



[12] 发明专利申请公开说明书

[21] 申请号 02130736.9

[43] 公开日 2003 年 4 月 30 日

[11] 公开号 CN 1414496A

[22] 申请日 2002.11.13 [21] 申请号 02130736.9

[71] 申请人 北京航空航天大学

地址 100083 北京市海淀区学院路 37 号北京
航空航天大学 6863 信箱

[72] 发明人 郝爱民 沈旭昆 梁晓辉

[74] 专利代理机构 北京北新智诚专利代理有限公
司

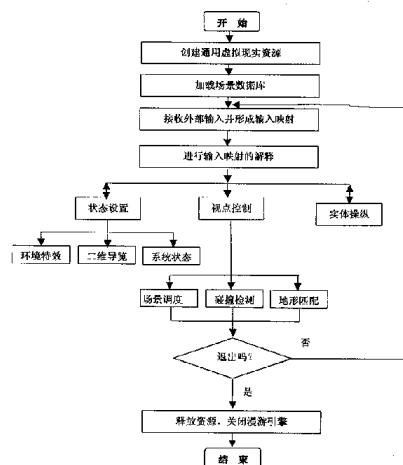
代理人 张卫华

权利要求书 2 页 说明书 12 页 附图 4 页

[54] 发明名称 一种通用虚拟环境漫游引擎计算机
系统

[57] 摘要

本发明属于计算机虚拟现实技术领域。其实现方法包括创建通用虚拟现实应用资源，加载符合场景描述规范的场景数据库文件，定制漫游状态机制，设置输入映射与解释机制并在仿真循环的每一帧都接受一次外部输入，支持多种输入设备接入漫游系统，根据外部输入命令进行视点控制、实体操纵及状态设置，在视点控制过程中实施多种场景复杂度消减策略、进行碰撞检测和地形匹配，支持使用标准二维输入设备选择并操纵三维场景中的物体，设置多种漫游控制状态和环境特效的步骤。用双相机模型进行视景观察、碰撞检测与漫游控制，多种场景复杂度消减策略支持，给出输入映射机制，功能完整，接口清晰，可方便扩充输入设备等。适于一用户开发虚拟现实环境系统。



1、一种通用虚拟环境漫游引擎计算机系统，由个人计算机、图形加速卡、步行器、立体视觉显示及跟踪设备、场景数据库以及漫游引擎核心组件构成，该漫游引擎计算机系统实现方法包括以下步骤：

- (a) 创建通用的虚拟现实应用资源；
- (b) 加载符合场景描述规范的场景数据库文件；
- (c) 定制漫游状态机制；

其特征在于还包括：

(d) 设置输入映射与解释机制并在仿真循环的每一帧都接受一次外部输入，支持多种输入设备接入漫游系统；

(e) 根据外部输入命令进行视点控制、实体操纵及状态设置，在视点控制过程中实施多种场景复杂度消减策略、进行碰撞检测和地形匹配，支持使用标准二维输入设备选择并操纵三维场景中的物体，设置多种漫游控制状态和环境特效，满足不同用户的需求。

2、根据权利要求1所述的通用虚拟环境漫游引擎计算机系统，其特征在于：所述的视点控制是使用双相机模型进行漫游的视点控制，创建观察视点相机和行走视点相机，观察视点相机受限于行走视点相机，行走相机用以实现漫游过程中的多点碰撞和保持行进方向，观察相机有三个转动自由度。

3、根据权利要求1所述的通用虚拟环境漫游引擎计算机系统，其特征在于所述的视点控制进一步包括：

- (a) 在视点相机及行走相机的运动中如检测到碰撞，则判断碰撞的性质；
- (b) 如果是场景控制面，则启动场景调度策略，进行室内外场景的调度管理；
- (c) 如果是可操纵的运动实体，则按模型中定义的运动坐标系及运动参数进行场景中实体的运动控制；
- (d) 如果确认发生了碰撞，则给出碰撞响应。

4、根据权利要求1所述的通用虚拟环境漫游引擎计算机系统，其特征在于步骤(d)进一步包括：

- (a) 输入映射与解释机制用以将漫游引擎和输入设备隔离，以最大限度的减少输入设备对漫游系统内核的影响；
- (b) 从视点控制及设备功能中抽象出一个中间层；
- (c) 将输入设备的控制映射到该中间层并使其成为视点驱动的控制源。

5、根据权利要求1所述的通用虚拟环境漫游引擎计算机系统，其特征在于：在整个漫游过程中，同时支持多种场景高度及复杂度消减策略，包括以下步骤：

- (a) 按观察视点相机与各场景模型的距离调度多层次细节模型的精度替换；

(b)用场景数据库中的三维空间平面和行为参数确定场景控制面，进行室内外场景及复杂室内场景的调度划分，在漫游过程中，基于视线的碰撞检测算法检测到模型控制面后，将建模时定义的控制行为标识解析出来，映射到场景控制对象集中，根据控制对象集中的对象列表，装载可见对象，卸载不可见对象，实现场景调度管理；

(c)模型加载时自动消减实体模型中的冗余多边形；

(d)生成实体表面的纹理。

6、根据权利要求1所述的通用虚拟环境漫游引擎计算机系统，其特征在于：所述的碰撞检测与响应进一步包括：

(a)分析相机运动的原因，建立步行相机和观察相机向前的碰撞检测线段；

(b) 观察相机碰撞检测；

(c) 步行相机碰撞检测，如果发生碰撞，则进行虚拟环境场景调度处理；

(d) 若未发生碰撞，则相机前进，并进行地形匹配。

7、根据权利要求1的通用虚拟环境漫游引擎计算机系统，其特征在于：所述的地形匹配技术进一步包括以下步骤：

(a)行走相机前进；

(b)如果视点位置发生变化，调用碰撞检测模块；

(c)碰撞检测结果为发生碰撞，则忽略视点位置改变，行走相机和观察相机升高一定高度，同时按地形表面方向设置观察相机的视向；

(d)若结果为未发生碰撞，则视点到达一个新位置。

8、根据权利要求1所述的通用虚拟环境漫游引擎计算机系统，其特征在于：状态设置功能包括开/关雾化效果、开/关二维地图向导、开/关碰撞检测、选择透明处理方式、设置天气状况等的一种或几种。

9、根据权利要求1所述的通用虚拟环境漫游引擎计算机系统，其特征在于：所属的环境特效包括以雾化为技术支撑的恒定帧频方法的步骤。

10、根据权利要求9所述的通用虚拟环境漫游引擎计算机系统，其特征在于：所述的恒定帧频方法进一步包括：

(a)取出前2秒内每帧平均耗用时间；

(b)计算当前帧频；

(c)计算当前帧频与目标帧频之差；

(d)如果当前帧频大于目标帧频则增加雾化可见距离，否则减小雾化可见距离；

(e)设置新的雾化参数并结束。

一种通用虚拟环境漫游引擎计算机系统

技术领域

本发明属于计算机虚拟现实技术领域，具体地说是一种虚拟现实环境的通用漫游驱动计算机系统。

背景技术

由于基于虚拟现实技术的城市规划、虚拟建造、场景漫游的重要特点和将会产生的巨大社会、经济效益以及对相关产业带来的根本性变革，一些西方发达国家从 80 年代中期开始投入大量资金及人员支持虚拟现实方面的研究。其中美国高级技术中心、亚特兰设计公司、北卡罗来纳大学、派拉迪姆公司等机构在理论和实践上都取得了许多成果，并开始推出初步实用、价格较为昂贵的专用商业软件工具包。

