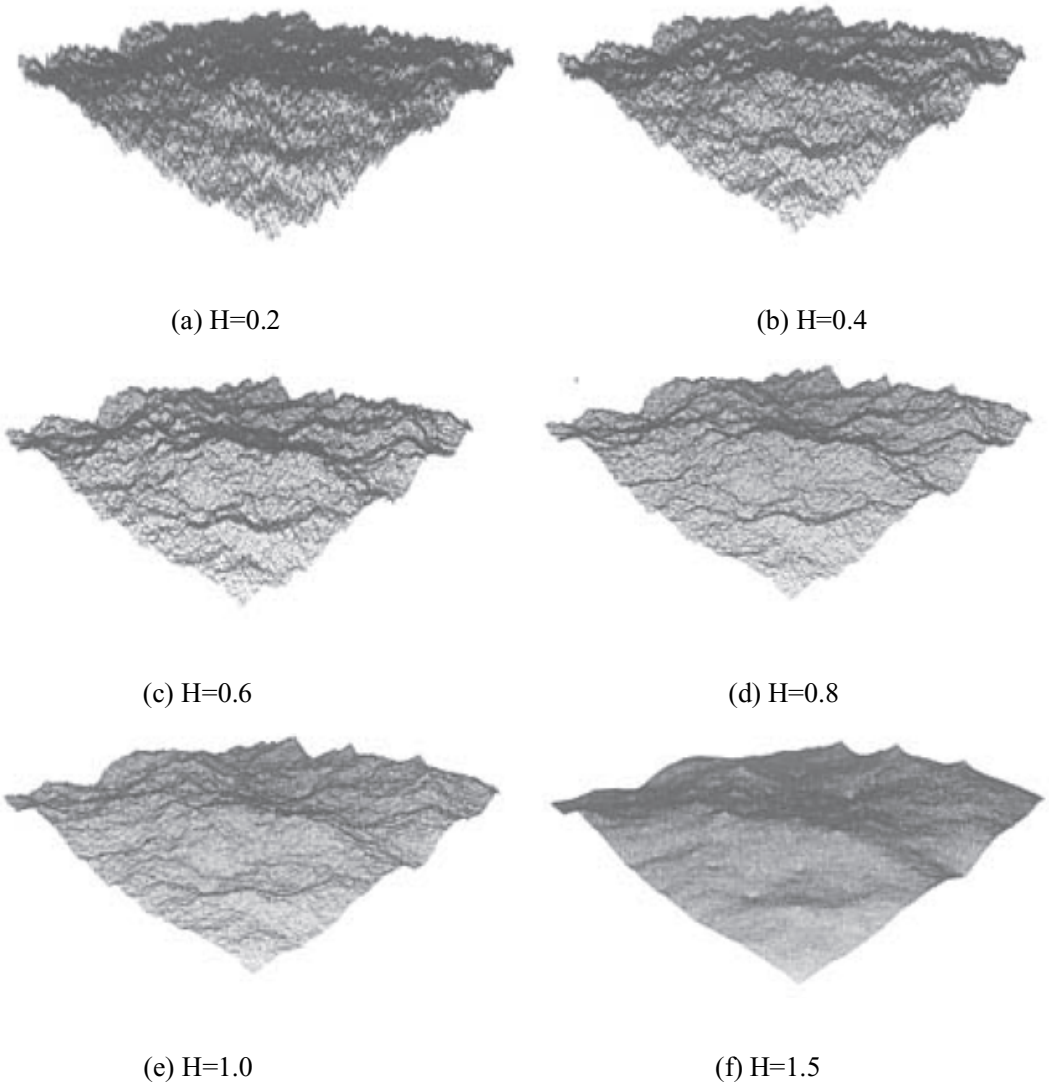
4.6.2 分形参数 H 对地形形状的影响

分形对象的分维数在一定程度上反映着它的形状特点，分维数又与分形参数H

有着密切的联系。地形的分维数D与分形参数H有如下关系：

$$D = 3 - H \tag{4-29}$$

当H在区间(0,1)之间取值时，它对地形的坡度起决定作用。在随机高度值生成的过程中，每迭代一次地形的随机数范围就减少pow(2.0,-H)倍。这样生成的地形图起伏有致，否则便有参差不齐的视觉效果。可见，H值与地形坡度成正比关系，如图4-8所示。由式4.29可知，在H由0增大到1.0的过程中，分维数D由3逐渐减小到2，相应的地形模型的形状也将会由“粗糙”逐渐变的平缓。图4.8给出的例图清楚地证明了这一理论所阐明的变化均势。另外，由图4.8可以看到，当H=1.0时，地形模型并没有像理论推导的那样，趋近于分维数D=2.0的平面，而仍然是典型的山脉地形。事实上，当分形参数H的值大于1.5时，该模型才表现出一定的平面特征。



