哈尔滨工程大学

硕士学位论文

仿真平台基于游戏引擎的展现技术的研究与应用

姓名：魏颖梅

申请学位级别：硕士

专业：计算机软件与理论

指导教师：高伟

201103

摘　要

自然景物的真实感模拟作为计算机图形学的一个重要方面一直就受到极大的关注。通过对不规则物体模拟，人们获得许多逼真、实时的自然景物模拟方法。近些年来，沙尘暴这一灾害性自然现象频繁发生，给人们的日常生产生活带来了严重的后果，引起了社会的广泛关注。论文基于粒子系统方法对沙尘暴这一自然现象进行实时仿真展现，主要研究内容如下：

首先，对国内外沙尘暴仿真现状进行了分析。国内外多集中于后期特效处理以及动画处理上，实时仿真较少。

其次，对游戏引擎影响展现效果的技术进行了分析研究，对不规则物体模拟方法尤其是基于粒子系统的方法进行深入研究。在粒子理论基础上对沙尘信息包括外观属性、运动形式等抽象提取建模，提出沙尘粒子系统模型。

最后，在 OGER 引擎及其粒子脚本等基础上进行详细的实验设计，运用 LOD、布告板以及材质渲染等技术，最终实现沙尘现象在三维虚拟场景中的实时模拟。仿真具有较好的真实感效果并且渲染速度稳定快速。

论文将粒子系统应用于沙尘暴的自然现象的建模仿真，取得了较好的真实感以及实时效果，对沙尘现象的仿真可视化模拟具有重要的实际应用意义，在军事训练模拟、虚拟场景、游戏以及城市规划、公共交通等众多领域都具有实际的应用价值。

关键词：自然现象；不规则物体；仿真；粒子系统；沙尘暴

Abstract

Realistic Simulation of natural scenes as a hot topic of computer graphics has been greatly concerned. Through the researches about that the simulation of the irregular object, people get many realistic, real-time simulation method of the natural scenes. In recent years, more and more researches are carried out on sandstorm simulation; the sandstorm as a type of disastrous phenomenon occurs frequently brings people with serious consequences, and caused widespread concern in society. The real-time simulation that based on particle system method of sandstorm phenomenon is displayed in this paper. The study in the thesis is as follows:

Firstly, the current situation of sandstorm simulation in domestic and foreign has been studied. The researches on sandstorm simulation are mostly focused on post-treatment effects or animation processing, real-time simulation receives much less attention.

Secondly, some display technologies that are applied in the game engine has been studied in this paper. And the simulation of irregular objects especially the particle system and the theory of particle system are researched and analyzed. This paper distills the information about sandstorm including appearance properties and movement format about sandstorm, then gives the sandstorm particle system model.

Finally, simulation of sandstorm phenomena based on the Ogre-Engine and particle script has been designed, and realizes the real-time simulation effect in 3D virtual scene with the LOD, Billboard and Texture rendering technology. Simulation obtained a good realistic effect and its rendering speed is stable and fast.

In conclusion, particle system is used for the simulation of sandstorm phenomena, achieved a good real-time simulation effect. The simulation of sandstorm has important practical significance in real life, especially in military training, virtual reality, game field, urban planning, and public transport and so on.

Keywords: Natural Phenomenon, Irregular Objects, Simulation, Particle System, Sandstorm

第1章 绪论

1.1 引言

人类从蛮荒没有文字，没有语言到用绳结记事，到第一台计算机的出现，经历了漫长的发展。在现代多元化快速发展的社会生活中，计算机已经成为人们生活中不可分割的一部分，计算机技术被广泛的应用于国民经济和社会生活的各个领域。计算机成为人类越来越重要的助手，人类越来越重视及依赖计算机，与计算机相关的各种技术也得到了迅猛的发展。各个学科中应用计算机技术越来越成为一种趋势，凭靠计算机，人类自身得到了极大的解放。作为仿真科学的一个重要发展——计算机仿真技术，正成为应用在各个领域最广泛的技术之一。

传统的仿真技术主要应用于科学计算，来处理科学家和工程师每天面临的海量的数据信息，对传统仿真最后的结果的分析十分的困难，也十分的复杂，而且最终也难以得到高效、直观、形象、整体的仿真结果，无法及时做出相应的判断与决策，所以计算机仿真的高效和结果的形象逼真就显得尤为重要。计算机图形学尤其是三维图形学的发展使得上述问题得到了极大的改善，科学计算可视化、视景仿真技术与计算机图形学的发展是息息相关。现在最热门的虚拟现实技术（Virtual Reality 技术），其底层技术就是三维图形学技术，VR 技术涉及到众多领域，例如高性能的计算技术图像处理，人工智能和模式识别等。同时虚拟现实技术也跨越了计算机领域，被应用到数学、物理、气象、生产生活、心理学等众多和人类息息相关的领域。虚拟现实技术已经发展成为仿真的最高级表现。

沙尘暴是一种较为常见的自然现象，起因于一定的地理环境和气候条件，而近些年来，人类的影响因素基本上已经成为了主要因素。由于人类对能源、资源等的需求导致人类不断的开发自然环境以满足自己的生存需求，由于人口的迅猛增长等等因素，人类对自然的干扰破坏越来越严重。时至今日，已经成为了全球性的环境问题，已经威胁到了人类自身的生存发展。沙尘暴天气的出现标志着环境恶化为题的出现。强沙尘暴天气影响着社会生活的方方面面，造成房屋坍塌，供电受阻，产生不可预料的结果，火灾或者人畜伤亡等，沙尘运动过程中伴随着灰尘化学等物质，在一定程度上加剧了环境的污染，也带来了疾病，影响人类的身体健康，影响农作物生长，直接导致经济财产，生命等的威胁。沙尘暴危害到人类生活的点点滴滴，环境污染，空气质量下降，人身安全受到威胁，生产生活受到影响，沙尘来袭的天气，能见度迅速下降，进而导致交通事故的频发等等。

近些年来，计算机图形领域的热点之一就是对自然现象存在的描述，进而用计算机语言来实现自然现象。通过仿真展现，使自然景观的再现将更具真实性，以此来更好的研究沙尘暴现象对于人类生活的影响。论文通过借助游戏引擎的一些已经较成熟的相关技术来实现沙尘暴这一自然现象在虚拟场景中逼真、具有良好真实感的实时展现效果。

1.2 国内外研究现状

1. 国外发展现状

ID software 公司第一次让人们了解到游戏引擎的相关概念。从创始人约翰.卡马克最先将横版特效应用于卷轴游戏开始，到德军司令部的发行。游戏引擎不断地进化。游戏引擎在渲染方面第一次使用射线追踪技术，根据像素发射光束在行进规程中遇到障碍物反射的原理，游戏程序就是在障碍物处创建单维深度缓存信息（Dimensional Depth Buffer Information），建立纹理图像。然而这时候的引擎还算不上是真正的 3D 引擎，实际上是伪 3D 的。

Doom、Quake 引领时代。Doom（Id 系列引擎之 Id Tech1）问世，Doom 改善了《Wolfenstein 3D》的一些缺陷，并且加入许多新的效果。游戏中增加了可升降的楼梯及路桥效果，增加了立体声系统，而且光照效果不再单一，对真实环境的采样更真实，这些都是 1993 年所取得的进步。ID 公司的引擎之一《Quake》问世，也称为雷神系列。经典的 CS 游戏就是由该引擎开发的。雷神系列引擎的优点在于：该引擎是真正意义上的 3D 引擎，而非以前的伪 3D 引擎。雷神系列引擎首次支持多边形模型、动画和粒子特效技术。雷神之锤 II 引擎支持硬件 3D 加速，以底层图形库 OpenGL 为渲染基础，首次借助新增加的 256 色材质贴图实现了彩色光影效果。彩色光影带来的好处是当游戏世界中物体在发生反射时，能够展现更加丰富多彩的效果。引擎对 DLL 动态链接库的支持，使得软件和 OpenGL 渲染变成了现实。作为 ID 公司的 Tech3 系列引擎，雷神 III 借助 3D 加速卡技术的发展而出现。雷神 III 支持 32Bit 材质、高细节模型和动态光影特效，虚拟游戏世界中的各个实体对象模型所表现的光线效果更贴近于现实，依靠顶点光影技术对模型进行采光，随游戏玩家在场景中的移动而改变实际的造型。Doom3 重新震撼上市——Id Tech4 时代。Doom3 的标志就是即时光影效果的实现，即静态光源下的即时光影，借由 shadow volume（阴影锥）技术，在动态光源下使得即时光影能够在游戏中大规模应用。除此之外，凹凸贴图、多边形、物理引擎和音效技术等也是 Doom3 所具有的特色。游戏引擎技术的发展使游戏玩家能够在虚拟世界中获得更好的体验。

Unreal 引擎引领时代。Unreal 的崛起，Unreal 是 EPIC 公司研发的引擎。Quake/Doom 关注更多的是 3D 图形的处理方面，而 Unreal 引擎涵盖了物理特性、动画演示、音频效果、碰撞检测等众多方面的内容。1998 年 Unreal 第一次发行，该引擎为游戏增加了许

多特效效果，例如美丽的天空，真实的烟雾，微波粼粼的水面。从画面效果上 Unreal 的出众当之无愧。Unreal 支持 DX 规范。引擎的通用性使得 Unreal 被广泛应用于三维建模、建筑设计、特效技术、动作捕捉等众多领域。Unreal2 集成了物理加速技术，借助该技术使得游戏在物理效果处理上获得提高，Unreal2 平台通用性更强。Unreal3 随着 DX9 时代的来临而推出，集百家技术之长，成为使用最广泛的游戏引擎。Unreal3 支持 64 位高动态范围渲染、多种类光照和高级动态阴影效果，以较少的多边形实现只有数百万多边形模型才能实现的高效渲染效果，在节省计算资源的同时也不会降低画面质量。除了上述技术外，对 PhysX 物理引擎、SpeedTree 植被引擎、EAX5.0 音效引擎、AI 引擎等进行了借鉴应用。

同 Unreal 引擎相比，Source 引擎毫不逊色。在该引擎中引入了模块化的思想，该款 3D 引擎具备 Unreal 引擎的大部分技术，从图像渲染到 AI 人工智能，从纹理材质到物理检测。这款引擎至今仍活跃在引擎的舞台上。Source 引擎最让人惊叹的是人物模型的丰富表情，同时与表情配合使用的半自动声音识别系统。

CryENGINE 引擎，算是后起之秀。引擎中使用了“PolyBump”特效进而创造出出色的动态光影效果，减少多边形数目的但是却不影响真实感的表现。

游戏引擎涵盖了游戏设计的方方面面。有通用的引擎也有面向某一方面的引擎。有专门处理图像的引擎，专门处理动画效果的引擎，专门负责碰撞检测的引擎，也有专门处理光影效果、声音处理的引擎，可谓五花八门。现在的引擎则是这些负责某一方面的某些引擎的一个综合体。

因为游戏市场潜力巨大，获得的利润高，同时也由于其他领域对引擎的迫切需求。游戏引擎逐渐自立门户，脱离游戏开发，越来越多的公司积极研发自己的主打引擎，希望能通过这样来占领市场。游戏引擎对于图形学方面的应用，推动了图形学的发展，进而促进了其他学科的发展。

2. 国内现状

一直以来，在国内无论是国家还是公司企业由于种种原因对游戏引擎的重视度不高，导致国内游戏引擎的发展缓慢。当认识到这一领域的重要性的时候，又由于缺乏人才而不得不从国外买入技术，由于不能获得技术的及时更新等，最终导致市场上的众多游戏都是基于国外引擎开发的，这种状况的持续使中国游戏引擎产业一直停滞不前。

由于社会文化的不同，游戏在中国的发展很缓慢，更不用说会有人去研发游戏引擎，而游戏引擎技术本身开发就比较困难。开发前期要求的投入高，风险大，对企业开发的技术实力要求严格，开发需要一批具备高深技能、知识丰富、具备成熟的系统底层开发经验的人员，所有的因素都导致在游戏引擎开发方面的进展缓慢。

标志着国内游戏产业及游戏引擎的发展的就是本土化的“起点引擎”的出现。作为国内的游戏引擎出现的里程碑，该引擎不论是在画面效果还是渲染性能上，都有很出色

的表现。

对于中国游戏产业来说，大幅度提高自主研发能力，是实现产业发展，突破国外市场垄断，具备自己游戏引擎技术的关键。而游戏引擎是重中之重。

国内在游戏引擎方研发方面起步晚，与国外先进技术相比，还存在一定的差距，在游戏引擎研发方面国内只有少数几家公司具有自主研发的能力，如完美时空、涂鸦软件等公司。高校方面对于游戏引擎的研究的投入精力也不够。为此国家在游戏引擎研发上投入大量人力物力，“网络游戏通用引擎研究及示范产品开发”、“智能化人机交互网络示范应用”两个项目正式纳入国家“863 计划”就是一个好的开始。浙江大学的 CAP 小型三维游戏引擎、电子科技大学自主研发的网络游戏引擎是高校关注游戏引擎的开始。2005 年北京市科委拨款自主研发的“GFX-3D”引擎就是一款图形图像处理强大的引擎，包括数据库引擎 OmUtil、图像引擎 Gfx3d、文本引擎 Tfx、多媒体播放引擎 Mfx[1]。

游戏学院和涂鸦软件进行合作，普及本土化游戏引擎的使用，努力培养研发方面的人才，使中国游戏自主研发首先达到“量变”，从而为未来的“质变”打下基础。国内比较知名的游戏引擎自主研发企业也有很多。以完美时空为例，该公司不断积累经验，吸收先进技术，逐渐成为国内游戏产业自主研发的中坚力量之一，自主研发了 Angelica、Cube Engine、EPARC、Raider 等多款游戏引擎，而在明年将要推出的“AX”可以说将自主研发提升到一个全新的领域。金山公司自主研发的 3D 网络游戏《剑网 3》首次采用国内自主研发的 3D 引擎，在画面展现上丝毫不逊色于世界顶尖游戏。

1.3 本文的研究内容

游戏引擎已经发展成为一套由多个子系统共同构成的复杂系统，提供游戏运行所需要的所有功能，然而游戏引擎具体的实现细节则依赖于应用本身，根据应用不同会进行适当的修改以满足应用的需要。

本文主要对以下几方面进行研究分析和学习：

1. 分析游戏引擎影响游戏最终展现效果方面的相关技术，包括如何提高画面质量、计算系统计算、加速绘制过程、实时展现等方面技术的最新发展和应用。

2. 对不规则物体运动，例如自然界中的雨雪、喷泉、烟雾、流水、云、龙卷风等这类自然现象的真实感模拟技术进行深入研究，掌握其理论思想。

3. 对已有的沙尘暴模拟方法进行分析，对沙尘暴自然现象进行分析抽象，从而得到能够用于模拟的信息数据，例如沙尘暴自身的属性，运动规律等。为利用粒子特效系统对沙尘暴进行建模做准备。