目前，在三维场景漫游的前沿形成了两个不同的研究方向，即利用可量测尺寸数据的基于几何的场景绘制和基于源图象序列的虚拟场景表示与绘制。基于几何的虚拟现实发展较早，随着计算机技术的高速发展，无论是硬件方面的图形加速卡，还是软件方面的三维建模工具、三维图形环境，都有众多的公司、科研机构推出了多种解决方案，如美国视算科技公司的图形加速卡性能价格比不断地大幅度提升；马尔特锦公司的建模软件 MultiGen II Pro 及其数据库格式 OpenFlight 几乎已成为仿真界的工业标准；VEGA(维格)、Performer(表演者)等图形系统已发展为功能强大、易于使用的商业软件。但是虚拟现实技术毕竟发展时间不长，基于几何的虚拟现实技术仍然存在许多有待解决的问题。相对而言，基于图象的虚拟现实技术起步较晚，其主要研究方向是降低基于几何虚拟现实技术的场景建模的复杂度及人工强度，降低虚拟现实技术系统对硬件性能的苛刻要求。但基于图象的虚拟现实技术在虚拟实体操纵、具有力反馈的人机交互、具有物理特性的碰撞检测及响应等方面还有很多理论要做进一步的研究。

卡罗来那大学伯克利分校场景漫游工作室是世界上从事场景漫游研究较早并取得突出成果的科研机构之一，他们 1990 年开始进行复杂模型的实时漫游策略研究。1996 年，在 SGI(美国图形工作站生厂商)的 Power series 320 工作站上实现了计算机系新系楼 Soda Hall(索塔礼堂)的实时漫游。Soda Hall 模型由 1418807 个多边形构成，占据 21.5 兆的硬盘空间，模型使用了 406 种材质及 58 种不同纹理。由于研究小组采用了高效的数据存储结构、多级层次细节技术、场景调度算法、实时可视区域判定算法及预算计算处理等多种技术，使得 Soda Hall 的实时仿真率即每秒钟刷新频率恒定在每秒 20 帧左右。在多年的研究工作中，北卡大学场景漫游工作室提出并不断完善了 UNIGRAFIX 视景数据库，开发了一些软件工具包，如 AutoCAD DXF 数据格式转为 UNIGRAFIX 格式的数据转换器、对象多级层次细节技术自动生成器和简易场景自动生成工具等。但是对国内而言，想要在代码级重用一个完整的漫游引擎实际上不可能的。

国内也有一些研究机构从事场景漫游技术的研究，其中以杭州大学工业心理学研究室实现的故宫漫游为代表。故宫漫游采用脚踏车作为交互工具，让漫游者实际上原地不动地在虚拟故宫里骑行。故宫漫游主要展现的是建筑物的外观即室外场景，其室内部分做了大量简化，采用了以纹理映射技术为主的建模方法，碰撞检测及碰撞响应等功能相对较弱，该系统不提供虚拟实体的操纵功能。

发明内容

为克服上述缺点，本发明的目的在于提供一种通用虚拟环境漫游引擎计算机系统，可以实现虚拟环境漫游中的基本但重要的共性功能，而且较好地封装了一批关键技术，降低了一般用户开发虚拟现实应用系统的难度，在代码级实现了系统功能的重用，既有利于规范开发，又可以减少开发周期和费用。

为达到上述目的，本发明的通用虚拟环境漫游引擎计算机系统，由个人计算机、图形加速卡、步行器、立体视觉显示及跟踪设备、场景数据库以及漫游引擎核心组件构成，其实现方法包括：创建通用的虚拟现实应用资源，加载符合场景描述规范的场景数据库文件，定制漫游状态机制，设置输入映射与解释机制并在仿真循环的每一帧都接受一次外部输入，支持多种输入设备接入漫游系统，根据外部输入命令进行视点控制、实体操纵及状态设置，在视点控制过程中实施多种场景复杂度消减策略、进行碰撞检测和地形匹配，支持使用标准二维输入设备选择并操纵三维场景中的物体，设置多种漫游控制状态和环境特效，满足不同用户的需求的步骤。

视点控制是使用双相机模型进行漫游的视点控制，创建观察视点相机和行走视点相机，观察视点相机受限于行走视点相机，行走相机用以实现漫游过程中的多点碰撞和保持行进方向，观察相机有三个转动自由度；视点控制进一步包括在视点相机及行走相机的运动中如果检测到碰撞，则判断碰撞的性质，如果是场景控制面，则启动场景调度策略，进行室内外场景的调度管理，如果是可操纵的运动实体，则按模型中定义的运动坐标系及运动参数进行场景中实体的运动控制，如果确认发生了碰撞，则给出碰撞响应的步骤；输入映射与解释机制用以将漫游引擎和输入设备隔离，以最大限度地减少输入设备对漫游系统内核的影响，从视点控制及设备功能中抽象出一个中间层，将输入设备的控制映射到该中间层并使其成为视点驱动的控制源；在整个漫游过程中，同时支持多种场景高度及复杂度消减策略，包括按观察视点相机与各场景模型的距离调度多层次细节模型的精度替换，用场景数据库中的三维空间平面和行为参数确定场景控制面，进行室内外场景及复杂室内场景的调度划分，在漫游过程中，基于视线的碰撞检测算法检测到模型控制面后，将建模时定义的控制行为标识解析出来，映射到场景控制对象集中，根据控制对象集中的对象列表，装载可见对象，卸载不可见对象，实现场景调度管理，模型加载时自动消减实体模型中的冗余多边形，生成实体表面的纹理；碰撞检测与响应进一步包括分析相机运动的原因，建立步行相机和观察相机向前的碰撞检测线段，观察相机碰撞检测，步行相机碰撞检测，如果发生碰撞，则进行虚拟环境场景调度处理，若未发生碰撞，则相机前进，并进行地形匹配的步骤；地形匹配技术进一步包括行走相机前进，如果视点位置发生变化，调用碰撞检测模块，碰撞检测结果为发生碰撞，则忽略视点位置改变，行走相机和观察相机升高