(a) H=0.2　(b) H=0.4

(c) H=0.6　(d) H=0.8

(e) H=1.0　(f) H=1.5

图4.8 分形参数H对地形形状的影响

4.6.3 初始方差s^2对地形形状的影响

由式4.22可知，每次迭代时，中点的随机偏移量的方差Δ_n^2与初始方差s^2成正比。但是，初始方差s^2本身并不是随机变量，一旦在迭代开始前确定了它的值，则在整个迭代生成的过程中它始终保持恒定不变．实验证明，只要初始方差的绝对值不等于0，就可以保证三维分形地形几何模型正常生成和显示，而其具体值的改变对山脉形状特征的影响是非常有限的，与分形参数H所产生的影响相比可以忽略不计。

4.6.4 模型优化

实验表明，该算法偶尔还是会产生局部尖刺的问题，所以我们还需要做毛刺处理，如图4.9所示。下面给出了一个简单可行的算法：

(1)定去噪因子Burr的初值；

(2)遍历网格数据，如果某一顶点与周围顶点高程差大于去噪因子Burr，则认为是毛刺，将周围顶点的高程均值赋给该顶点高程；

(3)减小去噪因子Burr的值，重复(2)的步骤，直到达到满意的效果为止。

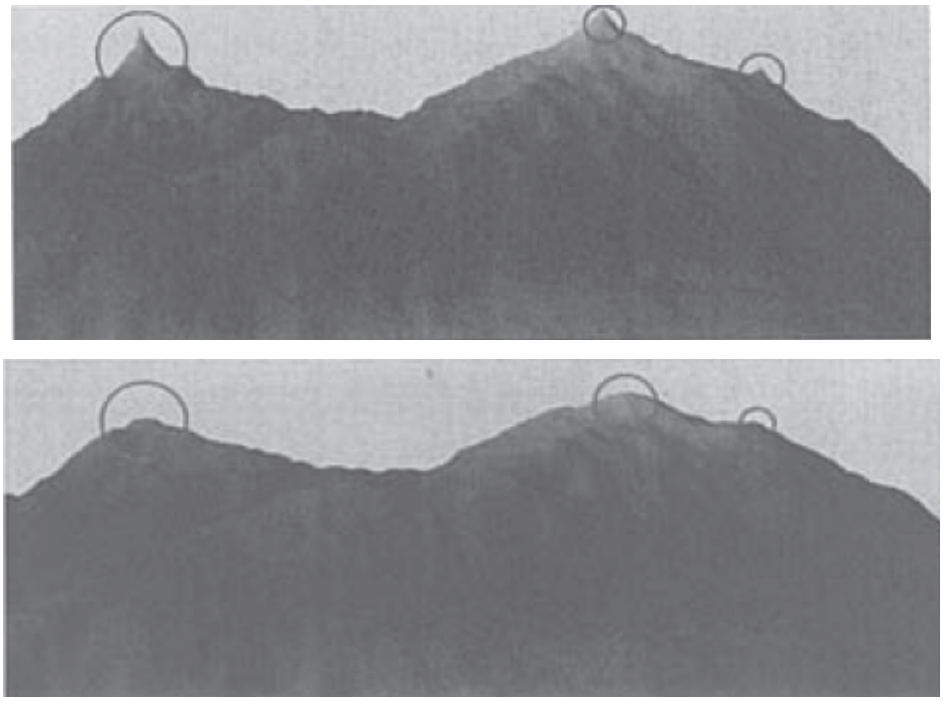


图4.9 三维分形地形的模型优化(去毛刺处理)

4.7 本章小结

(1)对自然景物模拟方法进行概述，分析了分形理论和分形布朗运动的原理；

(2)介绍了分形地形建模的各种方法，再进一步介绍随机中点位移法的分类；

(3)详细分析了随机中点位移法中的Diamond-Square算法思想，给出了其具体的实现流程；

(4)对分形地形的模拟生成进行了实验研究,验证及分析了结果并进行优化。

第五章 三维分形地形可视化系统设计与实现

本章采用以上描述的算法和技术，在 VC++6.0 平台上结合 OpenGL 图形库实现了一个三维分形地形可视化系统。该系统可通过添加按键响应函数，实现对三维地形旋转和缩放的控制。