4. 进行详细的实验设计，在开发工具上对沙尘暴这一自然想象进行实时仿真模拟展现。

1.4 本文的内容结构

本文的组织结构如下：

第1章，绪论部分，对课题背景、研究意义以及国内外研究现状进行介绍，列出本论文的研究内容和论文的结构。

第2章，研究分析游戏引擎中与最终真实感、实时性展现效果息息相关的各种技术以及经典算法。

第3章，研究粒子系统对不规则物体运动的模拟方法。重点是基于粒子系统的不规则物体模拟方法。

第4章，对沙尘暴现象进行分析抽象，获得沙尘暴属性以及运动规律，对该现象应用粒子系统进行建模，并且应用相关技术，使仿真更加真实，绘制速度更加快速。

第5章，进行具体的设计与实验，在对沙尘暴信息的抽象提取的基础上，运用游戏引擎相关技术实现沙尘暴效果的虚拟仿真展现。

第 2 章 游戏引擎相关技术分析研究

本章主要对游戏引擎中涉及到对效果展现起到影响的相关技术的分析研究。游戏引擎作为一套渐趋成熟的系统，不同的技术支持着不同功能的实现，而所有技术的支持不过是为了最终能够使游戏正常运行，尽量在画面质量、游戏响应速度、真实方面实现最大的平衡。本章重点主要是分析研究以下几方面，包括碰撞检测技术、纹理映射技术、可见性判断与消隐技术以及层次细节和布告板技术的原理等。

2.1 游戏引擎相关技术综述

游戏引擎已经成为多个子系统共同组成的复杂系统，这些子系统分别负责实现特定方面的功能。例如，对碰撞进行检测的碰撞系统，对特效进行处理的粒子系统，对声音进行管理的声效系统等等。引擎系统成为游戏开发的各个环节不可缺少的重要部分，每一部分，每一个子系统充分发挥自己的功能，为最终的玩家带来流畅的动画、精美的画面、壮观的场景、震撼的特效的享受之旅。

“引擎”其实只是一个形象的比喻，可以理解为电脑游戏软件中的某些核心程序的集合。而引擎的主要任务是负责控制游戏世界中的各种实体，包括人物、建筑物、特效效果等，对世界中的静止的运动的物体进行碰撞检测、计算物体不同时刻处于世界中的位置，及时将他们绘制到屏幕上并且配以正确的声音，及时对玩家的命令进行反馈。简而言之，游戏引擎就是游戏的大脑，发出命令控制所有游戏功能的主程序，引擎就是一个大框架，里面有效地管理各种功能的子模块。

引擎以一定的方式对各个子系统、各种游戏资源进行调度、分配、管理，各个子模块在游戏引擎中充分发挥自己的作用，相互配合最终使得游戏不仅在速度同时也能够在画面、特效效果上满足玩家需求。

2.2 碰撞检测技术

碰撞检测[2]即判断场景空间中的两个对象是否相交，更确切地说，碰撞检测解决的是检测“是否（if)”、“什么时候（when)”、“什么地点（where)”两个物体发生了接触的问题。“是否 （if)”返回一个布尔类型的值，来告诉程序是否发生了碰撞。“什么时候（when)”来确定物体碰撞的时间。“什么地点（where)”确定物体如何发生接触的。通过碰撞检测人们能够及时发现碰撞，及时获得碰撞的所有相关信息并对碰撞做出反应，而通过碰撞检测对场景中的对象的处理带来的好处是巨大的，碰撞检测使得人们要

模拟的物体能够按照符合物理运动规律而运行，更加贴近现实生活，最后的展现结果页更加真实形象。

碰撞检测在电脑游戏、机器人、虚拟样机、基于物理的模拟（如电脑动画）以及工程仿真领域都有着广泛的作用，尤其是虚拟现实技术以及分布交互式仿真的发展，促进了碰撞检测技术的进一步发展。以在游戏中的应用为例，碰撞检测确保了人类虚拟的稳定的世界的幻想得以维持。它可以避免当虚拟世界里的人物遇到障碍物时穿墙而过的这种不符合现实生活的情况的发生。

碰撞最先应用于静态检测，而随着所应用领域的广泛扩展，对碰撞检测的要求越来越高，现在的碰撞检测能够处理连续运动模型的干扰问题，随之而来出现了基于离散的碰撞检测算法和基于连续的碰撞检测算法，上面是从时间域上的分析。按照空间域类型划分为基于对象空间的和基于图像空间的碰撞检测。

基于对象空间的碰撞检测主要有[3]：

（1）基于空间剖分的碰撞检测。该方法利用一定的规则对场景空间的层次进行划分，划分成小的单元格。然后将对象分配给一个或者多个小单元格，然后对处于一个或相邻单元格的对象进行碰撞检测。常用剖分形式有均匀剖分、八叉树、BSP 树（Binary Space Partitioning）、Kd-tree 等。简单的 BSP 树例子如图 2.1 所示：

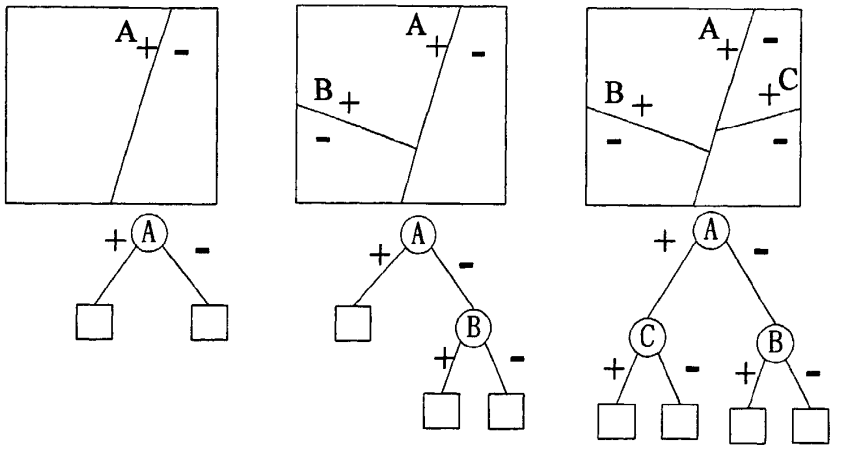


图 2.1 BSP 树剖分

（2）基于 GJK 单纯性的碰撞检测。该算法是以 Gilbert、Johnson、Keerthi 等提出者的名字命名的，第一次成功应用在 1997 年。该算法的基本原理是将场景中要被检测的两个物体当作是一个单纯形，通过单纯形的几何特性计算物体之间分离或穿透距离。Cameron 在原有算法基础上进行改进，提出 GJK 增强算法，以更好的满足实时性要求。2004 年，Bergen 等人开发的 SOLID，也是基于 GJK 的思想。

（3）基于包围盒的碰撞检测算法[4]。算法基本原理是用简单的几何体来包围位于场景中的复杂的模型对象，然后通过对包围盒的相交测试来判断两个物体是否相交，如果包围盒相交，则表示场景中的两个物体发生碰撞，然后再进行详细的相交性检测。否则

表示，场景中的物体并没有发生碰撞，进而保持了自己原有的属性。基于包围盒的算法是使用比较广泛的算法，基于包围盒的算法也有很多种，包括层次包围球树、AABB（Aligned Axis Bounding Box）层次树、OBB（Oriented Bounding Box）层次树、k-dop（k Discrete Oriented Polytope）层次树、QuOSPO（Quantized Orientation Slabs with Primary Orientations）层次树、凸块层次树、混合层次包围体树等。

（4）基于距离场碰撞检测算法。距离场可以作为一种表示图形的模型，表示距离物体表面的最小距离。距离场是一种矢量场，带有正负的方面，对于一个封闭的物体来说，正可以表示对象的外侧或里侧，用负表示对象的里侧或外侧，这种表示不需要物体对象的拓扑和约束信息，加快了碰撞检测的处理，距离场也有多种空间划分方法。

（5）基于随机碰撞的检测方法。例如太空中的两个星球体发生了碰撞，但是相关学者不关心是否发生了碰撞，而是碰撞的结果是什么。在虚拟场景中两个物体类似这样的情况可以用随机碰撞的检测算法进行处理。随机碰撞检测的核心问题是如何进行采样（可以是组成物体表面的点、边或是三角面片）以及如何在采样的特征中进行快速精确的检测。

除了上述算法以外，还有基于智能优化技术的碰撞检测、连续碰撞检测等。

而基于图像空间的碰撞检测算法来说，没有做具体的分类。基于图像空间的方法具有一定的优点，但是也存在一定的缺陷。基于图像空间的方法实现一般比较简单，且对计算机图形硬件的利用高效，近年来，基于图像空间的方法伴随着计算机硬件尤其是图形硬件的发展进入了一个快速发展的新阶段。

2.3 纹理映射技术

传统的几何造型只能从大概的方向上来描述物体对象，对于要模拟的对象的微小细节描述不详尽，而这些微小的细节发挥着重要作用，能使事物表现出更加真实的视觉效果。

纹理映射技术是实现物体高度真实感的有效技术，同时运用纹理映射技术也可以合成某些特效效果。

Catmull[Catmull1974]最早提出纹理映射技术，1976 年 Blinn 和 Newell 将这种纹理映射的概念应用到了双三次差值绘制的茶壶模型中。如图 2.2[5]所示，从几何学原理分析纹理映射。

纹理是就是物体表面包括花纹、颜色、图案等信息提取的图像，包括二维、三维纹理。而所谓的纹理映射其实就是把这些纹理映射到三维物体表面的技术。本质就是以纹理图像作为输入，定义纹理与物体之间的映射关系，根据该映射关系将图像映射到简单景物几何形态上表现具有真实感的表面花纹、图案和细微结构。

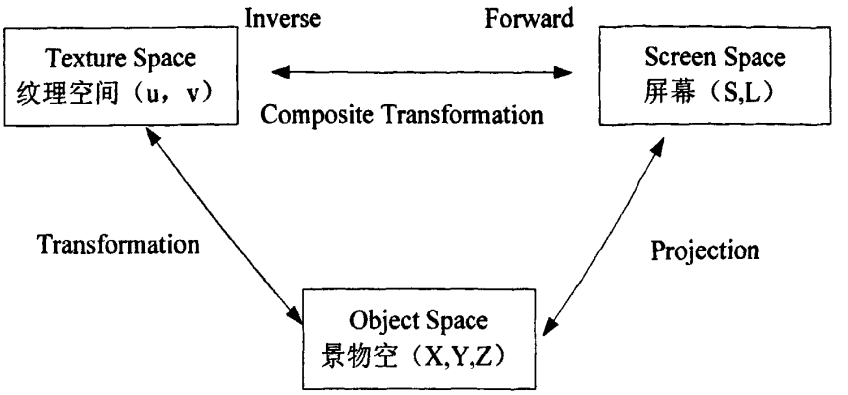


图 2.2 Texture-mapping Geometry

一维纹理映射可以用来模拟等高线或轮廓线，为线段或曲线创建颜色模式。二维纹理映射使用最为广泛，它将图像映射到几何对象的多边形表面上，但是二维纹理映射本身仍存在一些问题[6]。二维纹理缺少三维的信息，对于一些粗糙、凹凸不平的表面，二维纹理并不能正确的进行模拟。二维纹理映射到物体表面时会产生失真，同时如何保存从一个物体表面到另一个物体的纹理的连续性。二维纹理在这方面的效果并不理想，为了解决这些问题，可以在三维空间内直接定义纹理，三维纹理坐标用（u,v,w）表示，既可以体纹理，从而可以直接得到物体表面点的纹理像素值，也可以用第三个分量表示沿投射方向的深度，但难在体纹理获得困难。

纹理映射根据作用分为以下几类：用于改变物体表面颜色和花纹；用来改变物体表面几何属性（例如可以产生凹凸不平的效果）。

纹理映射的基本原理，在 Texture mapping 过程中，存在两个操作对象，即物体和图像，通过映射函数建立物体和图像的点到点的对应关系，纹理映射所要做的就是根据映射函数，为物体上的点找到图像中与之对应的点，并将图像上的点的颜色值赋给物体上的点。

纹理映射的步骤，首先确定物体上某个点作为纹理映射起点，然后进行参数化[7]，物体表面上点的纹理坐标的产生是纹理映射最基本的问题，直接影响纹理好坏；从参数化空间变换到图像对应点位置，取出该点的颜色值，可以对该颜色值进行某些变化计算。最后将值根据纹理融合函数，修改物体上点的颜色属性。

纹理走样是纹理映射技术中常出现的问题。纹理走样包括 Moire 效果、锯齿、模糊、水纹、亮点等。MIP 为纹理走样的主要解决办法，也是最有效的办法。MIP（Multum In Parvo）意即一个纹理映射出不同大小的多个纹理，MIP 技术在内存中保存一系列不同分辨率版本的纹理，来保证像素尺寸和纹素尺寸大致相等。比如一个精度为 128×128×128 的纹理，会有一个 256×256×256，64×64×64，…，1×1×1 的纹理版本，这些版本就叫做 MIP 纹理映射。

目前已经有了很多成熟的纹理映射技术，包括凹凸纹理映射、位移映射、立方体环境映射以及比较高级的光照映射。其中凹凸纹理映射又分为浮雕型凹凸映射、环境凹凸纹理映射、法向映射（为凹凸纹理映射中最常用技术）等。凹凸纹理映射的基本思想对物体表面法向进行干扰，使光照明计算发生变化。位移纹理映射在模拟凹凸的表面细节的同时，同样模拟微观的几何扰动，缺点是图形硬件不支持逐像素的位移映射。环境映射主要是模拟某个物体周围环境。光照映射则模拟光源对物体表面光亮度的影响，极大地节省了实时的光照明计算。如表 2.1 所示，从多方面纹理方法进行了比较。

表 2.1 各类纹理映射技术的特点比较

映射	阴影	遮挡	反射	改变几何	实用性	图形硬件实现
位移映射	无	可以	无	可以	弱	困难
光照映射	可以	可以	可以	无	强	简单
常规纹理映射	无	无	无	无	强	直接支持
凹凸纹理映射	可以	无	无	无	强	简单
球面环境映射	无	无	可以	无	中等	可以
立方体环境映射	无	无	可以	无	强	直接支持

纹理映射过程就好比是为了模拟一个现实世界中的物体，所以先要把骨架建立起来，然后为其穿上色彩艳丽的衣服、为其装饰使之能焕然一新的饰物一样等等。

2.4 可见性判断和消隐技术

可见性判断技术以及消隐技术的产生是为了在满足整个虚拟场景高度真实感的同时又能满足模型复杂度降低已达到实时绘制的目的。对于面前的场景，由于人的视线具有方向性，所能看到的东西具有局限，同时又因为场景中物体之间存在相互遮挡的原因，人眼所能看到的只是小部分场景。而决定这部分能够为人眼所见即为可见性判断。消隐与可见性判断是共同存在的，所谓的消隐，即剔除视点外看不到的部分。可见性判断可以减少场景中实际绘制的多边形数量，不仅提高了绘制时间，同时也减轻了 CPU、图形硬件等的负担。

可见性判断算法从处理层次上将可见性判断分为三大类[8]：物体层、顶点层、像素层。

（1）基于物体层的算法

该方法利用场景的有效组织使得位于视域体内的物体的归属性能够很快判定。典型算法是 BSP（Binary Space Partitioning）二叉空间分割树，简称二叉树。算法将场景组织成一棵树，利用空间连贯性加快场景的遍历，能够很好的减少绘制过程的复杂性。