一定高度，同时按地形表面方向设置观察相机的视向，若结果为未发生碰撞，则视点到达一个新位置的步骤；状态设置功能包括开/关雾化效果、开/关二维地图向导、开/关碰撞检测、选择透明处理方式、设置天气状况等的一种或几种；环境特效包括以雾化为技术支撑的恒定帧频方法的步骤；恒定帧频方法进一步包括取出前2秒内每帧平均耗用时间，计算当前帧频，计算当前帧频与目标帧频之差，如果当前帧频大于目标帧频则增加雾化可见距离，否则减小雾化可见距离，设置新的雾化参数并结束。

本发明的特点是：1、功能完整，接口清晰。使用本发明开发场景漫游系统，其工作量等同于创建场景数据库，漫游驱动部分几乎不必再重写代码。2、采用双相机模型进行视景观察、碰撞检测与漫游控制。双相机模型不仅可以有效地解决地面低矮障碍物的碰撞检测问题，同时也可以较好的解决在坡度路面上地形匹配的视向问题。3、多种场景复杂度消减策略的支持，特别是支持独创的基于场景控制面的调度算法。该算法保证了室内外结合场景的连续漫游问题，可以有效地提高漫游的实时性，同时，算法简单，便于掌握使用。4、遵循虚拟现实业界数据标准，在漫游引擎内封装了对虚拟场景的多种描述的解释，如门、窗的运动定义及高于地面物体对漫游行进的行为特征，如果是台阶可以进行地形匹配，使视点自动升高，如果是绿地围栏则不能通过，需要绕行等等。5、给出了输入映射的机制，同时支持鼠标、键盘、游戏杆、步行器、头盔跟踪器等多种输入设备，并可以方便的扩充输入设备接入。6、实现了基于雾化的恒定帧频技术，通过调解雾的浓度来控制场景中可见几何面片的数量，进而调整漫游帧频。

本发明的虚拟环境漫游引擎计算机系统与现有技术相比，其有益效果是：它是一个相对完整的漫游引擎，实现了输入映射与视点控制、虚拟场景调度管理、多层次细节模型切换、纹理、光照、地形匹配、碰撞检测与响应、实体操纵、二维地图向导、恒定帧频方法等引擎功能，与国内的两个典型漫游引擎相比，具有明显的优势，整体功能与美国伯克利大学的漫游系统相当，虽在大场景处理方面功能不如其强大，但本发明中的基于场景控制面的调度算法实现简单，功能可以满足中型室内外场景的要求，与伯克利大学的漫游系统相比，无需场景的预处理过程，同时，可以在基本配置的个人计算机上完成场景的实时漫游。在应用系统开发方面，不必再做具体的程序编制工作，只需要关心场景的建立过程，它是一个通用性强、功能完整、技术先进的虚拟环境漫游引擎。

附图说明

- 图1示出本发明的通用虚拟环境漫游计算机系统主程序流程图；
- 图2示出本发明输入设备映射与视点控制图；
- 图3示出本发明漫游引擎中的视点控制模型图；
- 图4示出本发明鼠标输入时屏幕区域划分示意图；
- 图5示出本发明模型控制面场景调度控制工作流程图；
- 图6示出本发明实施例进出某建筑物的模型控制面定义图；
- 图7示出本发明视点控制过程中的地形匹配技术示意图；
- 图8示出本发明二维地图多通道模型图；

图 9 示出本发明基于雾化的恒定帧频算法图。

表 1 为通用漫游引擎框架中定义的主要控制状态表；

表 2 为用键盘进行视点控制时采用的设备映射表；

表 3 用鼠标进行视点控制时采用的设备映射表；

表 4 为通用漫游引擎系统键盘功能表。

具体实施方式

下面结合附图和具体实施方式对本发明作详细说明。

本发明的通用虚拟环境漫游引擎是一个可以独立运行的漫游引擎，采用的软件平台为 Virtual Studio C++6.0 及 OpenGVS4.3 软件，操作系统为 Windwos2000。

参阅图 1，本发明首先要创建通用虚拟现实环境资源，这些资源是场景、实体、帧缓存等，然后加载符合场景描述规范的场景数据库文件，接收外部输入并形成输入映射，之后进行输入映射的解释，输入映射解释又分为状态设置、视点控制和实体操纵，状态设置包括设置系统状态、二维导览和环境特效，视点控制视点控制包括对场景调度、碰撞检测、和地形匹配的控制，由视点控制判断是否退出和是否释放资源。定制漫游状态机制时，设置输入映射与解释机制并在仿真循环的每一帧都接受一次外部输入，支持多种输入设备接入漫游系统，根据外部输入命令进行视点控制、实体操纵及状态设置，在视点控制过程中实施多种场景复杂度消减策略、进行碰撞检测和地形匹配，支持使用标准二维输入设备选择并操纵三维场景中的物体，设置多种漫游控制状态和环境特效，满足不同用户的需求。

参阅图 2，为了适应不同漫游应用的各种需求，本发明的漫游引擎中设置了定制漫游的状态机制。大多数虚拟现实应用都是 OPENGL 兼容的应用，而 OPENGL 的内部实现本身就是一种状态机制。所以，在本发明中设置状态机制即保持了与图形系统的一致性，又增加了漫游系统的灵活性。

通用漫游框架中定义的绝大多数功能都被定义为可选择项，也就是说，漫游者可以根据自己的需要打开或关闭某些功能，如开/关雾化效果、开/关二维地图、决定是否进行碰撞检测、选择透明处理方式等。另一方面，漫游者还可以对漫游系统的初始状态进行设置，比如观察相机的初始位置、相机行进的速度步长、转角的步长、系统模拟的气候条件(晴、多云、阴)、时段(早晨、中午、傍晚)等。在实现漫游系统视点控制时，多采用直接读取设备信息并进行处理的方式。随着大量 I/O 设备和新的交互控制技术的发展，将会有越来越多的交互设备得到使用。由于所配备设备的多样性和可扩展性，无疑会给软件的设计和维护带来复杂性。表 1 定义了本发明的漫游引擎框架中的主要控制状态。

基于上述原因，本发明定义了一个基本的设备映射，即从视点控制及设备功能中抽象出一个中间层，将输入设备的控制映射到这一中间层并使其成为视点驱动的控制源。

可以看出，输入设备的控制信号映射为控制指令，使得视点控制与输入设备的直接控制信号分离开来，成为独立的功能模块。使用这种“隔离”技术，既方便了漫游系统对输入设备的扩展，又保持了漫游引擎相对的独立性。