5.1 系统的实现平台

5.1.1 OpenGL 简介

OpenGL(即开放性图形库 Open Graphics Library)，是一个三维的计算机图形和模型库，最初是美国 SGI 公司为图形工作站开发的 IRIS GL，在跨平台移植过程中发展成为 OpenGL。

OpenGL 独立于硬件和窗口系统，可运行在各种操作系统的各种计算机上，并能在网络环境下以客户/服务器模式工作，是专业图形处理、科学计算等高端应用领域的标准图形库。OpenGL 的核心库包括 100 多个用于 3D 图形操作的函数，主要负责处理对象外形描述、矩阵变换、灯光处理、着色、材质等和三维图形图像密切相关的工作。

整个 OpenGL 的基本工作流程如图 5.1[34]：

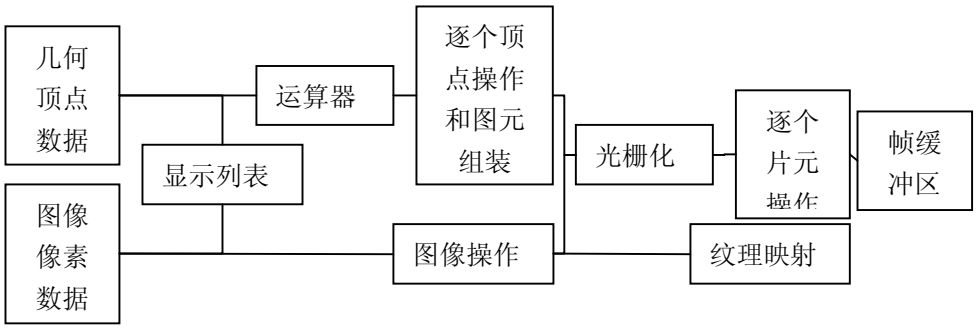


图 5.1 OpenGL 基本工作流程

其中几何顶点数据包括模型的顶点集、线集、多边形集，这些数据经过流程图的上部，包括运算器、逐个顶点操作等；图像数据包括像素集、影像集、位图集等。图像像素数据的处理方式与几何顶点数据的处理方式是不同的，但它们都经过光栅化、逐个片元处理、把最后的光栅数据写入帧缓冲器。在 OpenGL 中的所有数据包括几何顶点数据和像素数据都可以被存储在显示列表中或者立即可以得到处理。OpenGL 中，显示列表是一项重要的技术。

OpenGL 要求将所有的几何图形单元都用顶点来描述，这样，运算器和逐个顶点计算操作都可以针对每个顶点进行计算和操作，然后进行光栅化形成图形碎片；对于像素数据，像素操作结果被存储在纹理组装用的内存中，再像几何顶点操作一样光栅化形成图形片元。整个流程操作的最后，图形片元都要进行一系列的逐个片元操作，最后的像素值送入帧缓冲器实现图形的显示。

5.1.2 Windows 平台下 VC++中 OpenGL 程序设计方法

在 Windows 下用图形设备接口(Graphic Device Interface,GDI)作图必须通过设备上下文(Device Context,DC)调用相应的函数；用 OpenGL 作图是 OpenGL 函数是通过“渲染上下文”(Rendering Context,RC) 完成三维图形的绘制。Windows 下的窗口和设备上下文支持“位图格式”(PIXELFORMAT)属性，和 RC 有着位图结构上的一致。因此，在创建 RC 时与一个 DC 建立联系(RC 也只能通过已经建立了位图格式的 DC 来创建)，OpenGL 的函数就可以通过 RC 对应的 DC 画到相应的显示设备上。其中必须注意的以下方面[31]：

(1)一个线程只能拥有一个渲染上下文（RC）。即用户如果在一个线程内对不同设备作图，只能通过更换与 RC 对应的 DC 来完成，而 RC 在线程中保持不变(删除旧的 RC 后再创建新的是可以的)。与此对应，一个 RC 也只能属于一个线程，不能被不同线程同时共享。

(2)设定 DC 位图格式就相当于设定了相应的窗口的位图格式，并且 DC 和窗口的位图格式一旦确定就不能再改变。

(3)一个 RC 虽然可以更换 DC，在任何时刻只能利用一个 DC(该 DC 称为 RC 的当前 DC)，但由于一个窗口可以让多个 DC 作图从而可以让多个线程利用多个 RC 在该窗口上执行 OpenGL 操作。

(4)现在 Windows 下的 OpenGL 版本对 OpenGL 和 GDI 在同一个 DC 上作图有一定的限制。当使用双缓存用 OpenGL 产生动画时，不能使用 GDI 函数向该 DC 作图。

