

智能 CAD 与数字娱乐新进展

孟祥旭 庄越挺 主编

山东大学出版社

序 言

近年来,计算机和网络技术的迅猛发展推动了智能 CAD、计算机动画、数字娱乐及数字艺术的深入研究和广泛应用。中外的计算机科学家和数字艺术家以空前的热情进行着更高起点的计算机艺术创造,软件、数字娱乐等产业界人士也高度关注着计算机艺术创造成果。

为了进一步促进我国在智能 CAD、计算机动画、数字娱乐及数字艺术等领域的研究与应用,为相关领域的专家、学者提供一个交流最新研究成果、进行广泛学术讨论的平台,由中国图象图形学学会计算机动画与数字娱乐专业委员会和中国人工智能学会智能 CAD 与数字艺术专业委员会联合主办,山东大学承办的第三届智能 CAD 与数字娱乐学术会议(CIDE2006)于 2006 年 10 月在美丽的泉城济南举行。

此次会议共收到投稿文章共 170 余篇,内容主要涉及计算机动画、游戏、智能 CAD、计算机图形学、人机交互、虚拟现实、多媒体技术、科学计算可视化、计算机辅助几何设计、娱乐中的人工智能等方面。其中既有理论研究,也有应用技术研究。每篇文章组织两位专家进行评审,经程序委员会推荐,最终有 22 篇被学术期刊录用,其中《计算机学报》5 篇、《计算机辅助设计与图形学学报》7 篇、《系统仿真学报》10 篇。在本论文集中收录了被学术期刊录用文章的摘要;另外,有 47 篇论文全文收到本论文集。

会议还特邀了国内外著名专家、学者就智能 CAD 与数字娱乐的最新动态和热点问题做特邀报告,他们是瑞士 EPFL—Vrlab 实验室的 Danniell Thalmann 教授、清华大学的教授、中科院自动化研究所的研究员等国内外知名专家、学者。

感谢向本次会议投稿的每位作者,没有他们的大力支持,本论文集的出版是不可能的;感谢大会程序委员会委员和参与审稿的各位专家的辛勤劳动;感谢会议组织者为本次会议的成功召开而做的长时间的准备和筹划。

本次会议承办单位山东大学对会议作了精心地组织和周密的安排,他们的大力支持保证了会议的顺利召开,在此表示衷心的感谢!

第三届智能 CAD 与数字娱乐学术会议(CIDE2006)

会议时间:2006 年 10 月 20 日至 10 月 21 日

地 点:山东 济南

主办单位:中国图象图形学学会计算机动画与数字娱乐专业委员会

中国人工智能学会智能 CAD 与数字艺术专业委员会

承办单位:山东大学

会议主席:潘云鹤(浙江大学) 展 涛(山东大学)

程序委员会主席:庄越挺(浙江大学) 孟祥旭(山东大学)

程序委员会委员:

鲍虎军(浙江大学)

董 军(华东师范大学)

费广正(中国传媒大学)

耿卫东(浙江大学)

黄铁军(中科计算所)

李翠华(厦门大学)

廖祥忠(中国传媒大学)

马利庄(上海交通大学)

潘志庚(浙江大学)

沈一帆(复旦大学)

孙正兴(南京大学)

童若锋(浙江大学)

王阳生(中科自动化所)

王兆其(中科计算所)

吴 飞(浙江大学)

吴中海(北京大学)

严启仁(中南卡通科技有限公司)

俞建新(南京大学)

于俊清(华中科技大学)

曾立人(浙江传媒学院)

查 立(上海花样年华数字媒体技术有限公司)

张 方(南望信息产业集团有限公司)

赵杰煜(宁波大学)

赵 阳(中国美术学院)

朱友芹(大众报社集团)

陈雷霆(电子科技大学)

董 璞(北大方正)

高志强(东南大学)

胡事民(清华大学)

黄心渊(北京林业大学)

李 青(香港城市大学)

刘 弘(山东师范大学)

潘春洪(中科自动化所)

沈华清(浙江大学)

史元春(清华大学)

谈 正(西安交通大学)

王文成(中科软件所)

王涌天(北京理工大学)

魏迎梅(长沙国防科技大学)

吴玲达(国防科技大学)

向 辉(山东大学)

杨 青(中科自动化所)

于金辉(浙江大学)

曾海标(中山大学)

曾 亮(国防科技大学)

詹永照(江苏大学)

张 健(南望信息产业集团有限公司)

赵乃良(杭州电子科技大学)

周激流(四川大学)

邹北骥(中南大学)

目 录

计算机动画与游戏

游戏角色的行为建模方法及其应用	曾 亮 达 来 刘衡竹(1)
基于 J2ME 的手机 RPG 游戏引擎架构	税琳琳 石民勇(5)
基于物理的卡通烟雾快速模拟	湛永松 费广正 石民勇(9)
蒙皮骨骼动画角色模型接口的设计与实现	王德才 杨关胜 刘晓明 吴明飞(14)
三维故事板在影视数字特技创作中的应用	蔡 明 周登富 刘 进 宫 林(20)
狄公断案的卡通短片剧情和角色的艺术设计	刘 箴(26)
中国古代神话故事的卡通短片设计	刘 箴(31)
计算机图形动画设计与制作——机器人大战	王福镇 赖炯权 吴恩华(36)

人机交互与虚拟现实

火灾情况下公共娱乐场所人群紧急撤离仿真原型系统	王慧鹏 纪庆革(41)
虚拟战场仿真中基于 A* 算法的路径规划	任立群 高志强 杜 君 徐 惠 郭沛仪(48)
基于 OpenGL ES 的移动 3D 图形引擎的研究与设计	刘 晶 董兴辉 张利柱(53)
罗经稳定过程仿真建模及三维交互技术应用	康 强 刘 鑫 尹兴波(57)
一个基于手写笔输入的书法学习系统	温丽敏 张俊松 于金辉(62)
基于视频中人脸姿态的人机交互系统	谈 正 全晓敏 邬正义(67)
谈交互式玩具的设计	王震亚 刘和山(71)