缺点：基于物体层的方法以面作为划分基本单位，导致产生了大量新的面片，增加了场景面片数、多边形数量。

（2）基于顶点层的算法

基于顶点层的算法有以下三类[9]，其中最简单的算法是背向剔除，如图 2.3 所示：

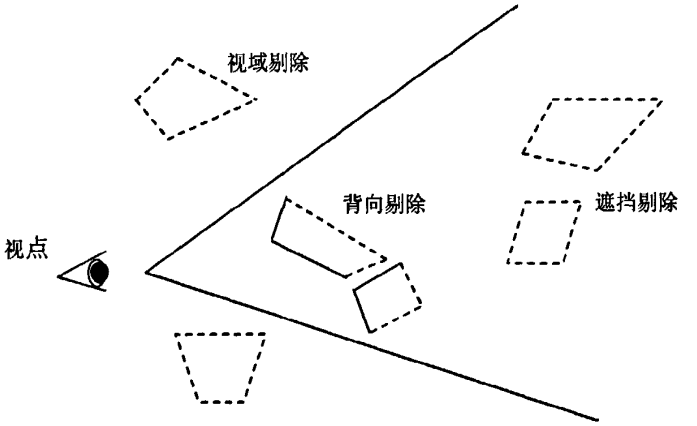
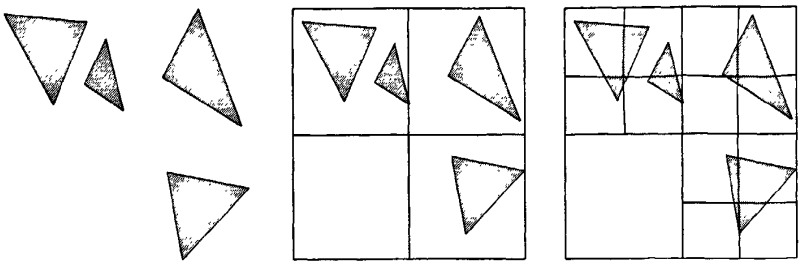


图 2.3 基于顶点的可见性剔除分类

在上图中，由视点出发的两条线限定的区域为视域范围。背向剔除假设前提有两点：单面绘制多边形，且场景物体封闭。显然，如果多边形的法向是背向相机的，那么多边形不可见。图中的“背向剔除”处，虚线表示的面片是背离视点的，所以剔除。“遮挡剔除”处的物体被“背向剔除”处的物体遮挡，所以剔除被遮挡的物体，这部分属于遮挡剔除。“视域剔除”所代表的物体在视域外，对于观察者来说，不可见，所以剔除，这部分属于视域剔除。

（3）基于像素层的算法

基于像素层的经典算法是基于深度缓冲（z-buffer）的算法。算法利用屏幕上同一像素的深度信息决定可见面的计算，缺点是没有利用空间的相关性。为此人们进一步提出了在图像空间建立深度缓冲的四叉树层次结构以及在物体空间建立八叉树层次结构的算法。四叉树例子如图 2.4 所示：



（a）场景多边形　（b）第一次剖分　（c）第二次剖分

图 2.4 四叉树示意图

在物体空间建立八叉树层次结构的例子如图 2.5 所示：

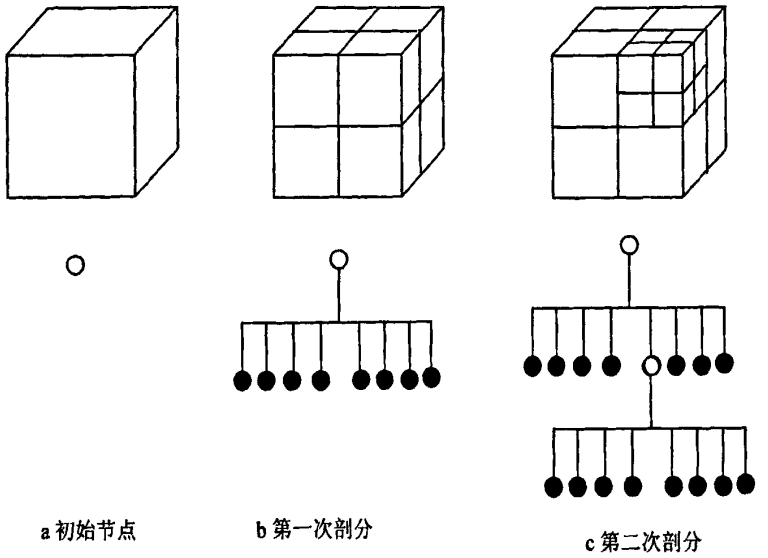


图 2.5 八叉树划分示例

可见性判断和消隐技术加快了场景的绘制速度，满足了实时绘制的要求，对于用户的请求能够做到及时的响应，从而给用户带来更好的体验效果。

2.5 层次细节技术

层次细节即 LOD 技术。LOD（Level of Detail）技术的提出就是为了解决场景或复杂物体（多边形数目）的复杂度进而提高渲染绘制速度。最先提出该技术的是 Clark，他为简化采样密集的多面体网格物体的数据结构而设计的算法。

LOD 的基本思想是为场景中的同一个物体准备具有不同细节程度的多个模型。当物体在场景中比较重要同时物体在场景中占据大面积时，那么选择细节程度较高的模型进行绘制显示，当物体在场景中并不是很重要而且所占据的空间也不多时，可以选择细节程度不高的模型，切换到该模型进行绘制显示。

LOD 算法包含 3 方面的内容[10]：LOD 生成、LOD 选择、LOD 切换。

1） LOD 模型生成

运用多边形简化网格方法来解决 LOD 模型的生产。基于多边形网格的算法主要有 2 类[11]：基于几何特征识别、小波变换的算法。两个算法各自有各自的优缺点，根据实际需要进行选择。还有其他的分类方法，以拓扑结构可划分为拓扑保持型、非保持型；以简化机制可划分为自适应细分型、几何元素删除、采样等；按照简化范围分为全局性算法和局部算法。视点相关性、实时性、误差可控性都可以作为分类的依据。

比较著名的非结构化多边形网格简化算法为 Renzen 的体简化（Volume Decimation）方法。典型的 LOD 模型生成算法有：

（1）Rossignac 和 Borrel 的多面体简化算法。算法流程如图 2.6 所示，优点是适用于任何输入模型，简化速度高效。缺点是不保持原模型拓扑，导致最终的模型视觉效果差，而且会产生不能控制的误差。

（2）Schroeder 的顶点删除法，算法流程如图 2.7 所示。该算法的计算量小，时间复杂度为线性，能够对细节、拓扑结构进行很好的保持，缺点是经过多次迭代后误差逐渐变大，导致模型质量不佳，影响最终效果。

（3）基于包络网格的简化算法。该算法是 Cohen 提出的，基本思想是基于几何结构控制简化过程。算法对模型的拓扑结构、基本特征尤其是棱角都能够很好的保持，对于全局误差能够有效的进行约束，支持自适应的近似。缺点，包络网格难于构造，对于输入模型有一定的限制。

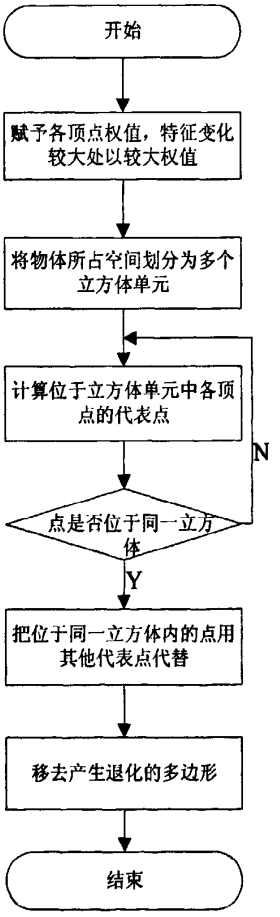


图 2.6 多面体简化算法

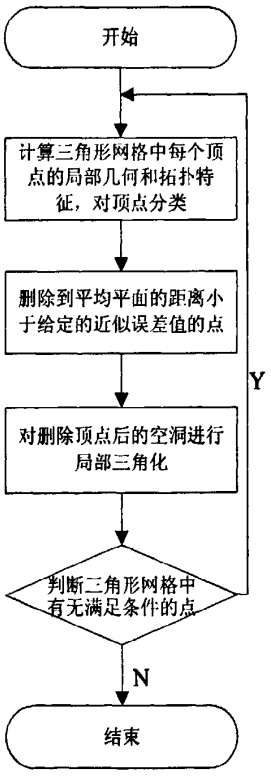


图 2.7 Schroeder 的顶点删除法

（4）重新布点法。算法的基本思想是为新模型生成一定数量的顶点数，该顶点集合可以包含原模型中的顶点，然后按照一定的规则将这些点分布到曲面上；在原网格模型上插入新加入的点，构造新网格；删除得到的模型中的不在新点集合中的点，即得到由新点生成的简化模型。主要应用于光滑曲面，但是计算量和误差较大。

（5）渐进网格法。算法优点在于从一定程度上保持了细节层次的连续性，也对面片色彩、纹理信息给予考虑。缺点是算法执行效率低。流程图如图 2.8 所示：

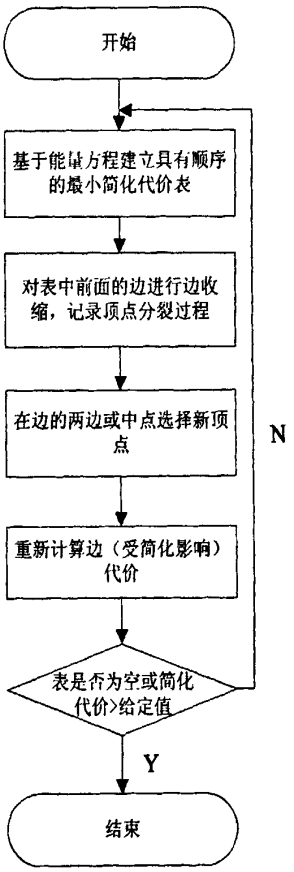


图 2.8 渐进网格法流程描述

（6）基于顶点对的迭代收缩法。算法思想是将一对顶点收缩到一个点，数学表达为$(v_i,v_k)\to v$的过程。移动v_i,v_k到v；用v_i代替所有v_k顶点；删除v_k以及所有退化的面。该算法是目前产生简化模型速度最快的算法，最大的优点是消耗内存少。

（7）视点依赖的场景简化模型。因为人的视觉的局限性，所以在一定时间内，人眼所能看到的物体只是场景中的一部分物体，即场景中的物体落入视域内的只有小部分，而这一小部分在视域内还要考虑到物体重要性以及位置。所以依据这些情况，可以将整个场景中的物体分成多个小空间，而当视点移到一定的空间是，即显示这个空间内的物体，同时依据物体的重要性，离视点远近不同绘制不同的物体。离视点近，在视域

内占有重要作用的物体可以采用细节层次较高的模型绘制，其他根据具体情况选择具有不同细节层次的模型来表达物体。在虚拟场中实际是根据摄像机的移动来判断的。基于视点依赖的场景简化在实时绘制方面比较突出。

2） LOD 模型的选择

LOD 模型的正确选择不仅关系到模型是否能够及时绘制，也直接影响到最终会给用户带来什么样的展现效果。LOD 模型选择有几下几种方法：基于人眼的视觉关系、物体空间位置关系、物体运动特性和帧的频率。

人眼辨别物体的能力受到以下几个因素的影响：物体离人远近、物体大小，所面对的情况不同，人眼看到的细节也为之不同，所以可以根据这点来选择，这就是好基于人眼的视觉关系的选择方法。

世界场景中，有些物体被其他物体所遮挡，不能出现在人的视域范围内，这部分成为眼睛看不到的部分，也是可以不必绘制的细节部分；物体距离观察者的欧氏距离越远，所获得的细节信息越少，则选择精细程度不高的 Model 来表示。这就是基于物体空间关系的选择方法。

观察者通过场景中物体运动速度的快慢，所观察到的信息也随着不同，这就是基于物体运动特性的选择方法。

恒定、稳定的帧速率对于交互极为重要，一旦帧速恒定，则相应的 LOD 细节层次模型也选定。

3） LOD 模型的切换

在具有不同细节层次的模型之间进行选择，然后依据需要展现不同模型。这个过程涉及到 LOD 模型的切换，切换直接影响展现的效果。这种情况会导致一种称为“突越”的突变模型的替换过程[12]的产生，如表 2.2 所示，为 LOD 切换算法比较。

表 2.2 LOD 切换算法比较

切换算法	突越现象	计算量	实现	存储量
连续 LOD 和几何变形 LOD	不明显	很大	很复杂	少
Alpha LOD	不明显	一般	复杂	大
混合 LOD	一般	大	一般	大
离散几何 LOD	很明显	少	简单	大

LOD 层次细节技术在减少场景多边形数量，降低场景绘制复杂度的同时，也合理的控制场景绘制的帧频率。当场景的帧率过低时，可以采用 LOD 算法减少细节层次。反之，当帧率很高时，意味着可以使用更高精度的模型。在速度与绘制效果之间达到平衡。LOD 技术在减少场景复杂度方面至关重要。

2.6 布告板及替代物技术

布告板技术以及替代物技术是从保持流畅渲染的同时，减少场景中多边形数目，降低场景复杂度，从而加快渲染速度方面出发进而发展的技术。

2.6.1 布告板技术

Billboard[13]技术的应用原理就是通过对一些看起来具有三维效果的二维对象进行渲染，来代替传统的方法。该方法是最简单的一种替代技术，Maciel 和 Shirley[14]最早提出用布告板技术来加速模型绘制速度。公告板是一种依赖摄像机确定方向的简单四边形，即公告板的角度随摄像机位置的调整变化同时变化。利用这一原理，事先将能够表现物体外观纹理的图像贴图到四边形上，代替物体，而不必再用物体的实际模型绘制。当观察者观察方面变化时，布告板随着转向观察者方向，即始终保持与观察者保持一致。

例如，在模拟树木的模型中，常用的方法并不是用 3D 模型来表现树木，而是用看上去像一个 3D 树木的 2D 图像作为纹理映射到一个四边形上，当观察者靠近时再用 3D 模型代替 2D 图像来表现树木。如果用户向北观察一个映射到多边形的树图像纹理时，如果此多边形是朝向南的，那么这些树就正好能被正确的观察。而当用户开始沿着一个圆弧向东移动，这些树仍然朝向南方，因此当移动时，用户对于树就会有一个观察角度。这样，树多边形在观察者的视野中就会逐渐变窄，由于用户是从多边形的正侧面进行观察的，直到最后当观察者位于正东时，树将会消失。