参考图 3，在漫游系统中，视点即为人眼的“化身”，其功能与现实世界中的摄像机或照相机相同。视点控制通常是指观察相机的运动控制。视点控制模采用与众不同的两组相机分别模拟人眼及脚的运动。模拟人眼的相机称为观察相机，模拟人脚的相机称为行走相机。观察相机的运动受限于行走相机。一般的漫游系统只设置观察相机，本发明设置行走相机的目的有两个：一是在漫游过程中实现多点碰撞检测，二是用行走相机保持行进的方向，而允许观察相机有更多的转动自由度，观察相机除了沿 Y 轴左右转动外，还可以沿 X 轴上下转动，模拟人抬头、低头的动作。行走相机的基本参数包括：距离地面的高度(walker_height)，前进或后退的步长(step)，沿 Y 轴的转动步长(θ)。walker_height 的设置可以提高漫游系统整体性能，因为行走相机将参与视点控制过程的碰撞检测，当 walker_height 设置为非 0 值如 0.10 时，虚拟环境中低于 10cm 的实体与视点都不会发生碰撞。对应于现实世界的行为是，所有低于 10cm 的障碍如某建筑物中的门槛是人可以跨越的。这样，漫游系统将减少大量无效计算，从而提升系统的整体性能。观察相机的基本参数与行走相机基本相同：距离地面的高度(eye_height)，前进或后退的步长 step, 沿 Y 轴的转动步长 θ 。除此之外，观察相机还设置了另一个转动参数，即沿 X 轴转动的角度步长 α ，它允许漫游者抬头、低头去观察三维场景。

视点控制模型中还为观察相机和行走相机各定义了三个用于碰撞检测的参数，v_p0、v_p1、v_p2 以及 w_p0、w_p1、w_p2，分别表示相机当前的位置、前进方向上一定距离处的一点、后退方向上相同距离处的一点。视点控制模型定义了相机运动：观察相机运动集合：{FORWARD, BACKWARD, TURN_LEFT, TURN_RIGHT, LOOK_UP, LOOK_DOWN}；行走相机运动集合：{ FORWARD, BACKWARD, TURN_LEFT, TURN_RIGHT }

由此可知，视点控制即为步行相机与观察相机的运动控制。而相机运动的方式无外乎两种：平移及旋转。本发明用视线方向上的位移步长 step 定义相机位置的变化，用相机沿坐标轴的旋转角度步长 θ 及 α 定义视线方向的变化。因此，输入设备的映射问题转化为输入设备控制量到相机运动类型及 step、 θ 、 α 的数学变换的问题。

键盘输入的映射最为简单，采用直接映射的方法，以观察相机为例，如表 2 所示。

参阅图 4 鼠标输入时屏幕区域划分示意图，鼠标输入的视点控制映射则不同，本发明采用了屏幕区域化分的方法来识别一次鼠标输入的控制功能。记当前鼠标位置为(xm,ym),由设备坐标变换的有关原理可知， $xm \in [-1, 1]$, $ym \in [-1, 1]$,且二维显示屏幕的左下角坐标为(-1, -1), 右上角为(1, 1)。于是，按图 5 所示划分屏幕为十个区域，分别标记为区域 1 到区域 10，并设：用鼠标设备控制视点时，相机最大平移步长为 MAXSTEP,最大转动角度步长为 MAXROTATE,令 speed = MAXSTEP * fabs(ym), rot = MAXROTATE * fabs(xm), 则鼠标设备的映射如表 3 所示。其他输入设备映射与鼠标设备映射基本相似。

基于几何式虚拟现实技术描述一个中等复杂程度的社区漫游虚拟环境，尤其是建筑物室内环境，一般要花费几万乃至数十万个实体表面三角形。由于图形硬件性能、计算机物理内存、CPU 主频等条件的限制，不采取场景复杂度消减策略的虚拟现实系统将无法保证实时交

互漫游。采用多种场景复杂度消减策略，可以提高漫游系统的整体性能。该策略包括基于碰撞检测的模型控制面场景调度方法、冗余多边形消减、纹理映射、多级层次细节模型等几种方法。

参阅图 5，场景调度管理有两方面含义，一方面，由于计算机物理内存容量的限制，有时无法将一个复杂的场景数据库一次全部调入内存之中。此时，场景调度管理是指分块装载、卸载模型数据。另一方面，由于计算机硬件，尤其是图形加速卡性能的限制，在一帧之中渲染过多的多边形也将超过系统处理能力的极限。减少一帧渲染的多边形数量是场景调度管理的另一层面的含义。

经典的室内场景调度控制如 PVS(潜视集, Potentially Visible Set)算法等大都将漫游过程与场景调度处理分成两个阶段进行。对于某一特定的虚拟环境，上述算法在“离线”状态下将建筑物按围隔房间的墙面、地板、天花板等平行于世界坐标轴的分割面进行空间划分，形成 Cell(空间单元)，然后在分割后的空间单元上标注可视区域如门、窗等，再逐一计算 Cell 到 Cell 以及 Cell 到实体的可见区域，最后将计算结果存储在场景数据库中。这样，在漫游阶段，漫游驱动程序无需再做复杂的可见区域的判断，直接使用预计算的结果进行场景调度控制，达到较好的实时效果。但是，预计算方法也存在一些明显的不足，如空间划分及可见区域预计算依赖于虚拟环境、预计算结果占用较大的场景数据库空间、预计算工作量较大等。尤其对于中空型建筑物而言，预计算可见区域算法的效率相对较低。

本发明的漫游引擎提出了一种基于碰撞检测的模型控制面场景调度控制方法。该方法将场景建模与漫游控制有机结合，用主观评价标准确定可见区域，达到了较好的效果。

可以看出，模型控制面在建模阶段设定并与其他场景数据一同存储在场景数据库中。控制面为一空间平面，由空间方程 $Ax+By+Cz+D=0$ 确定。与一般空间平面不同的是，控制面在建模过程中定义了一些特殊属性，其中最为重要的就是场景调度控制行为。控制行为在建模阶段仅仅定义了一个标识，具体的控制行为由漫游引擎解析和确定。

漫游过程中，基于视线的碰撞检测算法检测到模型控制面后，将建模时定义的控制行为标识解析出来，映射到场景控制对象集中，并根据控制对象集中的对象列表，装载可见对象，卸载不可见对象，实现场景调度管理。场景控制对象集由漫游程序反复实践，最终由主观标准确定。即确定在某一控制面处，哪些对象为可见对象，哪些为不可见对象，记录到漫游驱动的数据文件中。

图 6 是 进出某建筑物的模型控制面定义图，当漫游视点接近某建筑物正门前的模型控制面时，碰撞检测算法给出该处的控制面规定的控制行为，如进入建筑物，经行为解析模块分析后，找出对应的场景调度控制对象集，很自然地，应关闭建筑物外形轮廓模型，调入建筑物室内模型，同时对六个楼层的可见与不可见对象进行相应显示与禁止显示的处理。

模型控制面采用六元组 $(A, B, C, D, arg_1, arg_2)$ 方式描述，其中 A、B、C、D 唯一确定三维空间中的一个平面 $Ax+By+Cz+D=0$ ， arg_1, arg_2 为控制面给定的控制行为参数，由检测算法传递到控制行为解析模块。某建筑物场景模型中设置了多个模型控制面，图 6 给出了进出某建筑

物的场景调度控制示意。六个模型控制面设定了进出某建筑物的场景调度行为。除此之外，还在各楼层切换的楼梯处以及接近外窗的楼层地板附近设置了控制面，如当视点处于建筑物二层时，其它楼层除构成中庭的窗子、门、墙等对象外，大部分对象均不可见，楼层切换控制面可以控制这类场景调度管理，大大减少一帧之中绘制的多边形的数量。