图像处理与多媒体技术

基于 DT 网络的指纹匹配算法及其改进	贾同辉 李 宁 尹义龙(75)
图像边缘检测技术的研究及对人脸图像处理的实验分析	席小丁 周一民(81)
基于领域匹配的图像间颜色传递	陈 浩(88)
基于 HSI 颜色空间的二维直方图聚类彩色图像分割方法	余 存 赵合计 王志红 陈良庚(93)
基于 B 样条的一个图像序列颜色传递算法	赵 欣 陈昭炯(98)
Shot Boundary Detection and Key-frames Extraction Based on MPEG-7 Visual Descriptors	Kai Sun Junqing Yu Yunfeng He Ning Wang(103)
基于知识库的 Web 图像对象自动识别研究	刘旭花 孟祥增 吴鹏飞 刘俊晓(109)
微生物试剂颜色自动识别的研究	宋怀波 路长厚 蒋德才(113)
基于颜色特征图像检索中的边缘拟合算法	陈载清 白凤翔 张丽芬 陈建兵(117)
基于 Kernel 的马氏距离判别法	刘仰光 何钦铭(122)
应用遗传算法的视频纹理合成算法	孟 宇 李文辉 张 岩 王 岩(126)
Facial Expression Recognition using embedded HMM	Qianqian CHEN Jieyu ZHAO(133)
基于轮廓时空特征约束与信任度扩散的视频前景提取	董兆华 吴 飞 庄越挺(140)

图形与可视化

城市环境空气污染物预报数据的可视化	李观松 蒋志方 王锐 冯志全(148)
-------------------------	---------------------

- 科学计算在线可视化系统的设计与实现..... 贾 佳 周海芳 杨学军(153)
- 在 GIS 中应用 VR 与可视化技术的分析探讨 王永超 胡月明(159)
- 双层纹理拼合算法的研究..... 徐卫兵 翟玉庆 高志强 薛扣平(164)
- 基于地形概要金字塔的高分辨率图像纹理映射研究与实现..... 游鹏飞 皮学贤 李思昆(169)
- Research on Flood Analysis and 3D Visualization of Three Gorges Terrain
..... Guoxin Tan Wenli Xu Wei Liu(175)
- A New Method for Drawing Line Pattern
..... Qi Meng Xia Cunjing ZhengXin Yang ChengLei(182)
- General Framework for Non-photorealistic Line Drawing of 3D Models
..... Yuezhu Huang Chenglei Yang Qian Zhang(189)
- 一种基于八叉树的改进的三维体数据压缩算法..... 宋卫卫 欧宗瑛(197)
- 基于人体颜色包围体的服装布料碰撞检测技术..... 杨 艳 石民勇 费广正(202)

CAD 与 CAGD

THE INTELLIGENT 3D PLASTIC INJECTION MOLD CAD SYSTEM

- Kailing LI Jing GENG Guiheng BAI(206)
- 一个分布式基坑支护工程协同演化系统的设计与实现 * 李陶深 陈秋莲(214)
- 服装 CAD 中三维人台的建模与实现 陈 翔 张 欣 刘 正(219)
- 加权有理三次插值曲线的保凸控制问题..... 邓四清 方 遼(223)
- 带多形状参数的三次多项式样条曲线..... 谢进 胡艳玲(227)

游戏中的人工智能

- 用属性导向归纳法发掘 job-shop 调度中的排序规则 许化强 邱洪泽(231)
- Robocup 中角色决策的有限状态自动机模型 王志红 赵合计 余 存(238)
- 利用分层位图索引进行子集查询..... 孙 凯 郑永清(242)
- SARI 算法安全性分析 王效杰 龙和平(246)

推荐期刊文章摘要

计算机学报

- 骨架形状控制的伸缩内在量方法..... 任绍忠 刘利刚 王国瑾(249)
- 风格化人体运动数据的自动生成与编辑..... 肖俊 庄越挺 陈成 吴飞(249)
- 面向数字媒体服务的数字博物馆网格..... 王锐 郭海 杨承磊 刘士军(250)
- 基于中介面加快光线跟踪计算..... 黄沛杰 王文成 杨刚 吴恩华(250)
- 基于拓展 LIMD 算法的智能动态几何软件设计 林强 任磊 范科峰 陈颖 戴国忠(251)

计算机辅助设计与图形学学报

- 3x+1 推广函数构造广义 M 集及其艺术分形图像 李向华 王钰旋(252)
- 基于方向经验模型分解的纹理合成..... 张岩 李文辉 孙正兴(252)
- 全规整重网格化三维模型的压缩..... 赵向军 路梅 叶澄清(253)
- 卡通风格的人脸生成算法..... 阎芳 费广正 柳婷婷 马文慧 石民勇(253)
- 基于非均匀纹理层的短毛实时绘制..... 杨刚 孙汉秋 王文成 吴恩华(254)
- 参数曲线的最优参数化..... 郭风华 张彩明(254)
- 一种基于镜面对称原理的相机自校准方法? 冯志全 孟祥旭 刘伟伟 高剑(255)

系统仿真学报

- 基于三角带模板的车辙实时可视化方法..... 蔡兴泉 李凤霞 李立杰 战守义(256)

具有灵活目标的鱼群的行为规划体系结构.....	赵光俊 张文俊 张 目(256)
圆环射线分解投影轮廓的三维模型检索方法.....	贾 骥 章 征 卢 江 史哲文(257)
数字娱乐领域中的虚拟人情绪表现模型研究.....	刘 麓(257)
利用改进的 GLA 算法快速检测球场区域	王志芳 于俊清 何云峰(258)
基于 BP 神经网络的颜色模糊量化方法	韩晓微 晏 磊 原忠虎 范立南(258)
自适应跟踪算法在增强现实中的应用.....	吴 越 马利庄 朱 江(259)
基于影像美学原理的虚拟相机自动拍摄生成方法.....	姚 琤 肖 俊 庄越挺(259)
实时流体模拟中边界检测方法的改进.....	刘晓平 陈 皓(259)
基于 GPU 的地形遮挡剔除算法	达 来 曾 亮 李思昆(260)

计算机动画与游戏

游戏角色的行为建模方法及其应用

曾亮 达来 刘衡竹

(国防科学技术大学计算机学院,长沙)

摘要:以游戏角色的行为建模方法为研究内容,重点介绍了基于控制论的行为建模方法,并将其与游戏设计的应用实例结合,系统阐述了游戏设计中的行为建模过程和方法。

关键词:角色;行为建模;控制论

1 引言

PC 游戏的发展经历了从单机游戏到多机网络游戏,从文字游戏、二维游戏到三维游戏的发展过程,如今正如火如荼的兴旺发展。作为游戏的主要组成——游戏角色的设计好坏,是游戏设计成败的关键。游戏角色的设计包括很多方面,如角色的属性设计、角色的成长过程、角色的行为建模,等等。

游戏角色的行为是指游戏角色在特定的游戏环境中可以完成的任何活动及游戏角色对游戏环境所作的响应。游戏角色的行为建模可以采用基于有限状态机(FSM)的行为建模方法,基于 Agent 和多 Agent 的行为建模方法,以及基于控制论的行为建模方法等。各种行为建模方法各有利弊,此处不作详细论述。