自然现象的模拟可以借助布告板、Alpha 纹理及动画技术相结合实现，如火焰、爆炸、云等。树木的渲染常采用该技术，如图 2.9 所示：

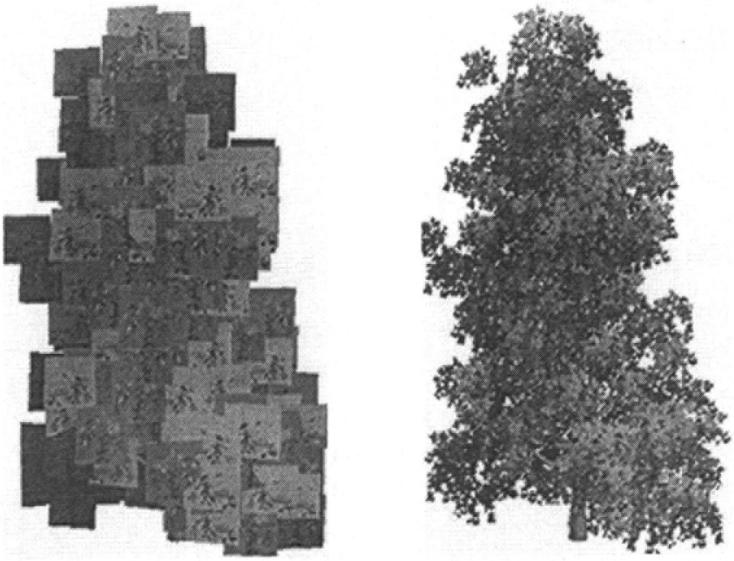


图 2.9 多布告板树以及效果图

四种最常用的 Billboard 几何表示方法如图 2.10 所示[15]：

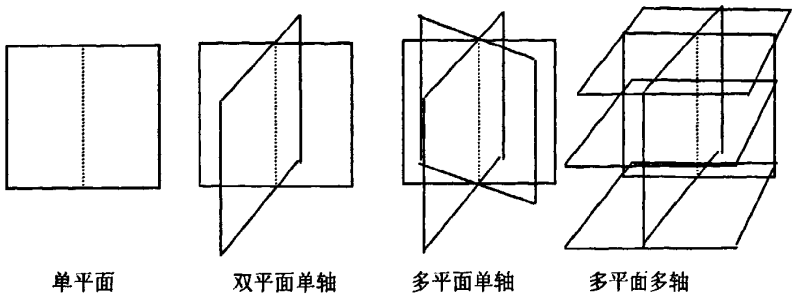


图 2.10 billboard 表示方法

单平面、双平面单轴和多平面单轴方法允许视点的变化在物体高度相同的区域内。最后一种方法视点可以在物体上方或下方，该方法广泛应用于飞行器模拟游戏中。

根据布告板四边形法向和向上的向量设置的不同分为以下几类：

1. 平行屏幕的布告板方法

该方法是最常用最经典方法，方法使作为布告板的四边形法向与视线方向重合。这种技术常用于显示圆形物体和粒子。

2. 平行物体的布告板方法

平行物体的布告板方法主要应用于除圆形物体外的其他物体。定义物体的向上方向为相机向上方向，法向仍平行于视线方向。

3. 视点朝向的布告板方法

前两种方法的缺点是对于透视投影变形效果的模拟起不到很好的作用。在相机的视角和真实人眼的视角相一致时，或视角和 Billboard 较小这种情况下，可以忽略变形效果，取布告板法向的方向平行于视线即可。否则，取布告板法向方向为布告板中心连线至视点，即为视点朝向的布告板技术，如图 2.11 所示：

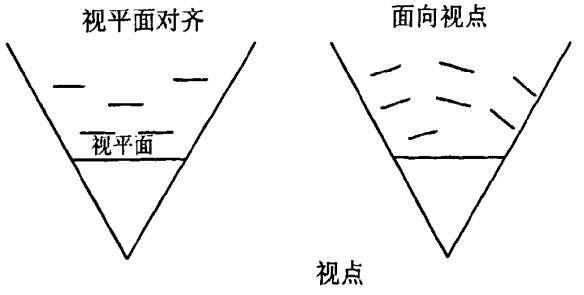


图 2.11 平行屏幕和视点朝向的布告板技术

2.6.2 替代物技术

Billboard 技术的应用原理就是通过对一些看起来具有三维效果的二维对象进行渲染，来代替场景中复杂物体的模拟。这样做的后果就是一旦相机靠近用布告板模拟的物体，就会导致失真现象的出现，因为只是用简单的几何和不变的纹理来代替原物体。为了尽可能大的弥补这种损失，可以使用 Impostor 即替代物技术。其原理就是利用游戏每帧间的连续性，用图像来代替物体本身。Impostor[16]在渲染消耗上非常低，非常适合表示一定视域内的群体模型的替代表示。替身图技术其实是纹理映射技术的一个扩展应用。

2.7 本章小结

作为多个子系统构成的复杂系统，游戏引擎涉及到各种技术。这些技术包括碰撞检测、纹理映射、可见性判断与消隐、层次细节、布告板和替代物技术，在视觉上给人以更真实的感觉同时，同样使画面完美更接近于现实世界，更重要的是加速渲染的速度，满足用户对于画面实时绘制的要求，进而得到一个运动流畅，其表现和移动在视觉上可信，屏幕上看起来逼真的虚拟三维世界。

第 3 章 不规则物体展现技术研究

不管是在游戏引擎中还是在虚拟现实中，都有一些很特殊的效果需要用某些特效技术来实现，就是自然景物，例如雨雪、风沙、烟火等，一般从不规则物体的运动方面进行研究，从而对自然现象进行仿真展现。本章主要对不规则物体的几类模拟展现技术进行研究。重点是研究怎样利用粒子系统原理对不规则物体进行建模展现。

3.1 不规则物体的展现技术

对自然现象中不规则景物的运动效果展现模拟在计算机图形学和动画中是难点和热点问题。不规则模糊物体的模拟比较困难原因有一些几点[17]：

首先，不规则物体多变模糊的外观，可谓千奇百态。自然景物的外观轮廓多为模糊的，没有固定的形状，在消亡之前一直处于动态变化中，并不能用一定的模型去总结概括出来，自然界有很多这类现象，例如火焰、极光、暴风雨、龙卷风、云、雾、雨、雪、沙尘、流水、浪花瀑布等等，这类现象并没有固定的形态外观，不规则性、随机不确定性导致对其进行模拟展现。即使用传统几何学和图形学方法去模拟，所表现出来的真实感效果也难以令人满意，尤其是对软硬件的要求更是制约了真实感的表现。

其次，不规则物体复杂的运动规律也很难把握。导致自然景物的不规则运动，究其根本原因就是物体的复杂运动。不规则物体受到外力作用也睡增加运动的复杂性。在计算机的世界中展示这些不规则物体运动产生的现象成为国内外学者研究和探索的重要课题。不规则物体的展现对于现实世界具有重要的意义，天气环境、自然现象对于人类的影响一直处于一个非常重要的地位，而对自然现象的仿真可以使人类更好的去理解，发现其中的规律，更好的改善生活。自然景物模拟展现的应用前景非常广泛，从电影特效到虚拟战场，从建筑设计到航天航空以及游戏都起到很重要的作用。

近些年来，对于不规则物体展现的研究取得了很多显著的成果，提出了一些关于烟、火、云、流水等等不规则物体的展现方法。具有代表性的方法包括：基于分形理论、纹理、数学物理模型的展现方法以及基于粒子系统的展现方法，根据需要，有时候会同时使用几种方法的结合。

（1）Fractal——分形理论是法国数学家 B.B.Mandelbrot 在 70 年代中后期首先提出来的[18]，并运用这种概念来模拟雪花和海岸线等自然想象。Ray Smith 等人在上述基础上将 L 系统应用到计算机图形学，L 系统渐渐的成为分形图形的分支之一。在实际应用中 L 系统[19, 20]主要应用来模拟织物的形态生长特点，用一定的规则表现植物的生长过程记忆伴随着生长不断改变的外观形态。国内外在以上研究的基础上又不断进行的研究，

提出很多新的方法理论。

（2）基于数学物理模型的展现方法

基于数学物理模型的模拟方法来自流体力学理论的应用。1995 年 Stam 等[21]提出用扩散过程来表现烟雾等气体现象的运动过程，之后，Stam 又通过稳定的半拉格朗日[22]方法来模拟三维流体的运动，能够实时绘制的效果。一般把一些不规则物体看作是流体，通过 Navier-Stokes 方程来对流体的形状外观和运动规律进行描述。Navier-Stokes 方程用一些物理量来解释一些气体随时空变化的过程，这些物理量包括粒子的浓度，扩散的速度，温度以及辐射性能等。根据不同的假设和流动条件可以简化 NS 方程，比如不可压缩的流体方程、无粘性流体方程等。

1. 连续性方程（质量守恒方程）

假设条件：流体密度为 ρ，速度矢量为 $u=u(u,v,w)$，时间 t，直角坐标系下流体的连续性方程一般形式为：

$$\frac{\partial \rho}{\partial t}+\nabla\cdot(\rho v)=0 \tag{3-1}$$

当 ρ=常数时，方程简化为下列形式：

$$\nabla\cdot u=0 \tag{3-2}$$

式（3-2）该方程可以用来表示不可压缩流体的运动。

2. 动量守恒方程

假设条件：f 为质量力密度函数，σ 为应力张量。ρ 为流体密度，$v=\frac{\mu}{\rho}$ 为运动的粘性系数，p 为压力，μ 为牛顿第二粘性系数。

$$\frac{\partial u}{\partial t}+(u\cdot\nabla)u=f-\frac{1}{\rho}\nabla p+v\nabla^2 u \tag{3-3}$$

如果流体没有粘性，则方程为

$$\frac{\partial u}{\partial t}+(u\cdot\nabla)u=f-\frac{1}{\rho}\nabla p \tag{3-4}$$

可以用式（3-2）和（3-3）表示不可压缩粘性流体的控制方程；而用式（3-4）表示不可压缩无粘性流体的控制方程。

基于流体力学理论应用的模拟是计算机图形学的热门问题之一。而基于数学物理模型的模拟方法一直占据着重要的地位，因为这些方法可以提供更让人信服的理论依据以及达到满意的模拟效果。

该模拟方法的优点是模拟效果真实感强，常常用于科学计算可视化等高端应用方面。缺点是计算量极大，对于计算机资源的软硬件要求较高。

（3）基于细胞自动机的展现方法

细胞自动机[23, 24]（Cellular Automata）也成为元胞或单元自动机，是时间和空间离散的数学模型。1991 年，Pakeshi 首次将细胞自动机概念引入到火焰的模拟方法中，将火焰看成是由众多简单的组元以复杂的组合形态和行为构成的集合，不能简单地用数学来描述。Pakeshi 从物体的温度、燃料密度、气体流向三个属性通过一些状态转换规则来构建一个细胞单元。通过对细胞单元值的改变来表现动态变化过程。Dobashi 等人根据细胞自动机原理实现了云的动画模拟。2001 年，SeongGyu Jeon 等人通过细胞自动机实现了爆炸模型。现在，细胞自动机模型不断的得到发展和完善，在很多学科都得到了应用。例如，将细胞自动机模拟应用在流体力学、分子力学、图像处理等领域。

（4）基于纹理的展现方法

纹理映射通过将纹理图片（可以是任何图片）“贴到”光滑物体表面来展现物体丰富的细节信息，以此来使物体的表现根据真实性，增加物体真实感的同时也减少了建模的复杂度，提高效率，但是，基于纹理的方法在表现动态不规则的物体，尤其是自然景物方面难以达到很好的效果，人们总是能发现人工痕迹的存在。但是由于纹理映射技术巨大的优越性的存在，促使人们不断的去寻找可以应用的方法，而人们意识到不规则物体的随机不确定性所产生的纹理也具有随机性，同时这些随机产生的纹理又具有一定的自相似性。基于这些以及随机过程理论，提出了过程纹理模型，该模型可以为特定的景物产生特定的纹理，以此来模拟不规则物体。

1985 年，Peachey 和 Perlin 将三维纹理场景和景物表面求交，将两者的交作为纹理的属性，然后对景物进行一定的处理，然后通过映射将纹理空间的纹理贴到景物表面。为了使效果更真实，Perling 运用湍流函数，对函数施以扰动的参数对已经获取到的纹理进行调整获得更多细节的纹理，并且将该方法模拟火焰和云等不规则物体。1989 年，Inakage 基于该技术对火焰效果实现了二维模拟。Arno Schodl[25]应用视频纹理技术，实现了对瀑布、风吹过草地场景的模拟展现。2004 年，林夕伟等[26]将纹理方法同粒子系统方法结合，对火焰燃烧进行了模拟。2007 年,王继州[27]运用粒子系统的思想，抛弃传统的模拟方式实现了用纹理图片直接对火焰等进行模拟的方法。主要是将纹理作为场景中的粒子，然后不断改变纹理片的大小，颜色，透明度及位置以及纹理片的叠加等获得了很好的火焰模拟效果。

尽管纹理映射技术在某些方面还需要进一步完善，但是由于该技术本身具有的特点，使得纹理映射技术能够不断的得到发展，应用也会越来越广泛。

除了上述介绍的比较经典的模拟方法之外，还有其他的模拟方法，例如基于光照模型的模拟方法、基于扩散过程的模拟方法等。如表 3.1 所示，列出了不规则物体模拟比较常用的模型的比较。

表 3.1 不规则物体模型比较[28]

模型	对象表示	运动表示	适用环境	计算代价	真实感
纹理映射	实体纹理函数	纹理参数变化	真实感要求不高	一般	较差
光照模型	密度集	光线变化	科学计算可视化	极大	良好
分形模型	分形集	分形参数变化	静止图像	不一定	静态逼真
物理过程	温度、风、密度场	NS 方程	科学计算可视化、高端应用	极大	强
扩散过程	粒子烟团	扩散方程	可视化，高端应用	极大	强
细胞自动机	细胞网格	状态转化规律	运动图像，中低平台	不一定	一般
粒子系统	粒子图元	随机过程	运动图像，多平台	不一定	良好

3.2 基于粒子系统的模拟方法

3.2.1 粒子系统概述

粒子系统的方法在众多领域都有应用，而在由不规则物体构成的自然现象的仿真方面的应用已经成为目前为止公认的最有效的仿真方法，同时这些应用也促进了粒子系统理论的发展和完善，现在有很多基于粒子系统理论提出来的对自然不规则景物的模拟。

采用形状简单的微小粒子作为粒子系统的基本元素是粒子系统建模方法的理论基础。这些微小粒子可以是点、小立方体、小球等。

1983 年，Reeves 第一次提出粒子系统概念，并且对火焰以及爆炸现象进行模拟。该粒子系统描述为是一个由一系列相同或不相同的动态变化的点，即粒子，粒子独立受力，粒子之间相互不影响的这样一个粒子集合系统。现在，集合的元素不只可以是点，也可以是球体、椭球体、立方体，甚至可以是纹理片，这些都可以作为粒子系统中的单个的粒子元。在流体力学的基础上，通过控制粒子元的数量、颜色、运动等来动态的模拟一些流体性质的自然不规则景物等。

对于粒子系统的分类，Tonnesen 在总结了前人工作的基础上，将粒子系统分为以下三类：独立粒子系统、固定连接的粒子系统、动态耦合粒子的系统[29]。所说的独立粒子系统是指粒子之间相互独立不影响。通过对粒子几何施以一定的受力场，例如风力场、电场等，来使粒子产生类似自然界中运动的复杂运动，从而模拟真实的自然景物。固定连接的粒子系统指的是粒子及其邻近的粒子之间有一定的影响。主要用来模拟给定材料的变形。而动态耦合的粒子系统，是指粒子集合间的变化影响随时间而变。基于动态耦

合的粒子系统最显著的应用是对于群体及群体运动变化的模拟。

不规则自然景物的模拟技术主要有两类。分别基于质点系和流质系粒子系统[30]。质点系如烟雾、火焰等有动态变化、不连续、模糊的边界。流质系如流水、岩浆，这一类的粒子系统具有明确、连续且动态可变的边界。

3.2.2 粒子系统技术原理

1） 基本思想

粒子系统是将要模拟的物体或景物看成是由众多粒子构成的集合。然后定义组成粒子系统的每个粒子颜色、生命周期、位置以及速度等自身所会具有的属性以及在粒子系统中可能的运动规律。存在这样一个假设前提：粒子集合由有限粒子组成，粒子有自己的属性，粒子的运动互不干扰，在这样的前提下，将定义的粒子的集合在场景中按照一定的规律（在一定的生命周期内，不停产生、刷新、更新这样的循环运行，直到展示结束为止）进行展示，为了使效果更真实，可以利用随机过程进行控制，使模拟空间内的粒子的运动、自身属性都具有一定的随机性，以此达到不规则物体的运动效果。通常在场景中的每个粒子都会经历一个产生、更新到消亡的一个生命历程，如图 3.1 所示：