三维场景建模的一个基本原则是使用最少的多边形获取相同的逼真视觉效果。描述实体模型的表面数据经常存在冗余现象，单独建模的实体在进行模型整合时也同样会发生数据冗余。消除这些冗余的表面多边形可以在很大程度上降低整个场景的复杂度。例如现有技术的多级层次细节窗户模型由 156 个三角形构成，窗户最上面的一个窗棱通常的建模技术将其用 12 个三角形(6 个矩形平面)描述，由于两侧顶端的平面与侧面的竖直平面重合，不仅是冗余数据，而且会造成 Z 值争夺，应该预以删除。如此对整个窗户进行一遍删减，同样视觉效果的窗户则可由 104 个三角形构成。再考虑模型整合产生的数据冗余，最上面的棱面将与墙面重合，也可以删减，依此类推，最终的窗户模型只要用 96 个三角形就可以完整地描述其几何拓朴信息。与 156 个三角形表示的模型相比，其视觉效果没有任何损失，却提升了近 40% 的系统性能。

纹理映射技术是计算机图形应用广泛采用的实体表面细节逼真表现技术，也是控制场景复杂度、加速图形绘制速度的有效方法。它将一个矩形数组定义的二维纹理图象映射到三维实体表面，或者通过一个过程修改实体表面的光强分布。

生成纹理的一般方法，是在一平面区域即纹理空间上预先定义纹理图案，然后在物体表面的点与纹理空间的点之间建立映射。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表面上。可以用类似的方法给物体表面产生凹凸不平的外观，或者称凸凹纹理。不过这时纹理值作用在法向量上，而不是颜色亮度上。无论是生成颜色纹理还是凸凹纹理，一般只要求与真实图案大致逼近，不必做精确的模拟，以便在不显著增加计算量的前提下，较大幅度地增加实体表面的真实感。

纹理映射技术可以大幅度降低场景复杂度，但此方法也有一定的局限性。当漫游者接近实体模型时，纹理表现的实体细节就会缺乏真实感。如果将多级层次细节模型与纹理映射技术结合使用，可以在保证实体细节真实感的同时有效地降低场景实时绘制的复杂度。

多级层次细节模型是指对同一个场景或场景中的实体使用具有不同细节的描述方法得到的一组模型，供绘制时选择使用。由于虚拟现实通常用多边形网格来描述场景中的几何实体，因而多级层次细节模型采用不同繁简程度的网格实现对虚拟场景不同精细层次的实时绘制。通常情况下，多级层次细节模型使用较多的多边形建立实体的最细节的描述，而用较少的多边形建立实体的轮廓描述。当漫游视点距离实体较近时选择细致模型，视点距离较远时选用轮廓模型。一般的漫游系统都采用了三级多级层次细节模型技术，以上述描述的窗户为例，高级多级层次细节模型以设计数据为基准，全部采用几何建模的方法给出实体描述，中级多级层次细节模型则将窗户部分以纹理代替，但窗台仍采用几何模型，低级多级层次细节模型将整个窗户作为一个纹理平面。

漫游引擎负责实现基于模型控制面的场景调度控制方法的控制行为解析及场景调度管理。由于漫游系统中存储了控制对象集，其中给出了某一模型控制面的打开即要加载的对象集合及关闭即要卸载的对象集合，实施行为解析及场景调度变得十分的简单。

当视点与模型控制面发生碰撞并返回控制参数后，行为解析只要简单地定位该模型控制面对应的控制对象集。然后，对于某一个模型控制面，按下面的方法将对象集中的对象逐一卸载、加载。

如图 7 所示，漫游系统在视点控制的过程中，必须进行视点与虚拟环境中各种实体间的碰撞检测，进而给出合理的碰撞响应。碰撞检测算法包含于相机运动控制模块之中，一旦发生了视点与虚拟实体间的碰撞，输入设备给出的前进或后退指令将被忽略，即相机不再向前或向后运动，这是本发明中给出的对碰撞的简单响应。地形匹配通常是指虚拟环境中的动态实体如地面车辆运动时姿态随地形高低起伏、左右偏转。在漫游系统中，漫游者控制的是人眼的“化身”，即视点，所以地形匹配即指视点的高度及视向随地形变化而改变。

在碰撞检测算法中，可以看到地形匹配模块被调用的时机。任何使视点位置发生变化的外部输入都将引起碰撞检测模块的调用，一旦碰撞检测的结果显示为已经发生碰撞，碰撞响应就表现为忽略视点的位置改变，视点停在原地不动。如果碰撞检测的结果为没有碰撞发生，视点就按规定的步长及方向到达一个新的位置，但这一视点位置及视向并非是视点对外部输入做出反应的最终位置，最终的位置要在地形匹配之后确定。

相机在视向方向上前进 step 距离之后，其所处的位置记为 WPOS(xw,yw,zw)。经地形匹配后的相机位置必然在视点坐标系的 Yv 轴之上。经过点 WPOS 建立竖直的直线方向，即 Yv 轴的方向，利用视线与空间多边形集求交的方法，求出与 Yv 轴相交的平面及交点坐标。比较交点中 Y 分量不超过某一给定值且最大的一个相交平面确定为地形表面。将行走相机置于该表面的 Y 值高度之上。观察相机的高度值设置为行走相机加上一个固定值 eye_height-walker_height。视向的地形匹配问题相对更为简单。取地形表面多边形的方向，令观察相机的方向与地形表面方向一致，但保持行走相机视向不变。

二维地图是漫游系统中广泛使用的漫游向导工具。与三维场景视图相比，二维地图的优势在于，它可以提供更加广阔的视野空间，便于漫游者从总体上把握当前所处的位置及周边环境状况。

一般情况下，开发二维地图显示模块需要在建模空间中提取二维基本特征，如道路、建筑尺寸数据，使用 OPENGL 图形系统进行线条绘制，再对多边形区域填充相应的颜色，还要控制一个视点指针，与三维场景中的视点保持一致的运动。

如图 8 所示，本发明使用了另外一种更加简捷的二维地图开发方法，利用三维场景正投影生成二维地图。这种方法依照计算机图形学正投影的原理，将三维场景模型“压缩”到一个平面上，然后运用相机资源，实现地图的显示、缩放以及二维与三维视点的同步运动。实现二维地图向导时使用了多通道程序设计技术，即建立了如图 8 所示的通道模型，图中的 mainchannel(左通道)也是主通道，用于显示漫游系统的三维场景，channel2D(右通道)用以显示