本文重点介绍课题组多年研究提出的基于控制论的行为建模方法,并将其与游戏应用实例结合,系统阐述游戏角色的行为建模方法及其应用。

2 基于控制论的角色行为建模方法

控制论的行为观^[1]认为:行为是实体对象(这里是角色的泛指)随时间推移而产生的自身

状态的变迁,它是实体对象内在规律和能力在外部干扰下的表现。当实体对象受到外部环境的作用时,它的行为既可能仍按照某种自主行为状态模式连续地运行,也可能进入激活行为状态模式,根据自身的状态、特性和外部干扰的属性特征,做出适应的反应。外部干扰不是对象行为的直接驱动力,而只是导致对象行为状态发生改变的一种因素,对象行为状态的改变完全取决于对象本身。实体对象对外部刺激的反应不是行为的全部,而只是行为的一部分。

郑援博士^[2,3]给出了基于控制论的虚拟实体行为模型 SCP 模型基本原理如图 1 所示。

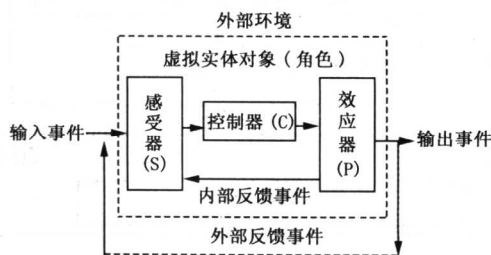


图1 基于控制论的虚拟实体对象行为模型 SCP

SCP(Sensor, Controller, Performer)是由三个相对独立而又密切相关的行为构件(感受器、控制器和效应器)组成的控制系统。其中,感受器

(Sensor)主动从虚拟实体对象内部和外部收集涉及该实体对象的所有事件,并将相关事件过滤出来,传递给控制器(Controller)。控制器的主要功能是对感受器传来的事件进行处理并测试某些相应条件,在此基础上,实行对虚拟实体对象行为的控制,为虚拟实体对象指定新的行为,它是虚拟实体对象行为的控制中心,使虚拟实体对象表现出行为的智能性、自主性和对外部环境的适应性。控制器指定行为后,由效应器(Performer)负责按新的行为执行,使虚拟实体对象表现出对虚拟环境产生效应的特定行为。效应器能够发送事件给其他对象,实现该虚拟实体对象的影响。控制器没有指定新的行为之前,效应器按原行为执行。

内部反馈事件由效应器直接发送到感受器,利用这种反馈,控制器能够对效应器的执行过程进行监督和调整,实现对虚拟实体对象行为的正确控制。外部反馈事件以外部事件的形式反馈

给感受器,它是虚拟实体对象对其他对象施加影响(输出事件)后,其他对象作出反应时发出的事件。利用外部反馈事件,虚拟实体对象能够对自己行为的作用有所认识,从而相应地对行为做出调整。

SCP模型从控制系统的角度抽象出实体的行为机构,使虚拟实体对象具备很强的自主性和独立性,还使SCP模型具备很强的实现连续和更复杂行为的能力,适用领域较广泛。模型表达能力强,易于重用已有的行为建模成果。

为了更好地适应应用系统对行为建模的需要,特别是复杂环境下智能行为仿真的需要,郑义博士^[4,5]对SCP模型进行了细化和扩展,建立了适用于计算机生成角色实体行为建模的ESCP结构模型。

ESCP结构模型如图2所示。

下面我们结合典型的实例,阐述游戏开发中基于控制论的行为建模方法具体应用。

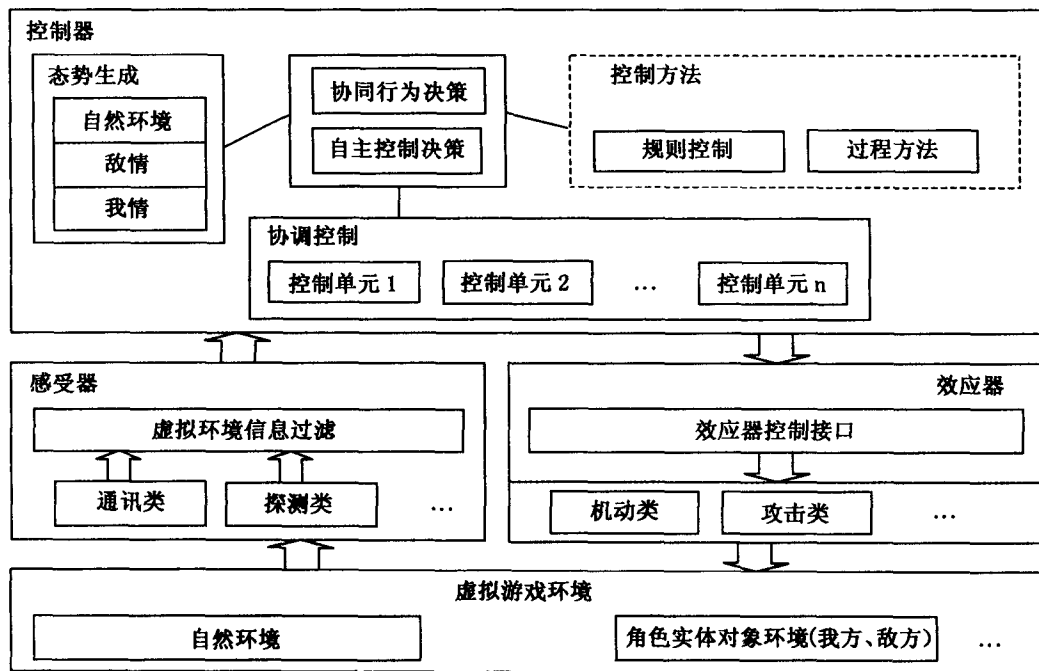


图2 基于控制论的实体对象行为模型 ESCP 结构模型图

3 应用实例

目前流行的游戏很多,无论单机版的游戏,

如《金庸群侠传》、《仙剑奇侠传》等,还是网络版的游戏,如《三角洲部队》、《传奇》、《奇迹》等,游戏角色的行为模型都可以用上节论述的基于控制论

的行为建模方法解释。因为个人情结,这里我们选择第二代网络游戏的代表《MUD》作为典型应用,具体解释行为建模方法的具体实现。

1978年在英国的埃塞克斯大学,罗伊·特鲁布肖用 DEC-10 编写了世界上第一款 MUD 游戏——“MUD1”,这是一个纯文字的多人世界,拥有 20 个相互连接的房间和 10 条指令,用户登录后可以通过数据库进行人机交互,或通过聊天系统与其他玩家交流。MUD1 是第一款真正意义上的实时多人交互网络游戏,它可以保证整个虚拟世界的持续发展。1992 年随着中国 Internet 网络的介入和发展,《东方故事》及其变体如《侠客行》、《西游记》、《风云》等 MUD 游戏在中国游戏者中相当流行,本人当然也是玩家之一,这个游戏目前仍然有人在玩。MUD 游戏虽然只是文字游戏,但它实际上已经为日后图形化、三维化的 MUD 游戏奠定了基石。《传奇》、《奇迹》我想应该就是变体的三维图形化的 MUD 游戏。