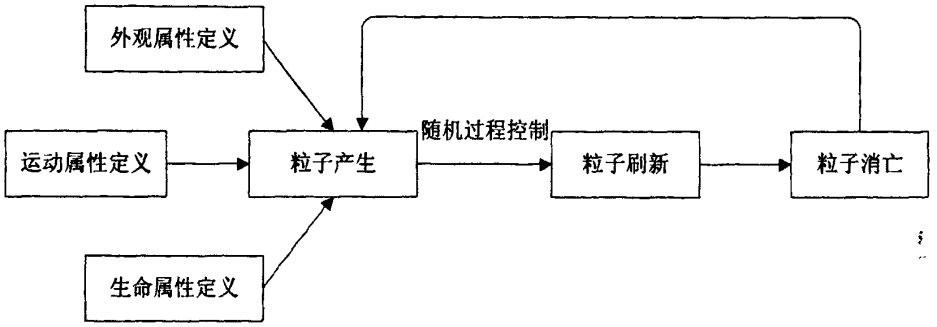


图 3.1 粒子在场景中的生命历程

粒子经历从产生到消亡的不断循环的过程，从而来模拟不规则物体随机的、动态的效果。

粒子的外观属性，包括粒子数量、颜色、大小、位置、透明度、形状，这些属性是同时间相关的；运动属性包括速度，加速度；粒子的生命属性主要是用来表示粒子的生命。粒子系统构成模型如图 3.2 所示。

2） 粒子系统数学语言描述

对粒子系统应用数学物理学原理进行分析表述为公式。

定义 3.1 粒子 粒子用属性构成的集合表示，这些属性对应实数域上的一个 n 维向量，n 为粒子属性的个数。

$$Particle^n = \{attr_1, attr_2, \ldots, attr_i, \ldots, attr_n \mid n \geq 3, n \in I\}$$

$attr_1, attr_2, \ldots, attr_i, \ldots, attr_n$ 代表依据模拟对象抽象出来的各种属性作为粒子的属性

值，通常包括形状、大小、透明度、速度、加速度、生存期等等，即物理规律下所表现出来的属性以及生命周期等。

定义 3.2 映射关系 粒子个体到正整数集的映射，每个粒子有一个索引，表示为 I_t 到 P^n 的映射：

$$G(t)=\{Particle_t:I_t\rightarrow Particle^n \mid I_t\subset J,n\geq 3,n\in I,t\in R\}$$

$W(i)=Particle^n$ 为索引 i 的粒子的性质和状态。

定义 3.3 粒子系统描述 映射关系下构成的有限集合：

$$S(t)=\{G(t)\mid t\in\{t_0,t_1,\ldots,t_m\}\}$$

S 为系统在时刻 $t_0,t_1,\ldots,t_m$ 的粒子状态集合，$S(t_0)$ 为粒子系统初始状态。

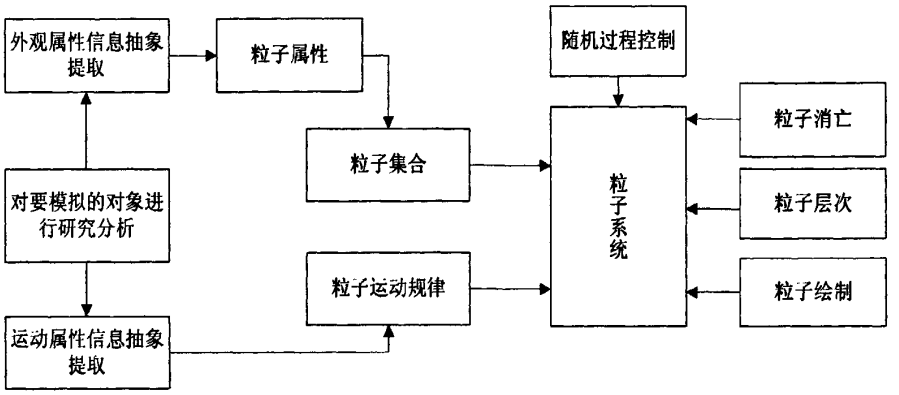


图 3.2 粒子系统构成模型

3） **粒子的生成**

用随机过程控制粒子的产生，主要有几方面：粒子数目，包括初始粒子数和新绘制粒子数；时间对场景中粒子密度的影响；产生空间，可以是球面、平面等。

粒子的产生形式有两种：

1. 平均粒子数法。*MeanParts* 表示每帧画面粒子平均数量，定义数量的最大变化范围 *RanParts*，则 j 时刻粒子实际数目为：

$$NParts_j=MeanParts_j+Random()*RanParts_j$$

Random() 为一个产生-1 到+1 之间随机数的一个随机过程。则每一帧粒子数在区间 $[MeanParts-RanParts,MeanParts+RanParts]$ 随机变化。

2. 面积粒子数度量法。粒子生成数量的取值依据每一帧在屏幕上占据的显示面积而定。不同时刻计算一帧所占的面积，然后确定粒子数量。表示为：

$$NParts(f_j)=(MeanParts(f_j)+Random()\times RanParts(f_j))*ScreenArea$$

$MeanParts(f_j)$ 和 $RanParts(f_j)$ 为第 f_j 帧屏幕面积所需粒子数的平均值和方差。$MeanParts(f_j)$ 可为常数也可以是变量。*ScreenArea* 为一帧所占的显示面积。

在烟雾的模拟中，烟雾还涉及一个过程，即烟雾的浓度过程，从淡到浓再到淡，为了实现类似的这种效果，可以控制每一帧产生的粒子数平均值来模拟，获得一个变化的

粒子数的平均浓度。可以通过式（3-5）获得平均浓度：

$$MeanParts = InitMeanParts + DeltaMeanParts \times (f - f_0) \quad (3\text{-}5)$$

也可以通过式（3-6）获得平均浓度：

$$MeanParts = InitMeanPartsPerArea + DeltaVarPartsPerArea \times (f - f_0) \quad (3\text{-}6)$$

InitMeanParts 和 *InitMeanPartsPerArea* 代表初始时刻粒子平均数，*DeltaMeanParts* 和 *DeltaVarPartsPerArea* 代表粒子平均数的变化速率，f 系统当前帧，f_0 初始时刻第一帧。

4） **系统粒子初始化**

粒子属性的初始化通用公式：

$$\mathrm{Pr}operty(f_0) = \text{属性的均值} + \mathrm{Random}() \times \text{属性方差}$$

（1）**外观属性初始化**

沙尘粒子的外观属性包括大小，颜色，形状以及生存属性等。

1. 初始大小。粒子大小的选择要平衡真实感和计算消耗，粒子的大小在确定粒子形状时就能够被确定下来。

2. 颜色初始化。粒子的颜色根据对要模拟的对象信息抽象得到。由 R,G,B 值和透明度来确定粒子颜色。初始值如下：

$$InitColor(R,G,B) = MeanColor(R,G,B) + Random() \times RanColor(R,G,B)$$

透明度为：

$$InitColor(Alpha) = 1.0$$

3. 形状初始化。点是最简单的形状表示，计算机中为一个像素。线段、多边形、球体、纹理面片都可以表示粒子形状。可以根据具体需要在衡量计算代价的同时进行选择。

4. 生存属性初始化。由生存期来决定粒子在场景中存活的时间，时间单位可取值为帧数或系统时钟周期。

（2）**运动属性初始化**

粒子初始速度和大小定义，大小确定如下：

$$InitSpeed = MeanSpeed + Random() \times RanSpeed$$

MeanSpeed、Ran*Speed* 为粒子平均速度和随机变化范围，方向由模拟的对象决定。

（3）**运动状态的改变**

场景中粒子的运动起到重要的作用，粒子的随机运动是模拟自然不规则物体现象的重要表现形式。每时每刻，场景中的粒子所处的位置和速度都不同，同时由于要更加真实的模拟，所以通常会引入风场、重力场等，粒子的运动就会按照一定的轨迹运行。空间中运动一般可以描述如下：

$$V = V_0 + \int a dt \quad (3\text{-}7)$$

$$P = P_0 + \int v dt \tag{3-8}$$

任意时刻粒子的速度为式（3-7）所示，任意时刻粒子的位置变化为式（3-8）所示。V_0为初始速度，a为加速度，P_0为初始时刻粒子位置，v为速度，粒子受外力产生加速度，运动方式随之改变。

5） 粒子的消亡

粒子在运动一段时间后，需要将其移除屏幕空间，在系统中删除。判断生命是否结束可以通过几种方法。一是以帧数作为粒子的生命周期的度量，则可以通过帧数的递减情况来判断。二是通过粒子属性生命值来进行判断。例如生命值为 1.0，该值随粒子运动递减，当达到小于 0 时，粒子生命结束。三是为粒子颜色、透明度设置阈值，当运动粒子的颜色和透明度超过阈值时，粒子的生命结束。

每一个粒子都会经历了从产生到消亡的整个动态的运动过程，整个粒子群的运动的综合很好的对不规则物体进行了模拟。

6） 粒子的渲染绘制

粒子绘制同粒子自身形态密切相关，主要分为以下几种：点粒子的绘制、面粒子的绘制、线性粒子的绘制、随机粒子的绘制[31, 32]。绘制同时利用光照、浓淡处理配以消隐技术来实现粒子绘制。

3.3 本章小结

本章主要是对粒子系统进行分析，阐述了粒子系统相关的理论知识。从对模拟景物信息的抽象提取，到粒子的产生、属性定义、运动状态的定义到粒子消亡等几个方面介绍了粒子系统基本模型的架构。第三章是论文第四章对要模拟的自然现象进行信息抽象的理论基础，在这个基础上对模拟的物体进行抽象，获得展现想要的信息，最终转化为计算机能够识别的信息，绘制到计算机屏幕上。

第 4 章 基于粒子系统沙尘暴模拟

本章在前几章的基础上，运用粒子系统理论对沙尘暴进行建模。对沙尘暴现象进行抽象分析，包括沙尘颗粒的外观属性、运动属性等等，以及如何运用一些相关数学物理知识建立沙尘暴的运动模型等。并且从真实感、实时性方面综合考虑如何运用 LOD、布告板等游戏引擎相关技术使沙尘暴效果展现更贴近现实世界。

4.1 沙尘暴自然现象分析

4.1.1 沙尘暴定义

定义 1[33]：沙暴（sandstorm）和尘暴（duststorm）的总称，是指强风将地面大量沙尘物质吹起并卷入空中，使空气特别混浊，水平能见度小于 1Km 的严重风沙天气现象。

其中，沙暴为大风把大量沙粒吹入近地层所形成的挟沙风暴；尘暴是大风将大量尘埃及其它细粒卷入高空所形成的风暴。

定义 2：沙尘暴也称沙暴或尘暴[34]，指的是强风将尘沙吹起使空气混浊，水平能见度小于 1 公里的天气现象。

4.1.2 沙尘暴外观属性

1. 颜色外观

沙尘暴出现时，黄沙随着风运动漂浮在天空中，浑天暗日，浓密的沙尘铺天盖地，遮住阳光，能见度非常低。风沙墙上层因为沙尘稀薄颗粒细小，阳光几乎能穿越沙尘射下来的原因常显黄至红色，中层呈灰黑色，下层因为沙尘浓度大，颗粒相对较大，阳光几乎全被沙尘吸收或散射，所以呈黑色。有的时候也会因为组成物质或区域不同而有所不同，更为人们熟知且典型的是黄沙遮日外观颜色。

2. 形状外观

沙尘暴刚刚形成时，是朦胧的黄沙，没有一定的形状轮廓，给人的感觉更像是烟雾，而当沙尘暴在不断的移动过程中，密度逐渐增大，形成一面壮观的风沙墙。

3. 运动状态

沙尘能够运动并且大范围的移动，主要因素在于强风的作用。强风时，靠近地面的空气很不稳定，下面受热的空气向上升，周围的空气流过来补充，使空气携带大量沙尘上下翻滚不息，形成无数大小不一的沙尘团在空气中交汇冲腾。

4.1.3 沙尘暴动力学原理

沙尘暴的发生步骤分为起沙尘、扬沙尘、沙尘输送和沙尘沉降几个过程。使沙尘能够在空中移动的风速值叫做临界风速，该值取决于沙尘颗粒大小、植被覆盖等。沙尘的移动形式为以下几种：滚动、跃移和悬移[35]。

1. 滚动。在地表层，一些较大的沙尘颗粒因其重量过大，不能离开地面进入风中随风运动，只能在地表滚动，并逐渐的在当地沉积。

2. 跃移。直径在100~500*um*的中等颗粒，在强风推动下短时间内进入风中，在离地面一定高度内移动，受地球重力影响时落回地面，沙尘颗粒相互之间发生碰撞并产生新的运动。其他类型的输送最终原因是跃移运动的结果。

3. 悬移。由于跃移和直接风力作用，直径D<100*um*的尘粒被大风刮起来，随风在空中输送，受大气环流影响运动到更远的距离。

本文对沙尘暴的三种运动方式跃移、悬移、滚动等进行分析，从而得到沙尘颗粒运动的简化模型，主要考虑的是沙尘的悬移运动。在游戏引擎以及虚拟现实方面对于沙尘暴的模拟展现所做的工作，尤其是在实时仿真方面所做的工作很少，基本上都是基于后期特效处理（Post Effect）在渲染前用 Shader 进行处理而成的。

2007 年，浙江大学 CAD&CG 国家重点实验室的王章野副教授曾用基于物理动画[36]的方法实现了沙尘暴的模拟，在 2008 年用雷诺平均两流体模型来模拟大气二元混合物[37]，在模拟系统中运用流体力学进行分析，将运动的两种流体，即空气流（风）和粒子流分别用两个不同的 Navier-Stokes 方程描述，并且通过引入一个相互作用力来表示两流体之间的交互作用，而且还考虑到了光的散射效果的影响。最终在一个框架平台内第一次实现了对龙卷风、沙尘暴以及火山云等实现了模拟，该模拟是在 GPU 基础上实现的对自然景象的展现，其他对于沙尘暴的研究主要集中在游戏、电影中的特效效果的处理上，而且是通过动画后期渲染等技术实现的模拟。

4.2 沙尘暴粒子系统建模

在第三章的关于粒子系统的理论分析中，论文已经给出了粒子的一般属性，包括外观属性、运动属性、生存属性等。

一个沙尘粒子可以通过下面方法进行表达：

定义 4.1 粒子 属性构成的集合，这些属性对应实数域上的一个 n 维向量，n 为粒子属性的个数，$Particle^n$ 为某一个粒子的属性集合：

$$Particle^n = \{prop_1, prop_2, \ldots, prop_i, \ldots, prop_n \mid n \geq 3, n \in I\}$$

定义 4.2 沙尘暴粒子 沙尘粒子外形、颗粒大小、表现颜色、透明度、生存期、位置、速度和加速度，表示为：

$$Partile^n = \{psh, psize, pClr, pAlpha, pLife, position, speed, Accelerate\}, n = 8$$