二维地图。主通道占据整个屏幕，二维地图通道只占据屏幕的右上角八分之一的区域。在 channel2D 中，不仅设置了相机，还预设了光照和观察体，只是相机的镜头总是正对着表示二维地图的模型平面。当然，任何系统都不可能为了快速、便捷地实现二维地图显示而将表示三维世界的模型全部调入 channel2D 中，那将导致系统负荷的双倍增加，完全得不偿失，而是将三维模型简化到只有地表数据(即将所有高于地面的实体部分全部删除)，经简单的修补后，形成二维地图的三维表示模型。建立了多通道机制以及二维地图表示模型后，二维地图与三维场景的视点指示同步问题转换为 channel2D 中的相机跟踪 mainchannel 中的行走相机问题，在跟踪过程中 channel2D 中的相机的镜头方向保持不变。二维地图的缩放功能通过为相机设置不同的焦距实现。

虚拟现实应用不仅追求高的交互仿真率即帧频，而且希望帧频保持一致、恒定，至少应保持在用户期望的某一区间，尤其对于作战模拟类型的虚拟现实应用，跳跃的帧频将导致错误的模拟结果，并使参与者感到无所适从。帧频是绘制一帧场景所用时间的倒数，因此，要获得恒定帧频，必须控制一帧绘制场景表面多边形的数量。

1996 年，北卡大学的托马斯梵可豪斯和瑟斯泰勒在 Soda Hall 漫游系统中提出了一种有效但较为复杂的恒定帧频的方法。他们为场景数据库中的每一对象设立三元组(O, L, R)，其中，O 表示场景中的一个对象，L 表示该对象的一个 LOD 模型级别，R 表示对该对象施以的绘制

$$\sum_s Cost(O, L, R) \leq \text{Target Frame Time} \text{ 且 } \sum_s Benefit(O, L, R) \text{ 最大}$$

算法。在对象三元组上建立两个评价函数 Cost(O, L, R) 和 Benefit(O, L, R)，Cost(O, L, R) 用于计算对象选用 L 级 LOD 时 R 算法的绘制时间，Benefit(O, L, R) 用于评价 L 级的多层次细节及 R 算法条件下该对象对整个场景的视觉方面的贡献。这样，恒定帧频的问题转化为算法中使用一个线程专门负责上述两个函数的计算，在每一帧绘制之前确定选用的多层次细节级别及绘制算法。

北卡大学的恒定帧频方法是一种高效的基于预取的优化算法，可适用于大多数虚拟现实应用，但该算法中的计算主要依赖主观标准，且整个算法较为复杂。

雾化不仅可以提高场景的真实感，还可以提高虚拟现实系统的整体性能。雾化的实质是将场景原有颜色与雾的颜色按一定的比例因子进行融合，比例因子的取值与雾化模型有关。与 OPENGL 兼容的图形加速引擎大都提供三种雾化模型：

其中 f 为比例因子，density 为雾的密度，z 为观察坐标系下视点到场景面片中心点的距离，

$$\begin{aligned} f &= e^{-(density \cdot z)} && (\text{GL_EXP}) \\ f &= e^{-(density \cdot z)^2} && (\text{GL_EXP2}) \\ f &= \frac{\text{end} - z}{\text{end} - \text{start}} && (\text{GL_LINEAR}) \end{aligned}$$

start 和 end 为雾化作用深度的启始值与终止值。

在 RGBA 颜色模型下，观察平面上每一像素点雾化后的颜色 C 由下式计算：

$$C = fC_i + (1 - f)C_f$$

C_i 为场景面片的 RGBA 颜色值, C_f 为雾的颜色, f 为上面讨论的比例因子。

上述雾化模型给定了雾化的一个重要性质, 即, 视点与实体距离越远, 场景雾化的程度越大, 实体在场景中的可见度越小。样条变换雾化模型对场景的雾化效果较为显著, 当视点与实体的距离超过一个给定值后, 实体雾化后的颜色全部由雾的颜色代替, 不需要再做实时绘制。

综合上面的分析, 本发明设计了通过调整雾化模型参数达到小范围内调整每帧场景显示平均用时, 进而达到交互仿真率恒定目的的方法。

参阅图 9, 本发明采用了一种简单但十分有效的恒定帧频技术, 即基于雾化可见距离调整的恒定帧频方法。这种方法可以认为是一种事后调整方法。所谓事后调整是指根据前面一段时间内的帧频与目标帧频的差来动态设定雾化效果, 使得每帧处理的实体表面多边形数量增加或减少, 从而达到调整帧频升降的目标。具体的算法可用图 9 所示的流程图表示。从中可以看出, 本发明实现雾化效果的算法, 是用当前帧频向目标帧频逐步逼近的方式, 使漫游系统的帧频在目标帧频附近“摆动”, 从而达到恒定帧频于一个固定区间的目的。这个算法适用于区域较大场景的虚拟现实应用。

表 4 给出了本发明实施例的通用漫游引擎系统键盘功能祥表。按照上述步骤就可以开发使用本发明的虚拟现实环境的通用漫游驱动计算机系统。

状态控制	热键定义	缺省状态
雾化效果启用/禁止	CTRL+Z	启用状态
二维地图向导启用/关闭	M	关闭状态
室外场景可见/不可见	P	可见状态
相机向前运动的碰撞检测启用/禁止	CTRL+G	启用碰撞检测功能
相机向后运动的碰撞检测启用/禁止	CTRL+R	启用碰撞检测功能
纹理映射打开/关闭	CTRL+X	打开状态
线框显示打开/关闭	CTRL+P	关闭状态
帧数统计结果显示/不显示	TAB+ '-'	禁止显示状态
调试信息输出允许/禁止	CTRL+T	禁止状态
DOF 实体自动操纵允许/禁止	O	允许状态
透明算法切换选择	F8, F9	AF 状态
二维地图相机变焦选择	F1-F5	最大焦距状态
帧缓冲模式单/双方式切换	CTRL+E	单帧缓存方式
鼠标光标显示/不显示	CTRL+B	显示状态
观察相机头部抬头/低头的允许与禁止	CTRL+H	允许状态

表 1

键定义	运动类型	step	θ	α
UPARROW KEY	FORWARD	0.7	n/a	n/a
DOWNDARROW KEY	BACKWARD	0.5	n/a	n/a
LEFTARROW KEY	TURN_LEFT	n/a	0.3	n/a
RIGHTARROW KEY	TURN_RIGHT	n/a	0.3	n/a
U KEY	LOOK_UP	n/a	n/a	0.15
J KEY	LOOK_DOWN	n/a	n/a	0.15

表 2

鼠标所在区域	相机运动类型	step	θ	α
区域 1	FORWARD	speed	n/a	n/a
区域 2	BACKWARD	speed	n/a	n/a
区域 3	FORWARD, TURN_RIGHT	speed	rot	n/a
区域 4	BACKWARD, TURN_RIGHT	speed	rot	n/a
区域 5	TURN_RIGHT	n/a	rot	n/a
区域 6	TURN_RIGHT	n/a	rot	n/a
区域 7	FORWARD, TURN_LEFT	speed	rot	n/a
区域 8	BACKWARD, TURN_LEFT	speed	rot	n/a
区域 9	TURN_LEFT	n/a	rot	n/a
区域 10	TURN_LEFT	n/a	rot	n/a