3.1 虚拟游戏环境构造

虚拟游戏环境构造包括自然环境对象和角色实体对象。自然环境包括地理环境、气象环境、水文环境等,角色实体对象包括我方和敌方对象。虚拟游戏环境可以通过角色感知器感知,也可以通过角色效应器改变游戏环境的对象状态。

下面是从某个 mud 网站摘抄的一节对游戏环境的描述:

风云南一

虽说比不上城里的热闹,比起乡野小镇来这里还是繁华不少,风云城周围星罗棋布有许多集市,城里的大富人家经常派人到这里采购新鲜的蔬菜水果,远道而来的客商也总在这里歇一歇脚。

春天,微风。

正午到了,忙碌的世界似乎安静了一些。

这里明显的出口是 west, east, north 和 south。

小贩(Xiaofan)

风兵(Fengyun garrison)

在上面的虚拟游戏场景描述中很清楚地看到虚拟游戏环境的构造方法:自然环境描述和角色对象描述是游戏不可缺少的两个部分。实际上虚拟游戏环境是通过无数个游戏场景构成,而每个场景都可以采用文字的或者图像的方法描述,像《大富翁》、《三角洲部队》就分别采用二维图像和三维立体场景方法进行描述。

3.2 感受器构造

感受器获取感知范围内的虚拟游戏环境信息。角色实体感知到的虚拟游戏环境信息是由虚拟自然环境事件、虚拟角色实体对象状态构成的集合。

感受器可以分为通讯类感受器、探测类感受器等不同的类型。其中,通讯类感受器完成角色对象之间的通讯;探测类感受器完成对虚拟游戏环境状态的感知。感受器对获取的虚拟游戏环境信息进行初步的处理,将与本角色实体相关的信息过滤出来,并将这些信息提交给控制器。

在 2.1 节描述的场景就是通过角色探测类感知器指令 look 得到的,也可以通过通讯类感受器得到其他的场景信息,如角色之间可以通过通讯类指令 Talk 交流游戏场景。

3.3 控制器构造

控制器将感受器传来的虚拟游戏环境信息进行分类,生成角色实体自身感知到的虚拟游戏态势。态势信息是角色实体对象所感受到的虚拟游戏环境瞬间的一个状态,为了正确地对虚拟游戏环境作出响应,控制器需要保存一段时间内感知到的虚拟游戏态势信息。

控制器根据虚拟战场态势进行决策,确定是否需要多个角色实体进行协同及如何协同,产生角色实体需要执行的行为。决策过程可以采用多种方法(图 2 中以虚线表示),如:基于规则的决策方法,基于过程控制的决策方法等。决策产生角色实体对象需要执行的行为提交给协调控制器,协调控制器完成行为执行所需的各个控制单元之间的协调。

实际上可以根据玩家对游戏角色的控制进行分类,包括人在回路的控制器和自主控制的控制器。完全由玩家控制的游戏角色,实际上决策过程完全取决于游戏者的大脑,而自主控制的游戏角色,也就是说游戏自然生成的一些角色的控制器完全由游戏设计者在开发过程中事先设计好。计算机生成游戏角色的控制器设计的好坏,直接影响到游戏角色的对抗性,从而决定游戏设计的水平。

3.4 效应器构造

效应器作用于虚拟游戏环境,使角色实体对象表现出确定的行为。游戏过程中角色实体的效应器包括机动类效应器、打击类效应器等不同的

类型。其中,机动类效应器表现角色实体对象的机动特性;打击类效应器表现对指定目标角色的攻击效果。对于不同的角色实体可以建立为相应的效应器等效模型。

例如,在 2.1 介绍的场景中,可以采用 fight 或者 kill 指令对小贩(Xiaofan)或其他角色进行攻击,而根据角色之间力量的对比,决定攻击的效果,也就是效应器的效能。

每个游戏角色可以包括很多效应器,这是游戏角色对虚拟游戏环境的直接参与手段,因此而改变游戏环境的状态。

上面介绍了游戏角色行为建模的几个环节,无论是文字角色游戏,还是二维三维图像角色游戏,角色的行为建模都需要从这几个部分进行考虑和细化。

4 结 论

本文结合游戏应用介绍了基于控制论的游戏角色行为建模方法,该建模方法不仅适合游戏

领域,在虚拟战场仿真计算机生成兵力系统的应用也很广泛。单个角色实体的行为建模方法已经基本成熟,但对多个角色实体的行为建模方法还不成熟,是下一步研究的难点和重点。

参考文献

- [1] Nobert Wiener. CYBERNETICS Or Control and Communication in the Animal and the Machine. New York, 1961
- [2] 郑援,李思昆等. 虚拟实体对象的行为建模方法研究. 国防科技大学学报, 1998. 1
- [3] 郑援. 虚拟世界行为建模技术的研究、实现与应用. 国防科技大学计算机学院, 1998
- [4] 郑义. 计算机生成兵力行为建模与实现技术研究. 国防科技大学计算机学院, 2003
- [5] 曾亮,郑义等. 基于控制论的计算机生成兵力实体对象行为模型[J]. 系统仿真学报, 2005. 4, 17. 773~774

基于 J2ME 的手机 RPG 游戏引擎架构

税琳琳 石民勇

(中国传媒大学动画学院, DECA 实验室; 北京)

摘 要:本文对手机游戏开发中的若干重要技术进行了深入探讨,提出了一种基于 J2ME 的 RPG 游戏引擎架构方案,在手机游戏开发过程中能够提高游戏生产效率,提高游戏的产量和质量。

关键词:手机游戏, 框架设计, 场景管理, 角色管理, 键盘缓冲, 图像压缩

1 引言

手机游戏产品的生命期非常短,一款比较好的游戏通常也只能维持在两个星期左右,因此,手机游戏必须要缩短研发周期,快速高效地提交新作品。而游戏引擎相当于游戏的心脏和骨架^[8],它把游戏制作中核心的部分固定下来,使新游戏省去了许多重复代码的编写,从而能够大大提高游戏开发的速度和质量。

本文在第二节里首先总括了游戏的基本结构,随后的几节分别从场景的显示和绘制、角色管理、玩家交互以及图片处理四个模块讲述了游戏开发中的关键技术,随后给出了引擎的具体实现,最后提出总结和展望。