经过上面的分析，用 VC 来调用 OpenGL 作图方法的步骤如下:

(1)先设置显示设备 DC 的位图格式 (PIXELFORMAT)属性。这通过填充一个 PIXELFORMATDESCRIPTOR 的结构来完成，该结构决定了 OpenGL 作图的物理设备的属性。有一些位图格式 (PIXELFORMAT)是 DC 支持的，而有一些 DC 就不支持。所以程序必须先用 ChoosePixelFormat 来选择 DC 所支持的与指定位图格式最接近的位图格式，然后用 SetPixelFormat 设置 DC 的位图格式。

(2)利用刚才的设备 DC 建立渲染上下文 RC(wglCreateContext)，使得 RC 与 DC 建立联系 (wgIMakeCurrent)。

(3)调用 OpenGL 函数作图。由于线程与 RC 一一对应，OpenGL 函数的参数中都不指明本线程 RC 的句柄 (handle)。

(4)作图完毕后，通过设置当前线程的 RC 为 NULL(wglMakeCurrent(NULL，NULL);)，断开当前线程和该渲染上下文的联系，由此断开与 DC 的联系。在删除 RC 的时候要先判断 RC 句柄的有效性。再根据情况释放或者删除 DC。

5.2 系统的设计与实现

5.2.1 系统的功能模块

本文实现了一个三维分形地形可视化系统，其主要由下述几个功能模块组成，如图 5.2 所示：

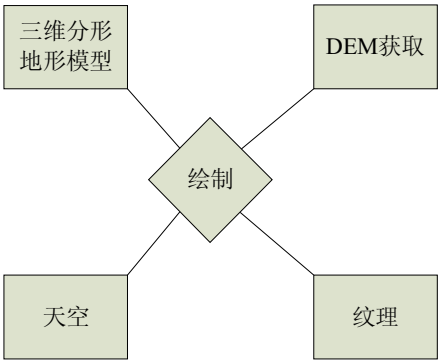


图 5.2 系统功能模块示意图

DEM 获取模块：该模块完成对地形图的读取，转化为 DEM 数据；

三维分形地形构建模块：该模块完成三维分形地形的建模和绘制，

天空模块：该模块完成天空的绘制；

纹理模块：该模块调用组织生成场景中的纹理，将纹理数据提供给绘制模块使用；

绘制：该模块不是一个独立的模块，它调用 OpenGL 函数对场景中的对象进行绘制输出，完成动画的屏幕显示，根据面向对象的编程思想，绘制作为各对象一部分，分散在个模块中。

该系统的各个功能模块耦合性强，各个模块都有数据或者控制传输。系统可以载入地形图，控制分形差值的随机种子和高度控制等参数，同时还可以控制地形的绘制模式，天空的绘制模式，绘制包围盒等附加信息。