表 3

热键定义	功能描述	热键定义	功能描述
↑ 上光标键	向前移动视点	f	雾浓度减小
↓ 下光标键	向后移动视点	F	雾浓度增加
← 左光标键	向左转动视点	O	自动开/关门窗功能的设置
→ 右光标键	向右转动视点	M	二维地图开/关
U	抬头	P	室外场景开/关
J	低头	T	调试信息输出开/关
L	直升机视点下降	r	减小材质中的 red 成份
H	直升机视点上升	R	增加材质中的 red 成份
PageUp	设置直升机视点	g	减小材质中的 green 成份
PageDown	复位直升机视点	G	增加材质中的 green 成份
Home	至某建筑物门前	b	减小材质中的 blue 成份
End	至某指定点前	B	增加材质中的 blue 成份
F1	二维地图相机变焦	C	云层的开/关
F2	二维地图相机变焦	CTRL+G	向前的碰撞检测开/关
F3	二维地图相机变焦	CTRL+R	向后的碰撞检测开/关
F4	二维地图相机变焦	CTRL+B	鼠标指针显示开/并
F5	二维地图相机变焦	CTRL+X	纹理绘制开/关
F7	透明算法一选择	CTRL+I	统计信息开/关
F8	透明算法二选择	CTRL+C	相机控制开/关
F9	雾化浓度设置	CTRL+H	观察相机开/关
1	将相机放置在某建筑一层	CTRL+P	线框绘制开/关
2	将相机放置在某建筑二层	CTRL+L	虚拟实体的放大
3	将相机放置在某建筑三层	CTRL+S	虚拟实体的缩小
4	将相机放置在某建筑四层	CTRL+M	虚拟实体的正向旋转
5	将相机放置在某建筑五层	CTRL+N	虚拟实体的负向旋转
a	增加日光的 ambient 值	CTRL+Z	雾化开/关
A	减少日光的 ambient 值	CTRL+T	纹理替换功能键
d	增加日光的 diffuse 值	‘-’	详细统计信息显示开/关
D	减少日光的 diffuse 值		
s	太阳升起		
S	太阳下落		