定义 4.3 沙尘暴粒子集合 定义为上述单独粒子的部分组合，并为粒子分配索引项。作为 $Index$ 到 $Particle^n (n \in 8)$ 的映射

$$MP = \{Index \rightarrow Particle^n \mid n = 8\}$$

对象索引的沙尘暴粒子。$MP(i) = Particle^n$ 为索引 i 的粒子的性质和状态。

定义 4.4 沙尘暴粒子系统定义 沙尘暴粒子集的有限集合，描述为：

$$G(t) = \{M_t : Index_t \rightarrow Particle^n \mid n = 8, t \in \{t_0, t_1, \ldots, t_m\}, I_t \in J\},$$

MP_{tk} 为沙尘暴系统在 t_k 时刻的状态，MP_{t0} 为粒子系统初始状态。$G(t)$ 为时间 t 的函数。

4.2.1 沙尘暴粒子的产生

沙尘暴粒子生成有几个重要部分组成：粒子发射器在什么范围内发射粒子；每次粒子发射器发射粒子数量，该数量影响粒子密度，进而影响效果真实性的表达，包括：初始化的数量和更新绘制时需要为消亡的粒子补充数量。

1. 粒子空间分布设计

为了能够最有效地利用粒子，对粒子在什么的空间按照什么样的发射模型发射等进行简化处理。这里依据一些因素进行改进，主要是从人的视觉特点出发，对于场景中的物体，只有落入人的视线范围内的才会被看到。在视区内，粒子的层次模型比较复杂，而对于在视区之外的粒子由于人眼的因素，看的并不是很清楚，同时，人眼对焦点处看的清楚，对于焦点或视线之外的场景看的模糊。三维场景中，深度信息对于人的身临其境感至关重要，为了加强沙尘暴模拟场景中的深度信息，对于远处的沙尘粒子可令其移动速度更慢一些，根据上述因素可以对粒子层次可以简化，这样可以减少计算的复杂度。视锥如图 4.1 所示：

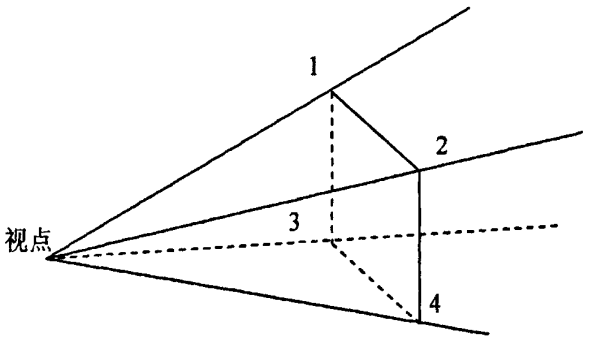


图 4.1 视锥

基于上述因素的影响，粒子的分布区域简化为如图 4.2 所示：

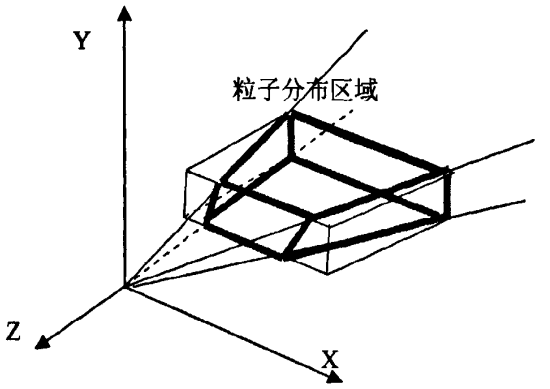


图 4.2 粒子可见区域分布图

图中粗线框包裹的空间即为粒子的分布空间。基于 LOD 细节层次的思想对沙尘颗粒的密度进行分级。离视点近处粒子细节层次复杂，数量多，而对于人眼焦点外，离视线远处的粒子可以将其细节层次进行简化处理，同时，粒子数目可以相对少一些，这样可以控制显示图像的时间消耗。

2. 新粒子数目产生

粒子产生的数量分为两部分，一是初始化时粒子的数量，为了模拟沙尘暴起沙的状态，产生一定数量的粒子，从而减少计算的复杂性；在粒子系统中除了初始时刻以外系统中新的粒子的产生采用物体面积衡量法实现，通过下列公式进行控制，同时为了能够更好地展示真实性随机性，采取随机过程的控制方式：

$$NParts_f = (MeanParts_f + Rand() * Varparts) * ScreenArea / psize$$

$psize$ 为定义的粒子大小，$ScreenArea$ 为屏幕大小，$Rand()$ 为一个均匀分布的随机过程，产生的值在 -1.0 到 ±1.0 之间。$MeanParts_f$ 为第 f 帧平均粒子数，$Varparts$ 为粒子变化的方差。粒子产生数量影响着系统中粒子群的密度变化。物体面积衡量法能够更好的对景物细致程度进行表现。

4.2.2 沙尘粒子的初始化

系统中沙尘粒子生成的同时，自身的属性也就同时确定下来了，这些属性包括前面定义的那些属性。形状、大小、颜色、透明度、生命值、位置、初始速度等。

1. 沙尘粒子的颜色

沙尘粒子的产生时的颜色就是粒子的初始颜色，根据三原色 (R,G,B) 通过下列公式给出：

$$InitColor(R,G,B) = MeanColor(R,G,B) + Rand() * VarColor(R,G,B)$$

$MeanColor(R,G,B)$ 代表粒子初始颜色平均值。每个粒子同时具有透明度的属性，0

代表最暗，1 代表最亮。透明度的取值为 0 到 1.0 之间的任意实数。

沙尘粒子初始颜色为暗黄色，透明度较暗，代表沙尘开始启动，当沙尘暴达到风沙墙的效果时，仍为黄色，但是透明度增大，密度最大。

2. 沙尘粒子初始形状和大小

外观属性对于沙尘暴的真实感的模拟很重要。形状选择不当或粒子数目选择过多或过少，都是影响模拟的真实性。沙尘粒子的形状可以采用点、也可以采用四边形或者小球体等。本文采用四边形，然后将通过 3DMAX 提取的沙尘暴的纹理图片作为沙尘的纹理图。实际模拟中，可以根据视点（摄像机）的远近等因素进而改变粒子的大小。

3. 粒子生命周期

沙尘粒子的生命期为从产生到消亡的时间。系统中并不是所有粒子的生命期都是一样长短。系统初始时刻，为每个粒子赋予 1.0f 的寿命值。并且通过给生命期施加一个衰减干扰值，衰减值不同，每个粒子的生命也不同，从而来表现出沙尘粒子运动的一个动态变化状态。

$$PLife(t) = InitalLife - Time \times Attention$$

PLife 为粒子生命值，*InitalLife* 为粒子初始生命值，*Time* 为系统时间，*Attention* 粒子生命衰减值。

4. 粒子初始速度

沙尘粒子的初始速度方向为面向观察者或者摄像机镜头，大小由下列式子确定：

$$Init\text{Rapidity} = Mean\text{Rapidity} + Rand() \times Var\text{Rapidity}$$

作为粒子系统中的两个系统参数，*Mean*Rapidity和*Var*Rapidity 分别表示粒子平均速度和速度变化的方差。速度方向为始终面向观察者，在粒子系统中为朝向摄像机镜头方向。

4.2.3 沙尘暴粒子的运动模型简化

沙尘暴在扩散的过程中伴随着跃移和悬移同时也伴随着沉积，而正是由于沉积的存在才导致扩散运动的终结，最后大气恢复能见度。整个过程通过扩散方程控制。建立扩散方程的数学模型，然后不同情况的受力不一样来表现沙尘暴现象。

[38, 39]考虑单纯重力场的情况，沙尘粒子运动受重力、浮力、空气阻力的影响，运动方程如下：

$$F_{合力} = G_{重力} - f_{浮力} - f_{阻力} = m\frac{du}{dt} \tag{4-1}$$

沙尘颗粒运动速度变化率为 $\frac{du}{dt}$，m 为沙尘颗粒质量，其中，$G_{重力} = mg = v\rho_1 g$，沙尘颗粒体积为 v，密度为 ρ_1。

重力场中

$$f_{浮力} = v\rho_2 g \tag{4-2}$$

其中流体密度为 ρ_2

$$f_{阻力} = \mu s\rho \frac{u^2}{2} \tag{4-3}$$

μ是阻力系数，s是垂直于颗粒运动方向的投影面积，u为沙尘颗粒沉降速度。

将阻力，浮力，重力代入式子（4-1）中，得到下面的方程：

$$m\frac{du}{dt} = v\rho_1 g - v\rho_2 g - \mu s\frac{u^2}{2} \tag{4-4}$$

该公式是通过物理学的流体力学和数学手段得到的沙尘颗粒扩散的数学模型，表示了沙尘颗粒在空气中沉降速度的变化率及其影响变化率的因素。对上述模型进行分析：

当速度 u 达到某一数值，流体阻力与沙尘颗粒的重力相等，$F_{合}$ 为零，沉降速度降至零；当颗粒运动中受到一个和运动的方向相反、流体速度和沙粒沉降速度相等的气流作用时，这时，沙粒的相对速度为零，沙尘颗粒悬浮在空中，不再下降；当气流作用于沙尘颗粒的力与沙尘颗粒相等、方向相反时，悬浮速度与沉降速度为空气和颗粒的相对速度。上述建立起来的数学模型对沙尘颗粒沉降速率、流体密度及所受阻力的关系进行了很好的表达。从这个模型分析，实际中的沙尘暴的运动成因、以及受力非常复杂。

因为沙尘运动过程比较复杂，在沙尘暴运动过程中，受到很多复杂的力的影响，例如，沙尘颗粒本身的重力、流体阻力、流体浮力，同时沙尘颗粒之间还存在相互作用的力，还有可能受到电场的影响等。所以本文只考虑沙尘粒子在重力场下受风力的影响所做的不规则运动效果的展现，而对其他复杂情况不予考虑，对模型进行简化。

沙尘粒子的运动模型简化为只受到风力的作用，不考虑扩散运动的影响。同时，不考虑粒子之间的相互作用力以及热运动的撞击力。

考虑沙尘暴能够跃移，最关键的就是外力影响。当风力达到一定程度，那么重力的影响可以忽略，从而只需考虑风力作用即可。强风促使沙尘不断的向某个特定的方向移过，风力可以通过下列方程进行控制：

$$Wind = MeanWind + Rand() * VarWind$$

式子中，*Wind* 代表场景中风的作用力，*MeanWind* 代表平均风力，*VarWind* 代表风力的随机变化范围。

在本论文中，考虑到风力的作用只是保持沙尘粒子能够不断的面向观察者，方向为面向观察者即视点方向。随机改变风速的大小，则可以使粒子产生动态变化的效果。

根据牛顿第二定律可知，物体在受到外力的情况下，一定会产生一个加速度，沙尘暴的运动由于受风的作用力大于重力的作用力，所以只考虑风力，而不考虑重力影响，风力产生的加速度为 a，则沙尘粒子的运动公式为：

$$\begin{cases} V = V_0 + \int a dt \\ P = P_0 + \int V dt \end{cases}$$

其中，P 为沙尘粒子在 t 时刻的位置，V 为沙尘粒子时刻 t 的速度，a 为沙尘粒子加速度。粒子的状态变化可以用 Euler 方法计算得到粒子在时刻 t 的位置和速度的变化。

$$\begin{cases} V(t+\Delta t) = V(t) + A(t) \cdot \Delta t \\ P(t+\Delta t) = P(t) + V(t) \cdot \Delta t \end{cases}$$

其中，$A(t) = a(t)$ ，即加速度为时间的函数。$V(t)$ 为上一时刻粒子速度，$P(t)$ 为上一时刻粒子位置。

在三维场景中粒子的运动速度和位置为：

$$\begin{cases} V_x(t+\Delta t) = V_x(t) + A_x(t)\Delta t \\ V_y(t+\Delta t) = V_y(t) + A_y(t)\Delta t \\ V_z(t+\Delta t) = V_z(t) + A_z(t)\Delta t \end{cases}$$

$$\begin{cases} P_x(t+\Delta t) = P_x(t) + V_x(t)\Delta t \\ P_y(t+\Delta t) = P_y(t) + V_y(t)\Delta t \\ P_z(t+\Delta t) = P_z(t) + V_z(t)\Delta t \end{cases}$$

4.2.4 沙尘暴粒子的消亡

沙尘暴场景中的粒子会因为两种情况而需要从系统中删除，一种是粒子的生命期结束，粒子死亡，需要从系统中删除；一种是粒子的寿命并未到达死亡状态，但是粒子的运动使得粒子超出了运动区域，或者粒子的透明度成为零等，也需要从系统中删除。本文从粒子的再利用考虑，当沙尘粒子飞行一段时间后，回收再利用而不从屏幕中删除。

4.3 沙尘暴粒子的绘制

粒子系统中对于场景中粒子的绘制有很多种方法，包括点粒子绘制方法、面粒子绘制方法、随机粒子绘制方法、线粒子绘制方法等。可以根据模拟的不规则物体的特征和实际需要来选择适合自己系统的粒子绘制方法。

4.3.1 点粒子的绘制方法

点粒子绘制方法是最简单的绘制方法，在该方法中粒子以点光源进行绘制。避免景物遮挡而进行的遮挡计算，排除了碰撞时在 z 轴上的深度排序。点粒子绘制减少了对计算量的要求，但同时，为了更逼真的模拟，势必要产生大数量的粒子，导致系统的单独帧的计算量将很大，进而影响模拟效率。点粒子的绘制方法应用于较简单的模型模拟很有效。

4.3.2 面粒子的绘制方法

面粒子绘制方法，是由 Stolk 和 Van Wijk 提出来的[40]，该方法主要是将某些具有能够反射有向光源的微小面片作为粒子，以这些微小面片的几何生成近似的粒子系统。赋予微小的粒子面片几何和时间属性。粒子产生时，属性随之产生。同时给粒子附加法向矢量，用来计算粒子的光亮度。系统事先计算粒子的亮度和颜色，然后经过处理使粒子变换到屏幕空间，经过扫描转换，最后被绘制到屏幕上。让粒子随机或按照一定规则分布在线段长度上、多边形区或实体面区域上。并且按照规则释放粒子，由粒子的密度来决定面的透明度。粒子系统连续释放粒子，产生流线；如果连续地释放线粒子，则能够生成流面。该方法已广泛应用于 3D 游戏开发中。

4.3.3 随机粒子的绘制方法

随机粒子绘制方法主要是为了解决流体中湍流效果的模拟。Hin 和 Post 用粒子的方法模拟的三维湍流场，对湍流现象本质进行分析，抽象出展现想要的信息，进而将湍流描述为由一个向前的运动和一个湍流运动共同组成的一个运动过程。用平均速度场来表示向前的运动，用一个涡流扩散方程表示向前运动过程中产生的湍流运动。为了模拟粒子的随机效果，在模拟中还会加入一个局部涡流扩散率控制的扰动，使模拟效果更真实。这个方法的很明显的一个应用就是水流运动过程中产生的漩涡现象。

4.3.4 线粒子绘制方法

线粒子方法是由 Karl Sims 提出来的。将粒子定义为由一个头和一个尾构成的结构，分别赋予头尾包括位置、半径、颜色等特征。粒子的形状属性由两个圆组成，中间用切线相连，通过线性函数或者高斯函数的递减表示不透明度，获得粒子从中心 1.0 到边缘 0.0 的亮度范围。其他参数通过对头到尾的线性插值得到。在场景进行绘制时，进行粒子头尾和半径到屏幕坐标系的转换，然后将粒子划分为包含颜色、透明度、深度信息的像素基片；然后进行深度排序和隐藏面计算等，最后将粒子显示到屏幕场景中。Sims 等人运用这种方法对闪电、瀑布进行了模拟，获得了很好的效果。线粒子绘制方法，计算量不大，内存消耗少，同时又能够获得良好的自然现象的模拟效果。