2 游戏的基本结构

在游戏中,至少需要三个功能:显示画面,接受玩家输入和对输入产生反馈,通常称为渲染,输入输出和逻辑三个模块^[1,3]。

将这三个模块组合在一起有很多方法,比如事件驱动:让画面保持不变直到检测到输入,触发逻辑运算并根据结果对画面重新渲染。但是游戏往往并不是这样,游戏是时间驱动的。无论是否有事件,游戏都在不停地循环,检查是否有输入,运行逻辑,渲染画面。从一定程度上讲,游戏很像电影,随着时间流逝画面在改变。

既然是时间驱动,游戏中就会有帧的概念。帧是某个时刻显示在屏幕上的画面。从整体上看,游戏就是一系列的帧不断播放着,像动画片

一样,区别在于动画片每帧的内容是动画师预先设定好的,而游戏中玩家可以通过交互改变播放的内容。开发游戏的主要任务就是安排每一帧具体播放的内容。在每一次游戏循环中,搜集玩家的输入、运行逻辑更新数据、然后根据更新后的数据安排下一帧显示的内容。所以一个最简单的游戏结构如图 1 所示。

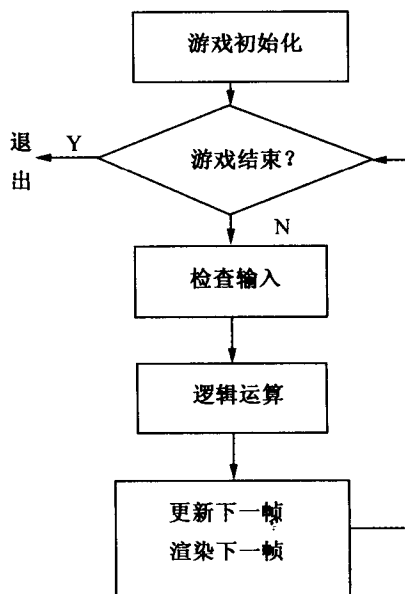


图 1 游戏的基本结构

这是一个最基本的结构,特别对于比较简单的 J2ME 游戏来说,这个结构更加有代表性。

3 场景渲染模块

场景是游戏角色所存在的世界。它从视觉上看是游戏的地图。不过从程序的角度看,场景是一种数据结构,不但包含了地图显示的图形信息也包含了角色在场景中活动所需要的物理信息和事件信息。比如地图上有些地方是不可以通过的,有些地方主角走过去会触发一个事件,等等,这些信息往往包含在场景中。

3.1 地图绘制

常见的方法是拼 Tile,俗称贴瓷砖。因为二维数组很好的对应了二维坐标。二维数组的每一项对应地图上某格使用那个瓷砖,整个二维数组就能表示整张地图。但是由于一维数组开销更少,所以在手机设备上使用一维数组更为合理。

3.2 地图卷轴

在卷轴式地图中,屏幕可以看成是一个摄像机,每次只显示地图的一部分。随着主角的移动,屏幕所显示的部分跟着变化,形成卷轴。如图 2 所示。

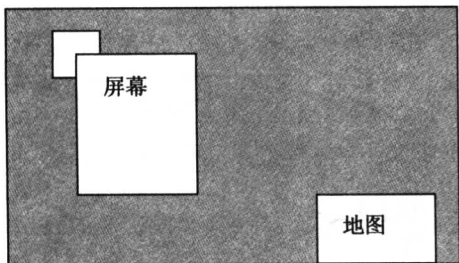


图 2 地图卷轴原理

卷轴式地图显示的时候,从屏幕(摄像机)所覆盖的第一个 Tile 开始绘制。需要注意的是屏幕所能覆盖的 Tile 包括部分覆盖的 Tile。图 2 中第一个 Tile 就是一个部分覆盖的 Tile,尽管只是部分覆盖,它也是需要绘制的第一个 Tile,否则屏幕上就会出现没有图像的部分。造成部分覆盖的原因是屏幕的卷动速度并不是 Tile 宽度的整数倍。实际上,卷轴速度往往可能是变化的,这样的卷轴显得比较自然。

按照一定的顺序将屏幕所能覆盖到的所有 Tile 都绘制一遍,场景的绘制工作就完成了。这些 Tile 绘制的时候要采用相对于屏幕的坐标,而不是相对于地图的坐标。游戏中经常用到屏幕坐标和地图坐标的转换。转换的方法很简单。设

Tile 在地图上的坐标为 (x, y) , 屏幕相对于地图的坐标为 (sx, sy) , 那么 Tile 在屏幕上的坐标就是 $(x - sx, y - sy)$ 。注意单位要统一, 往往 Tile 的单位是格, 所以要转换成像素单位, 即格子坐标乘上 Tile 宽度或高度。到这里其实地图的主要功能都已经实现了, 再加上管理 Tile 图片的部分, 一个简单的 Tile 引擎就成形了。

3.3 碰撞和事件检测

场景中有些地方是无法通过的, 有些地方经过时会触发某个机关, 这些碰撞和事件信息也存储在场景信息中, 用另一个一维数组表示, 可以称之为物理层。

碰撞检测在游戏开发中是一个永恒的话题, 由于使用了物理层, 我们只要计算出下一帧角色所要到达的位置是哪一格, 如果这一格的物理层信息是不可通过, 则阻止角色前进。这种碰撞检测不同于物体间的碰撞检测, 不必和所有的物体进行遍历判断。缺点是不够精确。减小物理层格子的大小可以提高精确度, 但会使数据变多。所以物理层的碰撞检测只用在地图上, 如果有比较特殊的物体, 就让它作为一个角色存在, 利用角色间的碰撞检测。

4 角色管理模块

角色可以是存在于场景中的一切东西, 它可以是活动的如 NPC, 也可以是静止的, 如一个箱子。角色的数据结构视作用而不同, 但基本的应该有坐标, 速度, 以及使用到的图片等数据。

4.1 玩家角色

玩家角色是玩家在游戏世界中的化身, 代替玩家在游戏世界中游历。这里涉及三个比较主要的概念: 任务、经验和物品交易。任务和物品交易信息保存在 NPC 的属性当中, 由具体的 NPC 负责处理; 经验从战斗中获得, 保存在怪物的属性当中, 由怪物负责处理。玩家角色要负责的只是接受键盘输入的控制, 进行相应的描绘, 以及判断下一步是否可以跨越, 更新 RMS 数据等。

把这些功能从玩家角色中划分出去的原因是游戏中只有一个玩家, 玩家角色对其他角色的碰撞和事件处理是很容易判断的, 但是游戏中怪物和 NPC 却不只一个, 要玩家角色需要把所有的怪物和 NPC 都遍历一次, 手机中会难以承受这样大的开销。