表 4

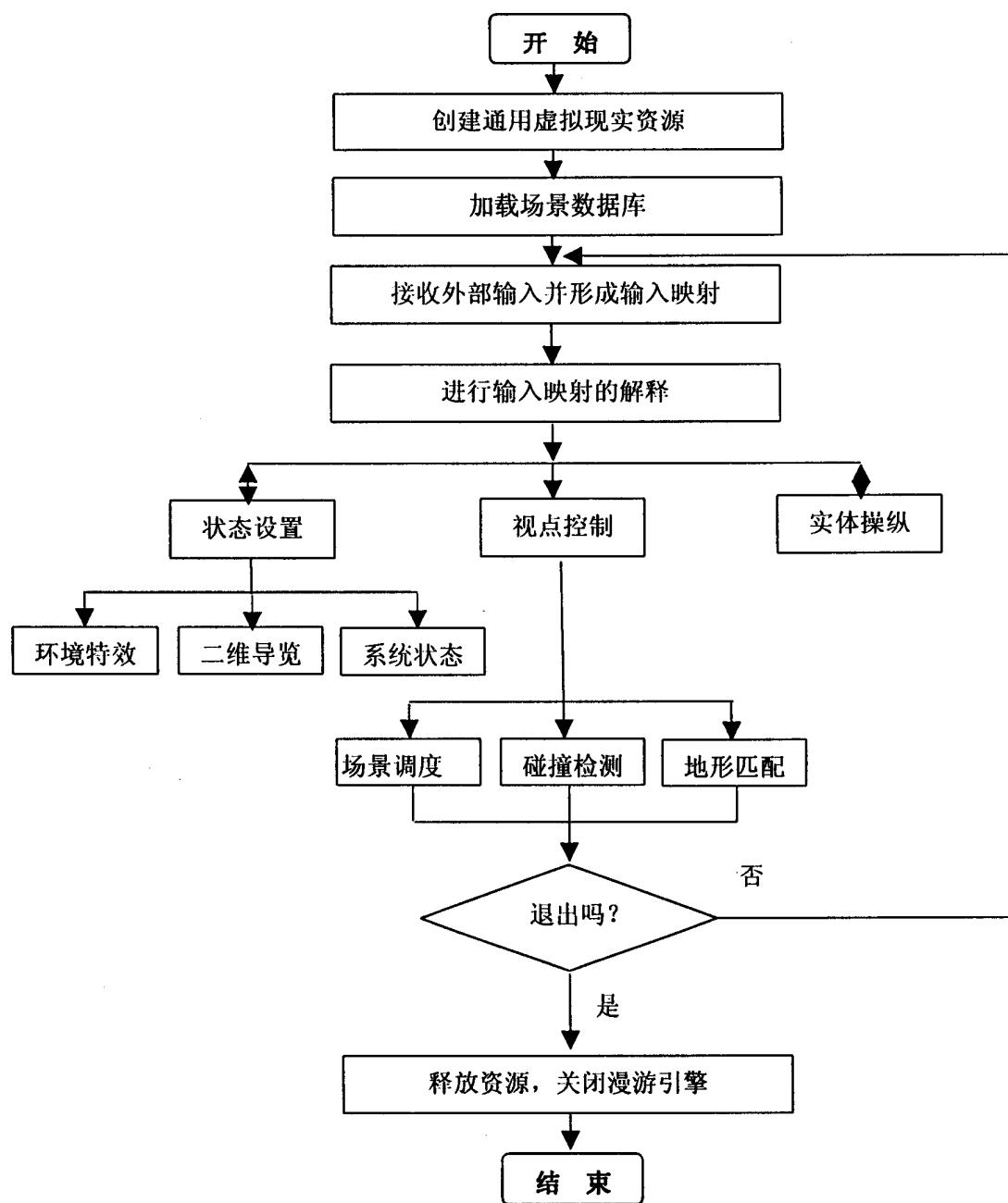


图 1

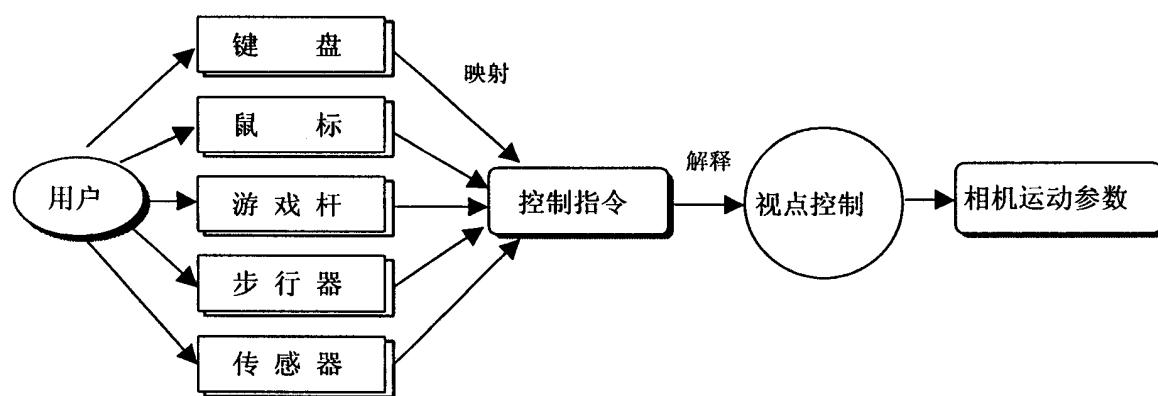


图 2

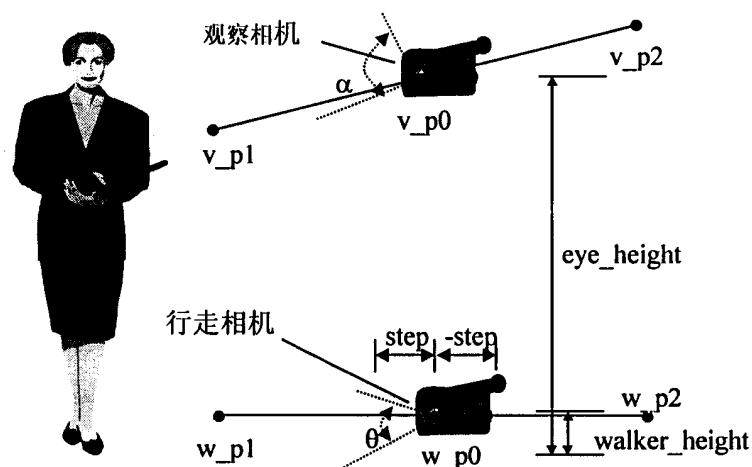


图 3

(1,1)

区域 9	区域 7	区域 1	区域 3	区域 5
区域 10	区域 8	(0, 0)	区域 4	区域 6

(-1,-1)

图 4

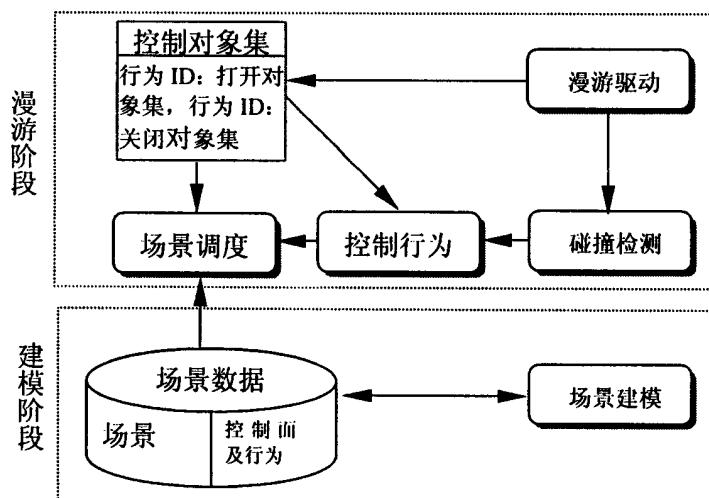


图 5

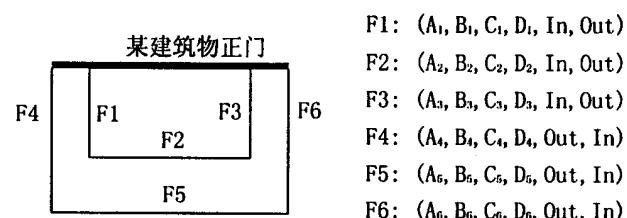


图 6

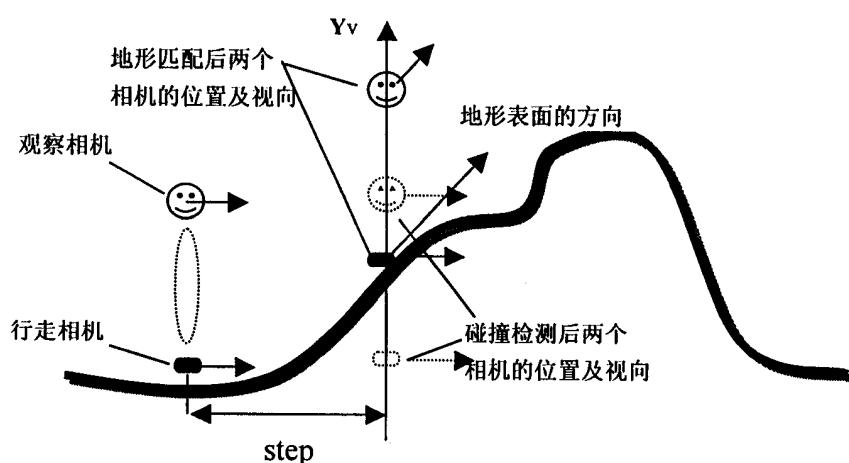


图 7

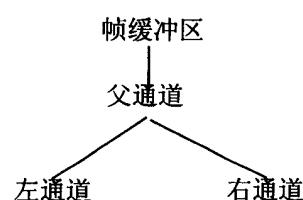


图 8

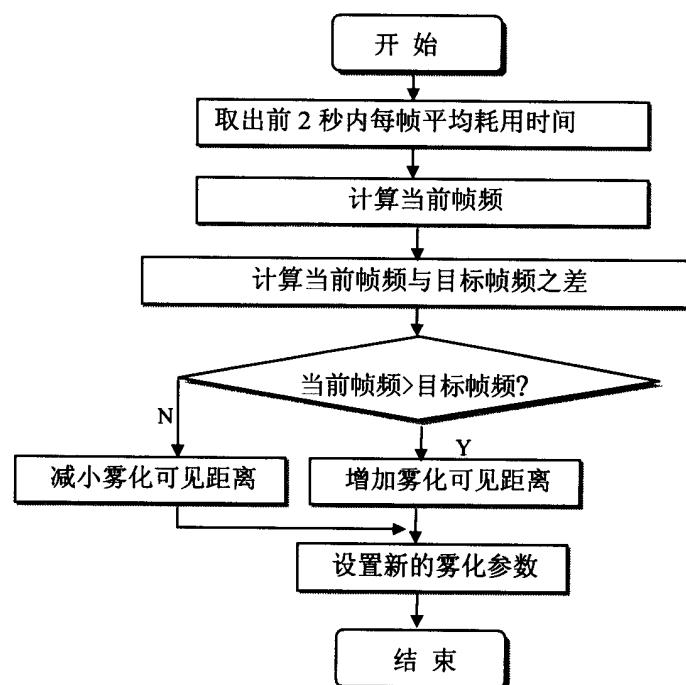


图 9