4.3.5 本文沙尘暴粒子的绘制

本文从计算量和粒子数量，运算速度方面考虑，沙尘粒子的形状采用四边形代替粒子对沙尘粒子进行绘制。采用四边形从一定程度上减少了场景中点粒子绘制时的粒子数量。并且在实际的绘制中应用纹理贴图和布告板技术进行渲染，即将沙尘纹理“贴图”于四边形上；利用布告板技术使得粒子在绘制时，始终面向观察者。为了表现沙尘颗粒运动边界的不规则性，在绘制时，采用十字交叉贴图来实现这种边界不规则性。对于粒子数目众多而导致的场景复杂度问题，本文通过 LOD 技术进行简化，从而提高绘制速

度。基本思想是根据粒子与观测者距离的远近对粒子群进行密度分级。分为三个层次，即从面向观察者开始，分别为近处、中间处、远处。距离观察者近处的粒子层级可以复杂些，粒子发射的数目越多，远处粒子层次可以简单些，粒子数目可以相对少些，通过这种方法提高对粒子的利用率。距离观察者近处可以用几何方法生成一定数量的粒子，中间部分，利用始终面向摄像机的布告板技术，远处可以直接采用 2D 纹理进行处理。

4.4 本章小结

本章通过对沙尘暴的自然现象进行信息抽象，抽象出沙尘粒子自身的属性，获得了沙尘颗粒的颜色、透明度、形状大小、速度等属性，同时沙尘颗粒动力学分析，获得了沙尘颗粒的运动方式，并且对沙尘颗粒运动模型进行简化获得运动模型。在此基础上实现沙尘粒子系统模型的设计工作，对于粒子的产生，运动，消亡以及绘制方法进行了探讨，应用布告板技术、纹理映射图、LOD 细节层次技术等来更好的完成沙尘暴真实、实时的模拟。

第 5 章 效果展现

5.1 开发工具介绍

1. 环境搭建

本文基于非常著名的图形引擎 OGRE 的粒子系统进行对沙尘暴现象的模拟。使用工具，及平台配置如下：

图像引擎为 OGRE1.7.2 版本；

开发工具为 Microsoft Visual Studio；

开发平台为 Windows XP 专业版 32 位；

处理器为英特尔酷睿 2 双核@2.53GHz；

内存为 2GB。

2. OGRE 功能特性

OGRE（Object-oriented Graphics Rendering Engine）是一款开源的图形渲染引擎，该引擎是用 C++开发的面向对象的 3D 引擎。OGRE 本身具有很多优点：

（1）自动管理透明物体(系统自动帮你设置渲染顺序和深度缓冲)，支持 Billboard，以实现特效；

（2）类库对更底层的系统库 Direct3D 和 OpenGL 的细节进行了抽象，并提供了基于现实世界对象的接口和其它类。支持多平台，包括 Windows、Linux、Mac OS X 等；

（3）简单可扩展的对象框架，能够方便插入到已存在的应用程序框架中；

（4）自动处理渲染状态和空间剪裁、强大的材质管理和脚本系统，可以不用修改一行代码去进行材质维护。支持模型和场景的 LOD 技术；

（5）支持所有纹理混合和绑定技术，支持多种纹理图片格式，包括：PNG，TGA，DDS，TIF，GIF，JPG，支持特殊格式的纹理，包括：一维纹理（1D），容积纹理，体积纹理和压缩的纹理格式如 DXTC；

（6）全面支持并且方便使用的天空盒 SkyBox、天空面 SkyPlane，以及穹顶 SkyDome；

（7）高级插件方式的粒子系统，可扩展的发射器，效果器，渲染器。支持多种阴影技术。全面支持渲染到纹理（Render-to-Texture）技术和投影纹理，材质 LOD（细节层次，Mipmapping）技术；

（8）OGRE 图形引擎支持脚本语言，可以通过脚本对粒子以及相关材质的相关属性进行设置，并且可以随时进行修改，这样大大降低了对程序的依赖性，不必再源代码

进行修改设置等，从而加快开发。

3. OGRE 模块组成

OGRE 是一个庞大而纷杂的对象和模块的集合，每个模块之间相互配合，共同实现 OGRE 的强大功能。OGRE 的模块如表 5.1 所示：

表 5.1 OGRE 模块组成

OGRE 模块	子模块
OgreMain	
PlatformManagers	SDL Win32
Plugins	BspSceneManager、FileSystem、GuiElements、OctreeSceneManager、ParticleFX
RenderSystems	Direct3D7、Direct3D8、SDL
Tools	3ds2oof、3dsMaxExport、BitmapFontBuilderTool、MilkshapeExport、PythonInterface、XMLConverter

其中 OgreMain 为 OGRE 的核心模块，OgreMain 模块的特性包括以下：场景组织体系、Material 管理、插件动态加载系统、数学支持库、渲染器和几何管道、网格/几何实体管理、资源管理、天空/背景渲染、公告板系统和粒子系统、日志和异常处理、事件监听器、编解码和图像加载器、自定义内存管理器、基本动画、骨骼动画、字体渲染/字体加载以及覆盖表面等。

5.2 OGRE 粒子系统下设计思路

5.2.1 粒子系统设计模块

本文将粒子系统分成四大模块的管理：粒子属性管理、粒子喷口设置、粒子集管理以及力场控制等，粒子系统模块构成如图 5.1 所示：

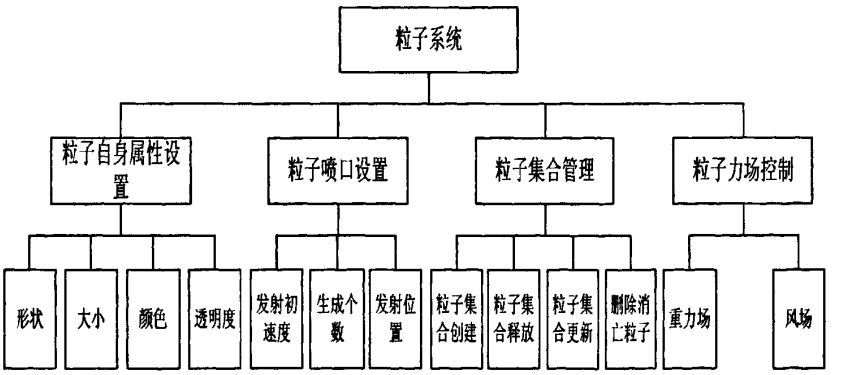


图 5.1 粒子系统模块图

粒子属性管理主要是为系统中的粒子赋予各种能够表现自身真实性的属性；喷口设置主要是对系统中粒子发射的情况进行控制，包括粒子初始速度，粒子初始数量以及更

新时需要补充的粒子数量，最后确定粒子的发射空间分布区域。粒子集合管理主要是对粒子系统运行时，粒子的创建、产生、更新、消亡等进行管理。而为了使粒子展现的效果更真实，为粒子运动施加力场，使粒子产生动态的变化效果。粒子系统模块中各部分分别由 OGRE 粒子系统的粒子系统管理器、粒子发射器、粒子影响器等实现。

5.2.2 OGRE 粒子系统

在 OGRE 的粒子系统中，粒子的属性信息由粒子发射器和粒子影响器共同作用决定。首先粒子发射器（Emitter）产生所需要的粒子。

OGRE 粒子系统包含控制层次如图 5.2 所示：

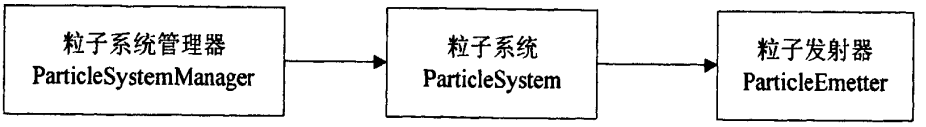


图 5.2 OGRE 粒子系统控制层次

1. OGRE 粒子发射器包含 Point、Box、Cylinder、Ellipsoid、HollowEllipsoid 以及 Ring 类型的发射器。

点状粒子发生器随机地从单一的点发射粒子，而盒状发生器随机的从一个区域发射粒子。Cylinder Emitter 发射器从沿 Z 轴摆放的圆柱体区域内随机发射粒子，深度决定了圆柱体的长度。Ellipsoid Emitter 发射器从一个椭圆体形状的区域里发射粒子。HollowEllipsoid Emitter 类似于 Ellipsoid Emitter 发射器，除了椭圆中心有一个空洞区域不发射粒子。Ring Emitter 从一个环状区域发射粒子。

粒子发射器定义粒子的颜色、形状、大小、生命周期、透明度、材质信息等属性。发射器中也有材质属性的定义，OGRE 同样支持材质脚本，通过材质脚本为粒子设置材质属性，只要将材质脚本模块引入粒子脚本即可。

2. 粒子影响器则负责管理粒子在系统中运行时，随着时间的改变属性的改变的控制，同时为粒子系统施加重力、风力、颜色衰减等效果。由 Linear Force Affector 影响器对所有粒子施加一个力，改变粒子的运动轨迹，可以用于重力、风力、或者其他线性力。

粒子影响器可在粒子存续期间对粒子进行修改。按类型分为：LinearForce 影响器，在所有粒子上产生力的效果。ColorFader 影响系统中粒子的颜色属性。

粒子发射器不断地发射粒子，粒子影响器在粒子运动中修改粒子的属性状态，从而形成最后展现想要的模拟效果。OGRE 同时也提供了光照处理、碰撞检测等更好地实现效果的真实性。

5.2.3 OGRE 优化渲染

OGRE 中最为频繁的渲染状态改变就是材料的变化，多为纹理的变化。为了优化渲染，必须使渲染状态的变化减少到最小。OGRE 支持的 Material 类，该类封装了物体的所有材料属性，强大的材质声明语言允许您在代码外维护材质资源，使得最终能够获得更快更好的渲染效果。

OGRE 一份完整的材质至少有一种技术（Technique）实现，每种技术实现中至少要有一个渲染通路（Pass），如图 5.3 所示。在 OGRE 中渲染的基本单位是通路，同时也是标志物体渲染状态的基本单元，也就是说对象的每个渲染通路将导致一次对图形硬件的绘图调用。OGRE 在所有的技术方案中尽可能选择最优的材质技术。通路对纹理进行了定义，对纹理映射等进行管理，在一个通路中可以多个纹理单元，当然也可以一个也不定义。

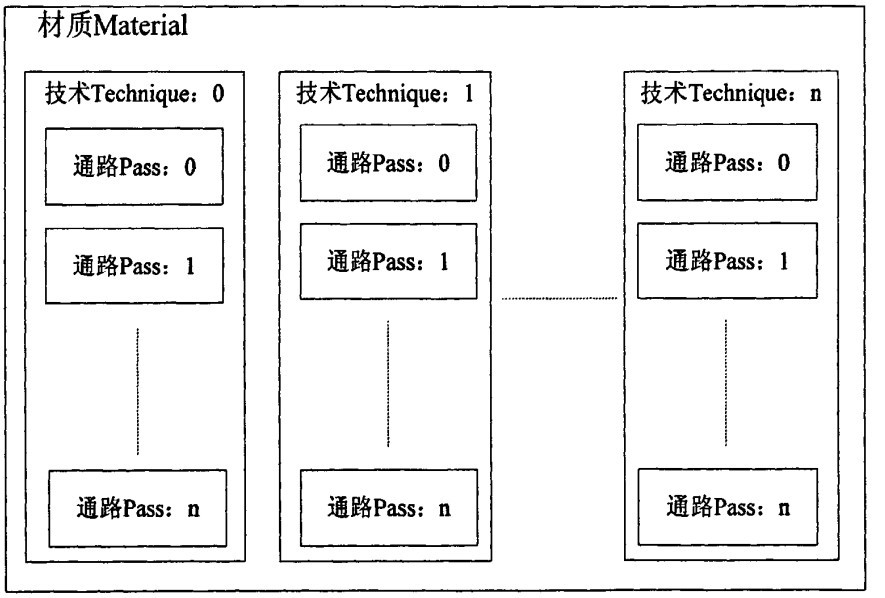


图 5.3 Ogre 中材质，技术以及通路之间的关系

OGRE 的 Material 类封装了物体的所有材料属性，同时也封装了 culling 模式和深度缓存设置等，这些都影响着物体外观，放到同一个类里便于更好的控制。Material 类包括以下几类属性：

1. 最基本的表面材质属性，如对颜色的反射率、shininess 等。
2. 纹理层及纹理的混合和纹理过滤方式。
3. 深度缓冲和 Culling 模式设置。
4. 光照影响控制。
5. Shading 选项和雾化效果。

5.2.4 沙尘效果绘制流程

粒子脚本及材质脚本设置。在粒子脚本中对沙尘粒子属性进行设置包括三部分：一是粒子自身属性，例如粒子的宽和高、材质、布告板类型等。二是粒子发射器属性，包括发射器的类型，在场景中位置，发射角度，粒子生命，发射速率，发射方向。三是粒子处理器处理，在这里给粒子添加重力场和风力场效果，令模拟更真实。

1. 场景建模。用 3DMAX 建立好三维虚拟世界场景中的实体对象模型，保存到资源库中。

2. 构造 Entity。Entity 是场景中渲染的物体的实例，如图 5.4 所示。是场景中重要的部分。例如虚拟场景中的建筑物，可以把它想象成 3D 网格，而 OGER 也正是通过网格来构造这些物体的。灯光、公告牌，粒子以及摄像机等没有 Entity。

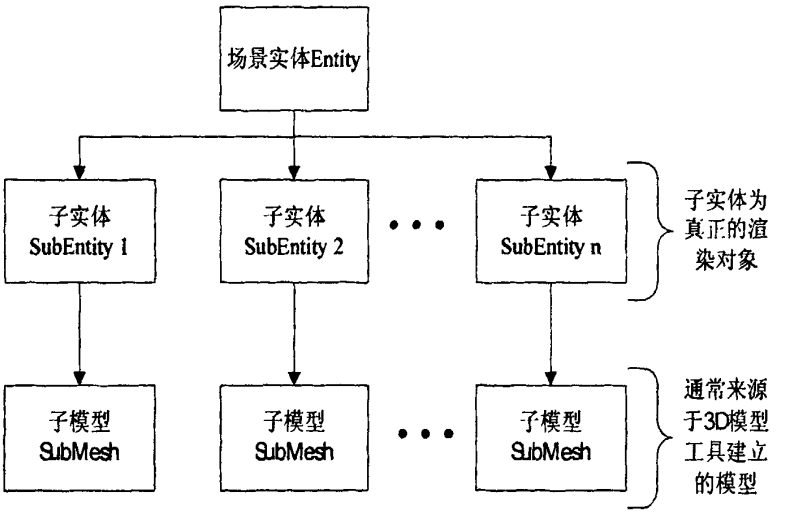


图 5.4 OGRE 场景管理结构图

实现函数：Entity * buildEvent =mSceneMgr->createEntity ("Build", "build.mesh");

3. 绑定 Entity 到场景节点 SceneNode。用网格构造出 Entity，为了使场景管理器能够对实体 Entity 进行管理控制，需要将 Entity 绑定到 SceneNode 节点。

SceneNode *buildNode=mSceneMgr->getRootSceneNode () ->create;

ChildSceneNode ("BuildNode"); buildeNode->attachObject (buildEvent);