4.2 NPC 和怪物人工智能

游戏中 NPC 和怪物可以是站立不动的,但多数时候他们会在一定范围内自由活动,并根据寻路算法有自己的行动路线。

常用的寻路算法有两种:状态空间搜索和启发式搜索。

状态空间是指从出发点到目标点之间由于求解条件的不确定性和不完备性,而出现的多条求解路径组成的图,有深度优先和广度优先两种搜索方法。但是状态空间搜索有一个明显的缺点:它需要在一个给定的空间中实施穷举^[4]。

启发式搜索是在宽度优先搜索的同时,先将下一层的所有节点通过一个估价函数进行过滤,然后再决定搜索方向^[3]。该方法在一定程度上缩小了搜索范围,提高了搜索速度,从而比状态空间搜索更适合在手机上应用。

游戏中广泛使用的 A* 算法就是一种典型的启发式搜索算法,但考虑到手机游戏中内存和容量的限制,需要将经典 A* 算法的估价函数和数据结构作一定调整。

在节点 n , 出发点到目标点之间的移动耗费可以用估价函数 $f(n)$ 来表示:

$$f(n) = g(n) + h(n)$$

其中, $g(n)$ 是状态空间中从出发点到 n 节点的实际代价, $h(n)$ 是从 n 节点到目标点最佳路径的估计代价。

很明显, $g(n)$ 的值是固定的, $h(n)$ 的选取是整个估价函数的关键。在手机中, $h(n)$ 可采用曼哈顿方法计算,即考虑从当前节点到目的节点之间水平和垂直的距离的总和,忽略对角线方向。

此外, A* 算法还保存着一张 Open 表和一张 Close 表,分别用来存放未访问和已访问的节点。在 PC 平台上这样的数据结构可能直接用链表就实现了,插入和删除也非常容易,但是 JAVA 是没有指针的,需要改用动态数组,存储的移动路径也不能用栈,而要改用 JAVA 提供的 Stack 类来处理。

5 玩家交互模块

游戏同其他艺术形式的最大差别在于同玩家的交互性,也就是接受键盘输入,进行逻辑运算并重新渲染。

手机上捕获键盘输入有专用的接口: keyPressed, 但这样势必把按键处理和其他逻辑运算

分离在 keyPressed 和整个游戏的逻辑循环中,但是如果所有逻辑都放在 keyPressed 显然更不合适,因为除了按键事件以外,游戏还有很多其他的逻辑要处理。所以我们采用键盘缓冲将按键信息存起来,然后在整个游戏的逻辑处理时就可以判断这一帧按键的状态。

利用按键缓冲,除了可以判断一个键是否按下、松开,还可以判断一个键是否一直被按住了,甚至可以判断组合键。

为了节约内存,可以用一个整数来表示所有的按键状态,整数有 32 位,每位代表一个按键状态,足够表示大多数手机的按键了。可以把目前的按键状态记为 Key。

同时由于不同型号的手机设备差异性比较大,读入同一个输入事件,可能会返回不同的数据格式,因此在处理按键事件的代码中需要对不同的手机作一个封装处理,把读回来的数据统一为用整数表示的形式,记为 keyCode。

那么,在处理按键按下的代码中将 keyCode 与 Key 进行或运算,在处理按键松开的代码中进行异或运算,即可得知接受玩家输入以后最新的按键状态 KeyNew, 这个 KeyNew 就是从 keyPressed 中获取了输入信息以后,保存出来,用在整个游戏逻辑处理中的键盘状态。

6 图片处理模块

手机中因为计算能力有限,所以许多经典的图形学算法都不能直接使用。相对用得较多的还是对图片资源的处理。也就是在保证游戏效果的前提下尽量节省手机的空间和内存。

PNG 格式是手机游戏中唯一支持的图片格式,每张 PNG 图片由图片内容信息和一定的格式信息组成。这些格式信息在每张图片中都要占用一定空间,如果在预处理时能够把 n 张 PNG 图集成为一张大图,使用时再裁切成单张图片使用,就可以节省 $n-1$ 份格式信息所占的空间,在手机中这通常是一个不小的数字。

不过作为游戏的图形引擎模块,还要考虑到一些更专业的技术,如角色换装。这项技术用一份图片数据与一份调色板片段结合,能够生成新的图片,如图 3 所示。

新图片与原图片相比,角色外型并不发生变化,变化的是服装和道具的颜色。这里,新角色的产生只多存了一份调色板数据,而很多的文件格式

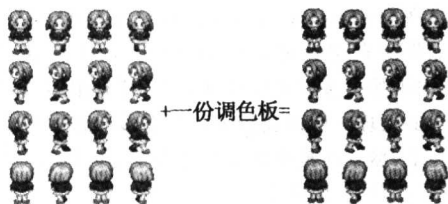


图3 角色换装技术

式信息,角色姿态信息等都复用的是原来图片的数据,这样就有效节省了手机内存。

7 引擎实现

把上述引擎架构方案具体应用到游戏开发中,可以得到一个包括地图编辑器和脚本编辑器在内的手机游戏开发工具,见图4。

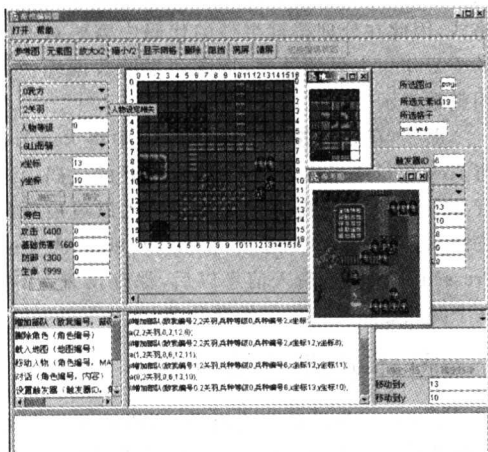


图4 游戏编辑器

该引擎可以广泛用于手机 RPG 游戏的开发。在引擎帮助下,美工和策划人员在编辑地图和游

戏事件时变得非常直观、方便,同时程序员也能迅速把握艺术家的创作意图。由于很多重复性的代码都在引擎中被固化下来了,程序员的工作也得到了进一步减轻。

8 结论

本文提出了一种基于 J2ME 的手机 RPG 游戏引擎架构方案,针对游戏开发的特点,分为了场景渲染模块、角色管理模块、玩家交互模块和图片处理模块,分别对其中的重要技术进行阐述,以期能实现代码复用,提高游戏的生产效率。