5.2.2 系统流程图

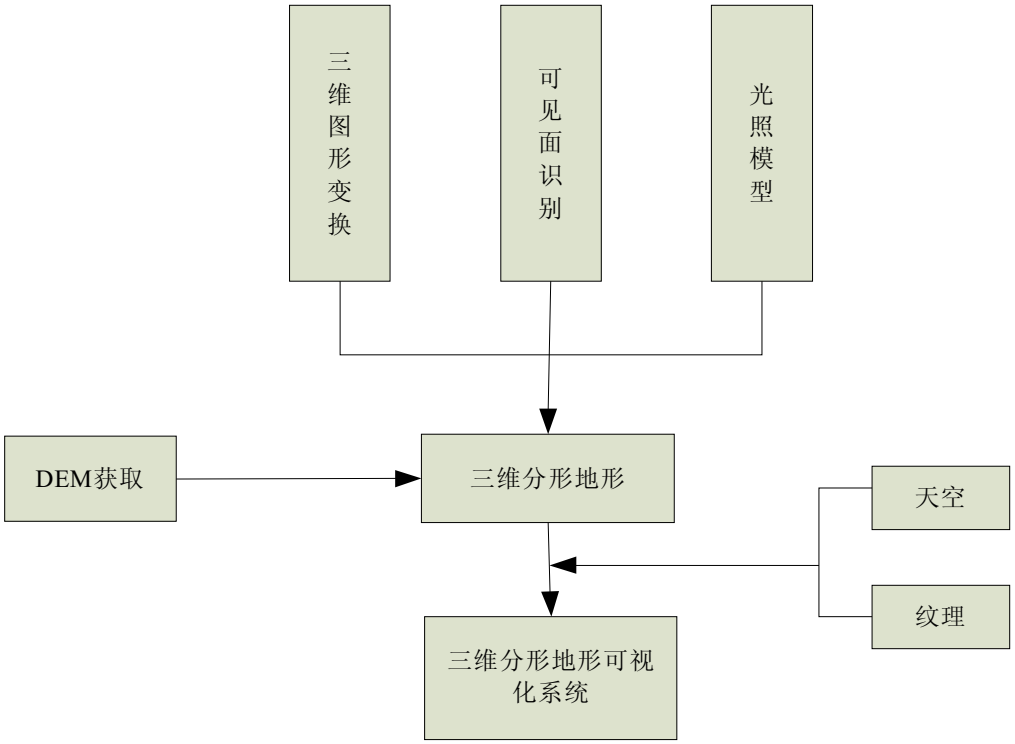


图 5.3 三维分形地形可视化系统流程图

文章的前面部分已经详细介绍了 DEM 的获取和三维分形模型的建立，下面的章节将对天空模块和纹理模块做简单介绍。

5.2.3 天空模块

关于天空的绘制常用的有四种方式：

（1） 将背景涂成天蓝色，以此简单的模拟天空。

核心代码：glClearColor(0.2f, 0.3f, 0.6f, 0.5f);
效果图如 5.4 所示：

图 5.4 蓝色背景天空

该方法简单易行，绘制快速，缺点是天空的真实度不高。

（2） 用三角形扇画多边形，映射天空纹理后模拟天空。

核心代码：

```
#define TMMAX 3.f//定义映射函数
#define TMMAX45 (TMMAX * 0.707f)
//画三角形
glBegin (GL_TRIANGLE_FAN); {
glTexCoord2f (0.f, 0.f); glVertex3f (0.f, 2.f, 0.f);
glTexCoord2f (-TMMAX45, -TMMAX45); glVertex3f (-10.f, -1.f, 10.f);
glTexCoord2f (0.f, -TMMAX); glVertex3f (0.f, -1.f, 15.f);
glTexCoord2f (TMMAX45, -TMMAX45); glVertex3f (10.f, -1.f, 10.f);
glTexCoord2f (TMMAX, 0.f); glVertex3f (15.f, -1.f, 0.f);
glTexCoord2f (TMMAX45, TMMAX45); glVertex3f (10.f, -1.f, -10.f);
glTexCoord2f (0.f, TMMAX); glVertex3f (0.f, -1.f, -15.f);
glTexCoord2f (-TMMAX45, TMMAX45); glVertex3f (-10.f, -1.f, -10.f);
glTexCoord2f (-TMMAX, 0.f); glVertex3f (-15.f, -1.f, 0.f);
glTexCoord2f (-TMMAX45, -TMMAX45); glVertex3f (-10.f, -1.f, 10.f);
} glEnd ();
```

效果图如图 5.5 所示：

图 5.5 三角形扇天空

该方法真实度比上种方式高一些，可以感觉到头顶的苍穹的高度。缺点是中心点纹理像素堆积，有些不自然，可看到天空边缘

（3）天空盒技术：天空用一个盒子来建模，经过纹理映射后表示天空。其核心代码为：

```
glBegin(GL_QUADS);
    glTexCoord2f(0, 0);
    glVertex3f(-1000, 1000, -1000);
    glTexCoord2f(0, 1);
    glVertex3f(-1000, 1000, 1000);
    glTexCoord2f(1, 1);
    glVertex3f(1000, 1000, 1000);
```

```
        glTexCoord2f(1, 0);
        glVertex3f(1000, 1000, -1000);
    glEnd();
            ...
```

其效果图如图 5.6 所示：

图 5.6 天空盒（有明显的纹理接缝）

该方法真实度较高，天空的边缘颜色也不会变淡，缺点是纹理接缝处会有明显的痕迹，需要特殊处理。

（4）天空球技术：用一个大球来模拟天空，经纹理映射后可以显示出近似真实天空的效果。

```
    glEnable(GL_TEXTURE_2D);
    glPushMatrix();
    glBindTexture(GL_TEXTURE_2D, texture[0]);
    glTranslatef(Io.x,0,Io.z);
    glRotatef(90,1,0,0);
    glColor4f(1,1,1,1);
    gluSphere(qobj,1000,32,32);
    glPopMatrix();
    glDisable(GL_TEXTURE_2D);
```

其效果图如图 5.7 所示：

图 5.7 天空球

该方法真实度最高，不足之处是相对所需绘制时间最长，球的极点出会产生纹理聚集现象，纹理接缝也存在。在本文的系统实现时，我们根据文献[32]的方法，通过改变映射纹理坐标和旋转天空球的方式对其缺点进行修正。