4. 场景编辑器对场景进行管理。SceneManager 管理出现在屏幕上的所有物体。对放到场景中的物体坐标通过摄像机进行跟踪。

5. 系统初始化时加载粒子脚本，建立用来查看场景的摄像机，通过摄像机建立视口，使用视口可以设置背景颜色，设置场景灯光以及阴影等。

6. 场景渲染。最后通过 OGRE 渲染将效果实时绘制到场景中并且展现给用户。在场景运行控制过程中，OGRE 负责对场景中的碰撞检测进行处理等以达到更好的真实感

渲染。

5.3 沙尘暴展现效果分析

论文在 OGRE 粒子系统理论及粒子脚本、材质渲染脚本，同时借助 LOD 细节层次技术以及布告板技术等基础上，实现了沙尘暴在三维虚拟场景中的真实感效果展现，获得较好的沉浸感。粒子系统理论的应用展现了不规则模糊物体运动的动态性和随机性。

图 5.5 到 5.8 是论文基于 OGRE 粒子系统理论基础上提出的沙尘暴粒子系统模拟效果。图 5.5 和图 5.6 为自然界最为常见的黄沙漫天的沙尘暴现象的仿真展现，图 5.5 为初始时刻效果图，图 5.6 所示为粒子数目增加到一定程度后表现出来的效果，从图 5.6 中可以看到，场景中的其他物体由于沙尘浓度的变化而导致模糊，从而表现出能见度的下降。图 5.7 和图 5.8 为沙尘颜色属性改变后的效果，图 5.7 和图 5.8 的过程同图 5.5 和图 5.6。实时仿真的平均的帧速率为每秒 24~34 帧。

表 5.2 不同仿真方法的比较

仿真方法	场景复杂度	真实感	帧速率
基于物体动画	复杂	较好	6 帧每秒
基于雷诺双流体模型的二元大气混合物模拟	较复杂	较好	6-30 帧每秒
基于粒子系统的沙尘暴模拟	一般	较好	24-34 帧每秒

同其他沙尘暴方面的仿真相比，这里包括基于物体动画的仿真，基于雷诺双流体模型的仿真，仿真时，在保持真实感效果的同时，在帧速率方面，有了一定稳定的提高。如表 5.2 所示，列出了不同仿真方法的帧速率。

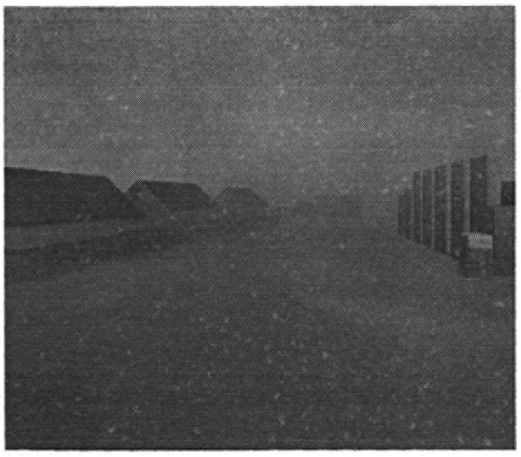


图 5.5 初始时刻效果

图 5.6 粒子浓度增大效果

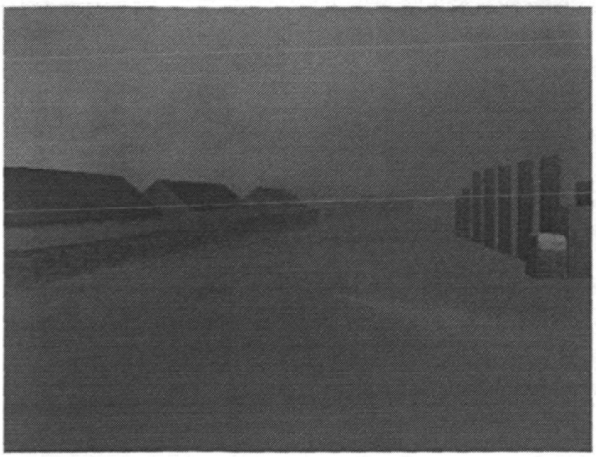


图 5.7 初始时刻效果

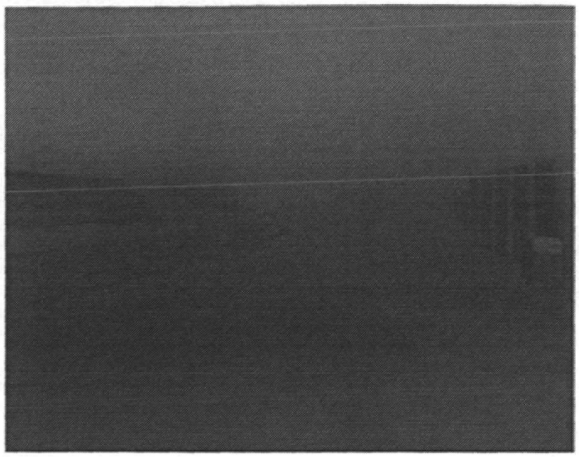


图 5.8 粒子浓度增大效果

5.4 本章小结

本章介绍了开发工具 OGRE，分析了该工具的优良特性，同时介绍了 OGRE 的粒子系统原理，OGRE 通过粒子脚本对粒子的形状、大小、材质、透明度等属性进行定义，用粒子影响器为系统中所有粒子施加外力，如重力、风力等，同时提供粒子的颜色衰减的控制，同时，OGRE 是开源的，允许开发人员进行二次开发，修改代码来完善应用开发。在三维虚拟仿真中，在实现沙尘暴真实感的同时，又获得了一定的帧速率的提高。

结　论

游戏引擎相关技术日趋成熟，并且针对游戏中不同问题不同功能的技术越来越多的应用到现代科学的众多领域。在满足自身应用需求的同时由于其他领域的需求又获得了进一步的发展。在自然景物不规则模糊物体仿真方面，应用游戏引擎中粒子特效技术能够更加真实反映不规则景物运动的动态性、随机性等特点，粒子系统方法被认为是最成功的模拟自然景物的方法之一。

论文所做工作包括以下几个方面：

1．从游戏引擎技术入手，分析研究游戏引擎相关的各种技术。这些技术对游戏整个场景画面能够及时展现至关重要。

2．深入研究不规则物体的模拟方法。重点研究粒子系统的模拟方法，掌握粒子系统理论的建模过程。

3．对沙尘暴现象进行抽象分析，提取出用于建立沙尘粒子系统的各类信息。同时研究分析已经实现的沙尘暴方法。基于粒子系统理论，分析沙尘粒子的属性包括外观属性还有运动等属性，在这些理论基础上建立沙尘粒子系统模型。深入分析沙尘粒子的运动，提出沙尘暴现象主要受风力的影响，根据牛顿第二定律，分析沙尘粒子受力情况，简化沙尘粒子运动模型。并且利用 LOD 技术对沙尘粒子空间分布进行设计。运用 OGRE 粒子系统和 OGRE 提供的对场景的有效管理等技术实现沙尘暴最后的效果展现，获得了较好的真实感以及稳定快速的帧速率。

论文进一步工作：对沙尘暴模拟进行进一步的深入研究，论文的沙尘暴运动模型比较简单，只考虑到风力的影响，而实际上沙尘运动是一个很复杂的现象，在运动过程中受到多种作用力影响，运动模型有待于进一步完善改进。运用其他各种方法对展现效果的渲染速度等进一步进行优化，进而模拟更真实流畅的效果。同时可加入声音，以更加真实的模拟真实环境。

参考文献

[1] 王丹. 基于游戏引擎 3D+GAMESTUDIO 的虚拟现实校园系统设计与实现. 西南交通大学硕士学位论文. 2010:3-5

[2] Christer Ericson. Real-Time Collision Detection. 2. Morgan Kaufmann, 2005:1-2

[3] 王祎. 虚拟现实中碰撞检测关键技术研究. 吉林大学博士学位论文. 2009:2-9

[4] 邹益胜, 丁国富, 许明恒, 何邕. 实时碰撞检测算法综述. 计算机应用研究. 2008, 25(1):9-11

[5] Weinhaus, Frederick M. Texture mapping 3D Models of real-world scenes. ACM Computing Surveys, 1997, 29(4):325-365

[6] J. M. Dischler, D. Ghazanfasrpour*. A Survey of 3D texturing. Computer& Graphics . 2001, 25:135-151

[7] Dominitz, A., Tannenbaum, A. Texture Mapping via Optimal Mass Transport. IEEE Transaction on Visualization and Computer Graphics, 2010, 16(3): 419-433

[8] 浙江大学精品课程系列. 计算机游戏程序设计. http://jpkc.zju.edu.cn/k/686/wljx/Netcourse/H05.html. 2009

[9] 李钊, 王天宝, 李建军, 石虎林. 大规模场景可见性判断算法研究. 无线电工程. 2008, 38(3): 46-48

[10] 刘寅. 基于 LOD 的三维游戏渲染引擎场景绘制技术的研究. 湖南大学硕士学位论文. 2009:3-6

[11] 郭阳明, 翟正军, 陆艳红. 虚拟场景生成中的 LOD 技术综述. 计算机仿真. 2005, 22(12):180-183

[12] Pedro V. Sander, Diego Nehab, Eden Chlamtac, Hugues Hoppe. Efficient Traversal of Mesh Edges using Adjacency Primitives. ACM Transactions On Graphics, 2008, 27(5):144-152

[13] 张文辉. 公告牌技术在室外场景模拟中的应用. 桂林电子科技大学学报. 2008, 28(2): 104-107

[14] Décoret, Xavier, Durand, Frédo, Sillion, FrançoisX, Dorsey, Julie. Billboard Clouds for extreme model simplification. ACM Transaction on Graphics, 2003, 22(3): 689-696

[15] 浙江大学精品课程系列计算机游戏程序设计. http://jpkc.zju.edu.cn/k/686/wljx/netcourse6_2.html. 2009

[16] Laycock, R. G, Ryder, G. D. G, Day, A. M. Automatic generation, texturing and Population of a reflective real-time urban environment. Computer and Graphics, 2007, 31(4):625-635

[17] 陆和军. 基于粒子系统的不规则景物模拟及其可视化. 安徽大学硕士学位论文. 2010:1-4

[18] 赵鸿飞. 基于骨架截集的交互式三维分形树造型技术. 兰州理工大学硕士学位论文. 2010:7-8

[19] 张树兵, 王建中. 基于 L 系统的植物建模方法改进.中国图象图形学报. 2001, 7(5) : 457-459

[20] 王春华, 韩栋. 基于树木综合特征的 L 系统建模. 计算机应用与软件. 2009, 26(8): 159-161

[21] Stam Jos, Fiume E. Depicting fire and other gaseous phenomena using diffusion processes. Proceedings of ACM SIGGRAPH Conference on Computer Graphics, 1995:129-135

[22] Stam J. Stable Fluids. Proceedings of the ACM SIGGRAPH Conference on Computer Graphics, 1999:121-128

[23] 周丽琨. 虚拟现实系统中不规则形体的几何表现. 武汉理工大学博士学位文. 2003 : 17-19

[24] 于乃功, 阮晓钢. 基于细胞自动机的生长系统仿真模型研究. 计算机仿真. 2005, 22(2):228-231

[25] Arno Schodl, Richard Szeliski, David H.Salesin etc. Video Texture. New Orleans: SIGGRAH. 2000:89-95

[26] 林夕伟, 于金辉.基于粒子和纹理绘制的火焰合成. 计算机应用. 2004, 24(4):77-79

[27] 王继州, 顾耀林. 火焰的快速模拟. 计算机辅助设计与图形学学报. 2007, 19(1):102-107

[28] 张芹, 谢隽毅, 吴慧中, 张正军. 火焰、烟、云等不规则物体的建模方法研究综述. 中国图象图形学报. 2000, 5(3):186-190

[29] David Tonnesen Foam, David Tonnesen. Particle Systems for Artistic Expression. Proc. Of Subtle Technologies Conference . 2001:1-4

[30] 贾丽. 基于粒子系统的自然现象仿真. 武汉理工大学硕士学位论文. 2008:6-7

[31] 白玉. 基于粒子系统的煤炭火灾模拟系统. 吉林大学硕士学位论文. 2010:23-25

[32] 陈蕾. 粒子系统理论及其在飞行模拟器实时视景仿真中的应用研究. 吉林大学博士学位论文. 2004:37-39

[33] 百度百科. 沙尘暴: http://baike.baidu.com/view/2097.htm . 2010

[34] 中国沙尘暴网. 什么是沙尘暴天气. http://www.duststorm.com.cn/Article/ShowArticle.asp? ArticleID=26531. 2009

[35] 广东环境保护. 沙尘暴发生的动力学原理. http://www.gdepb.gov.cn/ztzl/scb/t20051118_34785.html. 2005

[36] Liu Shiquang, Wang Zhangye, Gong Zheng, Huang Lie, Peng Qunsheng. Physically based animation of sandstorm. Proceedings of SPIE-The International Society for Optical Engineering, 2007, 6787:259-269

[37] Wang Zhangye, Gong Zheng, Peng Qunsheng. Simulation of atmospheric binary mixtures based on two-fluid model. Graphical Models, 2008, 70(6) :117-124

[38] 姜雄. 关于建立沙尘暴沙粒扩散的数学行及分析. 环境研究与检测. 2004, 17(3): 32-33

[39] 姜雄. 关于沙尘暴沙粒在高空运动的数学模型及分析. 辽宁科技学院学报. 2009, 11(3):41-42

[40] 张芹. 不规则模糊物体建模理论及其在虚拟战场中的应用研究. 南京理工大学博士学位论文. 2001:37-39

攻读硕士学位期间发表的论文和取得的科研成果

致　谢

在论文完成之际，要感谢论文完成中给予过我帮助的哈尔滨工程大学的老师和同学们以及我盼望我成才的父母。

首先要感谢的是导师高伟副教授以及张文燚教授。感谢高老师对于我的指导，老师为人和善，对待事情态度认真。老师用他严谨的治学态度让我认识到不论干什么事，都要做到态度认真，实事求是，严格要求自己。而在张老师所接的项目下对自己进行锻炼，让我真正的能够将书本上的知识转化为实际应用中的理论基础。在项目期间，我也掌握了新的学习方法，学会了如何更快速有效的掌握陌生事物的方法，使我的思维和眼界得到更好的拓宽，提升自己的个人学习做事能力。

更要感谢俞经善老师，俞老师在学习上给了我很大的帮助，老师为人勤奋、平易近人，耐心宽厚，用他独具一格的人格魅力影响着我。在此，向尊敬的俞经善教授表示我最衷心的感谢和崇高的敬意。

同时也要感谢杨静老师、印桂生老师以及开题组和检查组的各位老师，他们在论文开题时及检查过程中给予了我很大的帮助，及时给予我关于论文的指导并指出存在的问题之处，避免我偏离研究方向。

感谢实验室的王宇华老师、张海涛老师、于金峰老师，他们会及时的解决我们在日常生活上遇到的各种问题。

感谢实验室的各位同学，每当论文遇到瓶颈或者心情低落时，通过和大家的谈话交流来来及时解决问题，同时放松自己的心情，保持良好的状态。感谢我可爱的父母，他们给了我很大的支持，在我二十多年的成长生涯里他们给我极大的关爱和呵护，让我能够面对生活中所有遇到的各种困难。同时，也感谢研究生学习生活中教授我们知识的各位老师。

最后，感谢评审的各位专家的辛勤评阅和提出的良好意见。