参考文献

- [1] Jason Lam. J2ME and Gaming. 2003
- [2] Kasper Fauerby, Improved Collision detection and Response, 2003. July
- [3] 詹建飞, J2ME 精解, 电子工业出版社, 2006. 1, 108~116
- [4] 陈和平, 张前哨, A* 算法在游戏地图寻径中的应用与实现, 计算机应用与软件, 2005, 12, 118~120
- [5] 谢志鹏, 蔡灿辉, 游戏地图最短路径搜索设计与实现, 计算机工程与应用, 2005, 24, 115~117
- [6] 李远泰, 冯永晋, 基于 J2ME 技术的手机游戏图形处理, 中国科技信息, 2005, 16
- [7] 郑鹏, 林子禹, 脚本语言及其应用, 武汉大学学报(自然科学版), 1999, 10, 716~718
- [8] Nokia company, Symbian OS: 平台独立引擎的开发, Nokia forum, 2004. June

基于物理的卡通烟雾快速模拟

湛永松 费广正 石民勇

(中国传媒大学动画学院数字技术与艺术研发中心,北京)

摘要:提出一种利用流体物理模型和粒子系统相结合来快速模拟计算机卡通烟雾动画的方法。首先介绍如何根据流体物理模型对卡通烟雾的动态路径结构进行定义,然后基于该模型驱动粒子系统运动。场景中的烟雾浓度取决于粒子的空间分布情况。粒子属性中定义了一个浓度函数和作用半径,卡通烟雾的浓度特征可通过少量粒子表达。在绘制过程中,不同浓度的区域被贴上相应的烟团纹理,从而获得卡通化的烟雾效果。该算法不仅实现生动的卡通烟雾运动效果,而且只需少量粒子便可获得连续的卡通烟雾浓度特征,在很大程度上提高了模拟速度。试验结果表明该算法能简单高效地模拟生动自然并具有卡通效果的烟雾动画。

关键词:卡通烟雾;Navier-Stokes 方程;粒子系统;卡通动画;计算机动画

1 引言

传统的卡通动画以手绘的方式将现实世界艺术化地呈现出来。现实世界中除了包含人类、动物等角色对象,还有各种环境效果。烟雾是常见的一种环境效果,以卡通化方式呈现的烟雾效果能有效提升整个卡通动画的生动性。由于烟雾属于流体物质,具有很大的随机性,难以对其运动轨迹进行定义,传统的卡通动画中的烟雾效果具有较强的机械感。此外,流体物质的模拟通常需要较大运算量,影响了计算机模拟的实时性。

本文提出一种将流体物理模型和粒子系统相结合以生成计算机卡通烟雾动画的方法。粒子系统在流体物理模型的驱动下运动,其分布特性决定了场景中的烟雾浓度分布。针对粒子系统存在计算量大的缺陷,我们使用童若锋^[1]提出的方法对传统的粒子模型进行扩展,在其属性中定义了一个浓度函数和作用半径,从而只需少量粒子便可获得连续的烟雾浓度。为了实现卡通效果,我们在动画师的指导下,预先定义了一系列的卡通化烟团作为纹理,这些烟团能体现出卡通烟雾在不同浓度下的特征。在渲染过程中,系统为场景中的不同浓度区域贴上相应烟团纹理。本文算法简单高效。粒子系统的运动规律为流

体物理模型支配,在一定程度上保证了运动路径的合理性,避免了机械单一的运动形态;卡通烟雾的浓度分布则由粒子的浓度函数和作用半径控制生成,提高了模拟速度;预先定义的卡通化烟团纹理有利于实现风格化的卡通效果。我们的方法既保证了卡通烟雾运动轨迹的生动性,又只需少量粒子便控制了烟雾的浓度特性,在很大程度上提高了模拟实时性。

在下面的章节中,第一节将对相关工作进行介绍。第二节介绍本文方法,其中包括流体物理模型的定义,烟雾浓度场的生成以及卡通烟雾的渲染方法。第三节给出本文的一些试验结果并讨论。最后一节为全文总结。

2 相关工作

于金辉^[2-5]是较早从事卡通流体动画技术研究的学者。他根据艺术家绘制的卡通流体画面提取流体运动骨架,然后在骨架上设计不同的形状并进行绘制。其算法能获得生动的卡通火焰、流水以及烟雾等效果,但它属于一种过程化流体模型,流体运动骨架的设计过程需要较多的人工调整。

为了获得卡通化的渲染效果,Witting^[6]以及Neyret^[7]都采用了平移非真实感纹理的方法,即

纹理随着流体的运动而变形,但该法通常会因为流体的运动以及纹理混合而丢失细节信息。Lamorlette^[8]在2002年提出了一个风格化火焰模型,其算法将火焰的自然扩散过程分为八个阶段,并对每个阶段的火焰进行风格化建模,从而组成一个火焰动画系统。虽然该算法很有效,但它只适用于描述火焰扩散这种特殊情况,不具有通用性,而且工作流程非常复杂。

目前真实感流体的模拟广泛使用基于 Navier-Stokes 方程^[9]的流体物理模型,Fedkiw^[11]的工作代表了当前真实感流体模拟的最高水准。粒子系统^[12]在真实感流体的模拟中也有广泛应用。由于粒子系统的运动规律具有一定的可控性,因此,为了提高动画师在动画制作过程中的主动权,研究者们开始考虑将粒子系统引入风格化流体动画的算法设计中。Selle^[13]提出的卡通烟雾系统即属于这种类型。其算法使用图像缓存中的深度差信息求取卡通烟雾轮廓,此外,通过将粒子进行旋转和拉伸,系统能较生动地模拟卡通烟雾与天花板碰撞的效果。Fiore^[14]提出的卡通烟雾算法为动画师提供了较大的主动权。该算法允许动画师自定义烟雾的轮廓线和加速线,粒子的运动路径为轮廓线和加速线所支配。但以上两种方法的共同缺陷是在模拟过程中使用了大量粒子,从而带来较大的计算量,影响了模拟速度。童若锋^[1]提出了一种基于粒子系统的真实感烟雾快速模拟方法,其算法为了克服传统粒子系统中计算量大的缺陷,为每个粒子定义作用半径以及浓度函数,从而用少量的粒子就可以得到较为连续的速度分布,但其工作主要致力于表现真实感的烟雾效果。

3 本文方法

我们的卡通烟雾模型主要包括两个模块:运动路径定义模块,该模块确定了卡通烟雾的运动轨迹;形状绘制模块,该模块负责生成烟雾浓度场并进行卡通化绘制。下面我们逐一详细介绍这两个模块。

3.1 卡通烟雾运动路径定义

烟雾的运动过程除了受到外力影响,还包括其自身因素,例如平移和扩散。对于烟雾等流体类物体而言,其运动过程包含有很大的随机成分,从而破坏了关键帧之间物体各个部分的一一对应关系,因此关键帧插值法难以适用于流体动

画。如果从传统手工绘制的烟雾动画中提取骨架,则需要设计数学表达式来建立波状骨架。为了避免出现过分机械的感觉,骨架的设计过程需要较多人工调整,比较繁琐,而且这种方式设计的骨架只适用于描述特定情况下的卡通烟雾效果,通用性不强。