5.2.4 纹理模块

有关纹理的基本知识本文已在第三章论及，下面仅对地形纹理进行概要性的介绍。

地形纹理大概分两种类型：rts 型和 fps 型[33][35]。

rts 型和二维纹理做法相近，把地图分为很多块(Block)，每个块可能又由 m×n 个片组成。每个片使用一张纹理。该方法的优点是图像精细，适合表现细致的场景。缺点是渲染时纹理切换频繁。为了解决这一问题，可以设法做大面积 LOD，LOD 被限制在块内。但是，由于块边缘顶点纹理坐标问题，导致块间没法共用顶点，因而要采用地貌间过渡处理。

fps(frames per second)型贴图，整个地形用一张基本纹理（如:1024*1024）铺满，这张纹理是由程序计算生成（Terrain Texture Generation），适合室外场景。它的优点：贴图将覆盖任意的 LOD 范围，无纹理切换问题。可以把计算好的地形 Map 融入基本纹理，一次渲染。其缺点是图像精度不够，解决方法是在渲染时上面加一层细节贴图。

本文使用第一种方法进行纹理映射，我们对地形高度进行分层设色，将原纹理与地形高度颜色混合，使地形在不同的高度显示不同的颜色。

核心代码为：

```
...
glEnable(GL_TEXTURE_2D);
glEnable(GL_BLEND);
glBindTexture( GL_TEXTURE_2D, texture[1] );
glTexCoord2f(0.0, 0.0);
SET_COLOR(fleftY,h1,h2);
glVertex3f(fleftX, fleftY, fleftZ);
...
```

最终效果图如图 5.8 所示：

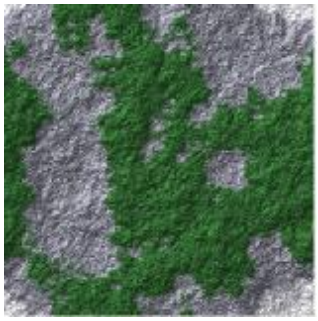
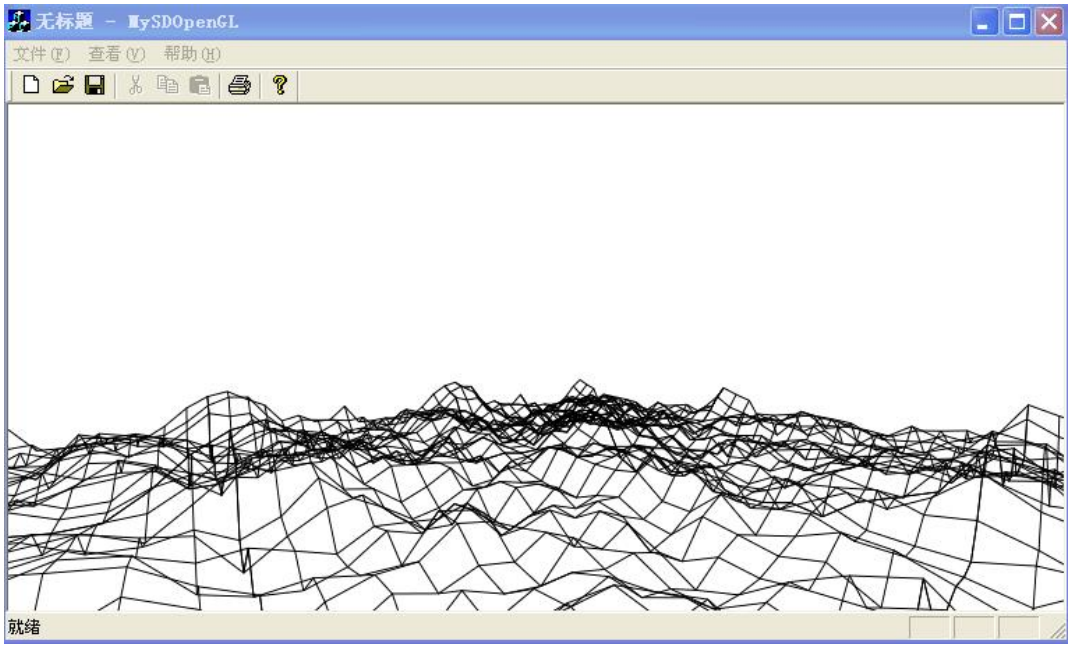


图 5.8 地形纹理图

5.3 实验结果

根据上面各节对系统各个功能模块的描述，形成最终的效果，效果图如图 5.9 所示：



(a)场景格网图

(b)材质和天空之后完全渲染效果图

图 5.9 场景效果图

5.4 本章小结

本章首先介绍了系统的实现平台 OpenGL 在 MFC 下的编程，然后详述了系统的功能模块和系统的流程，介绍了各个功能模块的实现技术。重点介绍了天空模块和纹理模块，介绍了四种天空技术和本文使用的纹理技术。

第六章 总结与展望

6.1 论文总结

生成高度真实感的虚拟自然场景一直是图形学研究领域中一个富有挑战性的难题。作者在本文中针对当前三维分形地形真实感及其实时绘制技术的研究现状，在吸取三维计算机图形学、计算几何、科学计算可视化、虚拟现实等先进理论和技术成果的基础上，对三维分形地形真实感及其实时绘制技术中的分形地形模拟、三维真实感地形生成等核心技术内容展开讨论与研究。围绕该课题本文所做的工作及获得的成果有如下几个方面：

(1)分形布朗运动的数学模型与真实地形特征基本吻合，随机中点位移法中的“Diamond-Square”算法是模拟分形地形的较好方法。并对该地形模型进行了分析和优化。

(2)本文基于 OpenGL 三维图形库和其工作原理，对三维真实感地形生成的基本过程(包括空间变换、光照处理、可见面识别、明暗处理和纹理映射等)进行了详细剖析，基本实现了三维分形地形的真实感绘制。

(3)基于 OpenGL 和 VC++实现了一个小型三维分形地形可视化系统。

(4)分析常用的天空建模方法，采用天空球建模生成本文系统的天空。

6.2 研究展望

本文虽然基本实现了三维分形地形的真实感绘制问题，也取得了初步的成果。但在实际应用中也发现了许多问题。例如，由于网格的细化因子是固定的，因此用于表示模型数据的数组其行列数只能是 Z 系列，否则就无法生成分形对象。在地形过程中，由于采用的建模方式的限制，因此，在进行复杂地形的模拟时，如果数据量过大，就会大大削弱绘制的实时性能等。

关于三维地形模型的研究涉及到多门学科的相关知识，对应的算法比较复杂，对于效率与效果的追求是没有止境的，故对于三维地形的研究还要进一步深入，许多问题的算法还需要进一步的优化，今后的工作包括以下几个方面：

(1)由于分形地形数学模型具有很大的随机性，对地形模型的可控性需进行进一步研究。

(2)本文的纹理贴图还存在比较明显的接缝问题，需要对地形纹理的拼接和多纹理的混合进行进一步研究。

(3)本文地形的绘制过程中没有用到三角形带、三角形扇绘制和显示列表，没有充分利用 OpenGL 的加速功能。

致谢

本论文是在我的导师璩柏青教授的悉心指导和帮助下完成的，从论文的选题、方案制定、工作实施到论文的撰写及修改都得到了璩柏青教授的许多指点和帮助，在此过程中我受益良多。

在近三年的研究生学习中，璩老师认真严谨的治学态度、优良的个人品质都感染并熏陶着我，为我今后的人生道路树立了标杆。在此，我衷心地感谢璩老师给予我的辛勤教导和无微不至的关怀，他不仅传授我知识，也教育我怎么做人。同时，感谢师母夏老师给予我的许多鼓励和关怀。

其次，对参考文献中所列的各文献的作者表示深深地谢意，同时，在整个论文完成的过程中，工业设计系的各位老师、实验室的杨青、邹磊、聂慧峰、付奇等诸位同窗以及我的朋友们给予了我许多帮助和关照，在此，一并向他们表示诚挚的谢意。

另外，感谢论文审阅委员会的诸位老师，感谢你们能在百忙之中审阅我的论文并出席论文答辩会。

最后特别感谢我的父母及其他家人，你们对我的支持和鼓励已无法用语言来形容，我已经取得的以及即将取得的成绩都是为了回报你们的无私付出。

参考文献

[1] 胡少林．基于 DEM 数据的三维地形建模方法研究与实现．国防科学技术大学.硕士论文 2002

[2] 李志林,朱庆．数字高程模型．武汉测绘科技大学出版社,2003,3

[3] 唐泽圣．三维数据场可视化．清华大学出版社,1999

[4] 王永明．地形可视化．中国图像图参考文献形学报,2000,6.5(6):449-456

[5] 龚建华等．地学可视化．探讨遥感学报,1999,8.3(3):236-244

[6] 中国测绘学会地图学与 GIS 专业委员会．地图学与地理信息系统的现状与趋势．测绘通报,1997 年第 6 期

[7] 刘飒．基于 OpenGL 的三维地形可视化技术研究[D]．大庆石油学院，2007

[8] 杨必胜，李清泉，史文中．三维 GIS 中多分辨率纹理模型的研究[J]．中国图象图形学报，2003 年 3 月第 8 卷(A 版)第 3 期:328-333