从视觉效果来说,虽然卡通烟雾与真实感烟雾的表现形式不一样,但两者并非完全对立。前者呈现给观众的是一种卡通化的效果,即使这种卡通化效果经过了人为的夸张处理,也应该呈现出一种自然流动的感觉。因此,我们考虑将流体物理模型用于描述卡通烟雾的运动骨架。流体物理模型遵循的是 Navier-Stokes 方程^[9]:

$$\frac{\partial u}{\partial t} = -(u \cdot \nabla)u - \frac{1}{\rho} \nabla \rho + \nu \nabla^2 u + f \nabla \cdot u = 0$$

流体物理模型的优点是能生成自然生动的运动轨迹。动画师通过调整该方程中的风力(即 f)等参数就可以随心所欲地改变运动轨迹,从而方便快捷地实现自己的设计意图。目前,研究者们普遍基于该方程来模拟真实感流体动画,并且已经提出了适用于计算机图形学领域的快速稳定的方程算法^[10],此后 Fedkiw^[11]对该法进行了补充,避免了原算法中由于使用半拉格朗日法所导致的细节信息丢失。

本文采用 Fedkiw^[11]的方法对卡通烟雾的运动路径进行定义和求解。当烟源位置和风力方向发生改变时,运动路径亦相应变化。图1所示是本系统所生成的路径图。我们定义的烟源位置为图片正下方,烟源处有周期性向上吹拂的风力,路径通过速度场的流线表示。图1是我们在烟雾运动过程中不同阶段所截取的任意两幅运动路径图。从图中可以看到,我们只需定义一些高层的参数,例如烟源位置、风力方向等,系统便可以自动获得生动自然的烟雾运动轨迹,避免了人工设计运动路径的繁琐。

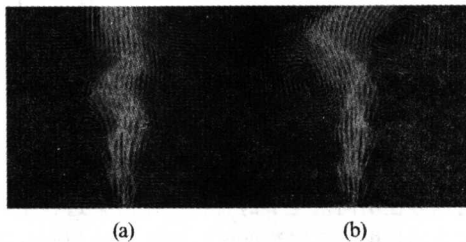


图1 烟雾运动路径图

3.2 形状绘制

卡通烟雾的性质主要体现在其绘制方式。在传统的手工绘制烟雾动画中,动画师通常在烟雾较浓的地方绘制颜色深的大烟团,在烟雾稀薄的地方绘制颜色较浅的小烟团。烟团的具体绘制方式取决于动画师个人的绘制风格,例如有的动画师倾向于使用椭圆型烟团,而有的则喜欢使用星型烟团。动画师的个人风格决定了手绘烟雾所体现的卡通艺术效果。

为了让计算机自动生成具有卡通效果的烟雾动画,我们需要重点考虑两个问题:首先,烟雾浓度在运动路径上是如何分布;其次,如何根据烟雾的浓度分布来获得卡通风格效果。

由于烟雾并不具有固定的形状,其浓度分布实时变化。为了描述卡通烟雾在运动路径上的浓度分布情况,粒子系统是一个好的解决方法。粒子系统具有良好的控制性,非常适合描述烟雾等流体状物质,但缺陷是通常需要大量粒子,影响模拟速度。

针对粒子系统存在计算量大的缺陷,我们使用童若锋^[1]提出的方法对传统的粒子模型进行扩展,在其属性中定义了一个浓度函数和作用半径,从而只需少量粒子便可获得连续的烟雾浓度。我们的场景被划分为一个网格区域,烟雾的浓度定义于每个网格点的中心处。当计算某一网格点的浓度时,如果粒子的作用半径 r 大于它与网格点之间的距离 d ,即该网格点位于粒子的作用域内,则粒子对该点的浓度值有贡献,其大小可由粒子在该时刻的浓度函数求取。粒子在各个年龄段的作用半径以及该作用半径内各采样距离处的浓度值为预先求出,其每一年龄对应一条作用半径,各作用半径组成一个作用半径表,每一作用半径对应于一个浓度函数表,浓度函数表中存放作用半径内各采样距离处的浓度值。在绘制过程中通过查表即可获得浓度值,减少了实时运算量,提高了模拟速度。

当获得连续的烟雾浓度分布以后,本算法就模拟动画师的行为,在烟雾浓度大的地方绘制深色的大烟团,否则绘制浅色小烟团。为了获得卡通化的效果,我们在动画师的建议下,预先定义了一些能反映烟雾浓度特性的图元作为烟团纹理。图2所示为本系统所用到的部分图元。

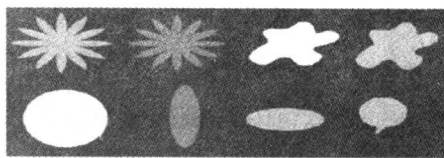


图2 烟团图元

在绘制过程中,我们根据场景内每个网格中心的烟雾浓度将其与某图元相关联,图元以纹理的方式贴到该网格区域内。对于烟雾浓度大的区域,我们为其贴上颜色更深的图元,反之,则贴上颜色较浅的图元。此外,我们还可以定义一个浓度阈值,当某区域的烟雾浓度小于该阈值时,则表明此处烟雾非常稀薄,因此不贴任何图元。由于图元是在动画师的建议下制成纹理,因此所生成的烟雾在物理模型的运动规律支配下,呈现出自然流动的卡通化效果。

我们的方法与Selle^[13]和Fiore^[14]的方法所采用的贴图方法不同。在他们的方法中,图元直接贴在每一个粒子之上。在粒子密集的地方,这种贴图方法会带来很大的冗余和计算开销。我们则是根据粒子的分布情况来获得场景中的浓度分布,只有浓度值大于一定阈值的,网格点上才会贴上相应图元。场景中浓度值大于阈值的网格点数目远远少于粒子数,因此我们的方法计算量小,能获得更快的渲染速度。

3.3 卡通烟雾的动画生成

前面我们介绍了本算法中卡通烟雾的运动路径定义以及形状绘制方法。归纳起来,本文的卡通烟雾动画可由如下步骤生成:

- (1) 指定烟源位置以及风力的大小和方向;
- (2) 在动画师指导下制定一系列具有卡通特性的烟团图元作为纹理;
- (3) 在烟源处周期性地产生一定量粒子,然后粒子受到Navier-Stokes方程的支配而运动;
- (4) 通过粒子的分布情况计算场景中每个网格点处的烟雾浓度;
- (5) 根据烟雾浓度为每个网格点贴上相应烟团纹理。

4 结果分析

我们在Dell Precision380工作站(奔腾3.2G CPU, Nvidia Quadro Fx540显卡, 1G内存)上实现了本文算法,开发平台为开放图形库OpenGL