[9] 龚健雅，朱欣焰，朱庆，熊汉江．面向对象集成化空间数据库管理系统的设计与实现[J]．武汉测绘科技大学学报，2000 年 8 月第 25 卷第 4 期:289-293

[10] 邓念东，侯恩科，张志华等．三维拓扑关系形式化描述及拓扑关系模型研究[J]． 西安建筑科技大学学报(自然科学版)，2007 年 12 月第 39 卷第 6 期:874-877

[11] 齐安文，吴立新．基于类三棱柱的三维地质模拟与拓扑研究[J]．矿山测量，2003 年 03 期:65-66

[12] 曹彤，刘臻．用于建立三维 GIS 的八叉树编码压缩算法[J]．中国图象图形学报，2002 年 1 月第 7 卷(A 版)第 l 期:50-54

[13] 李颖，陈业斌．基于四叉树的三维地形绘制算法[J]．安徽工业大学学报(自然科学版)，2008 年 1 月第 25 卷第 1 期:80-82

[14] 余军．城市三维景观建模与 VRML 浏览的实现[D]．西南交通大学，2006

[15] 朱庆，高玉荣，危拥军等．GIS 中三维模型的设计．武汉大学学报(信息科学版)，2003,28(3):283-287

[16] 王冰等．计算机三维图形常用算法与 C 语言程序．陕西电子杂志社，1994 年 11 月第 1版

[17] 乔林，费广正等编著．OPENGL 程序设计．清华大学出版社，2000

[18] (美)RICHARD S.WRIGHT,JR,(美)BENIAMIN LIPHAK 著．OPENGL 超级宝典．人民邮电出版社，2005

[19] 过润秋，周蕾，林晓春，王敬堂．基于分形法在计算机上仿真自然景观方法

研究[J]．计算机仿真，2005,22(12):155-157,173．

[20] Kruger,J.;Kipfer,P.;Konclratieva,P;Westermann,R.;A particle systemforinteractive visualization of 3D flows,Visualization and Computer Graphics,IEEE Transactions,2005,11(6):744-756

[21] 秦忠宝．用fbm方法生成自然对象真实感图形的研究[D]．西北工业大学．硕士学位论文.2001,04．

[22] 陶闯，林宗坚，卢健．分形地形模拟[J]．计算机辅助设计与图形学学报，1996,8(3):78-185

[23] 李宏达，叶正麟，王小平．分形插值曲面[J]．计算机辅助没计与图形学学报,2002,(4):1-4

[24] 马登武，叶文，成学军．基于分形迭代的地形模拟算法研究[J]．海军航空工程学院学报,2005,20(04):405-409

[25] 齐敏，郝重阳，佟明安．三维地形生成及显示技术进展[J]．中国图像图形学报,2000(4):269-275

[26] 梁俊，王琪，刘坤良．基于随机中点位移法的三维地形模拟[J]．计算机仿真，2005,22(1):213-215．

[27]S.Stachniak,w.Stuerzlinger,An Algorithm for Automated Fractal Terrain Deformation,Computer Graphics and Artificial Intelligence,2005.5:64-76．

[28] F.Kenton Musgrave,Craig E Kolb and Robert g.Mace:The Synthes isand Rendering of Eroded Fractal Terrains[J].Computer Graphics,1989,23(3):41-50．

[29] William H.Press[美]，胡健伟等译．c++数值算法(第二版)[M]．电子工业出版社,2005

[30] 英振华，石锐，胡捷．基于分形理论的三维地形场景的真实感绘制[J]．计算机仿真，2006,23(05):160-162．

[31] 和平鸽工作室．OpenGL 三维图形系统开发与实用技术．清华大学出版社，2003,132-133

[32] 牛凤枝．基于 OpenGL 的三维场景再现研究．河北工业大学.硕士论文

[33] 周永前．面向低空飞行与突防的三维演示系统研制．南京航空航天大学.硕士论文.2007

[34] 李胜睿，王乘恩，王智高等．计算机图形学实验教程(OpenGL 版).北京:机械工业出版社.2004．

[35] 璩柏青，许社教．计算机图形学．西安电子科技大学出版社,2003,1-3

西安電子科技大學

地址：西安市太白南路2号
邮编：710071
网址：www.xidian.edu.cn