

爆炸装置排除视景仿真系统的设计与实现

李代伟¹ 孙建顺²

(成都信息工程学院软件工程学院 成都 610225)¹ (西南交通大学信息科学与技术学院 成都 610031)²

摘要 为解决当前爆炸装置排除训练过程中训练装置制作复杂,且每次重复制作,投入人力物力较大等问题,设计了爆炸装置排除视景仿真训练系统。以虚拟现实技术为基础对系统进行了层次结构划分与硬件结构设计,并运用面向对象思想对系统进行模块化设计,利用简单工厂、模板方法等设计模式对排爆工具模块进行扩展性设计;结合 ODE 物理引擎解决了装置节点间的碰撞检测,利用 OSG 相交性测试与矩阵变换解决了虚拟手抓取物体,通过正确排除顺序与用户排除顺序比较的方法解决了装置排除正确性判断等关键技术,以 VC++ 为开发环境实现了爆炸装置排除仿真训练系统。实践证明:该系统仿真度高,达到了爆炸装置排除仿真训练的效果,且易于操作,具有较强的扩展性。

关键词 虚拟现实,视景仿真,排爆训练,虚拟手抓取

中图分类号 TP391.9 **文献标识码** A

Design and Implementation of Explosive Devices Exclude Virtual Simulation Training System

LI Dai-wei¹ SUN Jian-shun²

(College of Software Engineering, Chengdu University of Information Technology, Chengdu 610225, China)¹

(School of Information Science & Technology, Southwest Jiaotong University, Chengdu 610031, China)²

Abstract To solve the problems of complication and reproduction of making training devices and huge cost of human and material resources in the explosive devices exclude training process, the paper designed an explosive devices exclude virtual simulation training system. The divisions of system hierarchical structure and the hardware architecture were designed based on the virtual reality technology, and the modular design of the system was based on object-oriented thinking. The paper designed the explosive devices exclude tool module according to the design pattern of simple factory and template method. It solves the collision detection between the device nodes with Open Dynamic Engine (ODE), virtual hand grab objects with the Open Scene Graph (OSG) intersects testing and matrix transformations and the judgment of device excluding correctness key technology. And it implements the explosive devices exclude virtual simulation training system under the VC++ development environment. Practice proves that the system has high degree of simulation, reaches the effect of the explosive device exclude simulation training. And it is easy to operate with a strong scalability.

Keywords Virtual reality, Visual simulation, Explosive devices disposal training, Virtual hand crawl

视景仿真又称虚拟仿真、虚拟现实仿真,是计算机技术、图形图像技术等多种高科技的结合,已经广泛应用于模拟训练、虚拟现实及其它实时动态仿真系统的图形显示^[1-3]。传统的爆炸装置排除训练需要事先有专门人员制作各种装置实物模型,每次讲解时需要将其拆分,每位参训人员训练时又需要将其拆分,在此过程中存在大量的重复性工作^[4]。因此,采用视景仿真技术开发一套高仿真的视景仿真训练系统,可以大幅缩减训练成本,提高训练效率。

根据爆炸装置组成与排爆工具,论文设计了视景仿真训练系统层次结构,运用面向对象思想设计了装置组成模块、排爆工具模块等软件部分,设计中采用多种设计模式使系统具有较强的可扩展性。系统硬件设备采用桌面型虚拟现实系统^[5],并利用开源图形引擎 Open Scene Graph (OSG)与开源物理引擎 Open Dynamic Engine (ODE)等视景仿真软件开发工具实现了该系统。

1 设计思想及系统总体方案

1.1 设计思想

爆炸装置排除是一个借助排爆工具拆卸装置,分离炸药从而解除危险的过程。在排爆过程中要求排爆操作人员熟练掌握装置结构的组成情况,并能够正确判断装置的发火原理,从而采取正确的排爆方法进行排爆操作。将爆炸装置视为由若干部件组合而成,通过适当的排爆工具对各个部件进行排除,在排爆过程中必须按照正确的顺序进行,否则会引爆装置。

借助现有视景仿真技术,利用数据手套、位置跟踪器等硬件设备,结合图形引擎与物理引擎模拟爆炸装置拆卸的过程。通过对爆炸装置、排爆工具进行三维建模,运用数据库技术存储爆炸装置、排爆顺序等信息,设计并实现了爆炸装置排除视景仿真系统。同时,考虑了系统使用的通用性与可扩展性,对

系统进行层次划分,运用面向对象思想进行了模块化设计^[6],以实现任意结构爆炸装置模型的支持以及排爆工具的扩展。

1.2 层次结构划分

一个完整的虚拟现实仿真系统由硬件设备及驱动、数据库存储及访问、图形引擎、物体引擎、用户应用等软硬件组成。考虑到系统的扩展性、可维护性以及稳定性,将系统分为应用层、引擎层和数据层,系统层次结构如图 1 所示。

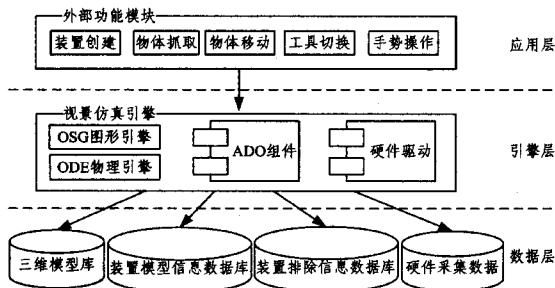


图 1 系统层次结构

(1)应用层

应用层是包含系统功能的层级,只维护具体功能的业务逻辑及其具体实现,包括装置创建、物体抓取、物体移动、工具切换、手势操作等功能。功能的实现基于引擎层提供的服务。

(2)引擎层

引擎层介于应用层与数据层之间,其主要任务是读取底层数据信息并加以封装,为应用层提供底层服务。该层包括 OSG 图形引擎、ODE 物理引擎、ADO 组件与硬件驱动程序,其中 OSG 图形引擎提供场景图形显示服务,ODE 物理引擎提供碰撞检测算法,ADO 组件用于访问数据库中的数据信息,硬件驱动程序用于读取硬件实时数据。

(3)数据层

数据层包括三维模型库、装置模型信息数据库、装置排除信息数据库与硬件采集数据。其中,三维模型库以三维模型文件形式存储装置模型、排除工具模型;装置模型信息数据库存储描述爆炸装置信息,包括装置模型文件路径、装置模型组成结构及各节点名称等;装置排除信息数据库存储排爆工具信息、装置排除顺序信息、装置节点排除方式等;硬件采集数据包括位置跟踪器实时位置姿态信息、数据手套手指状态信息。

通过以上系统层次结构划分,能有效地解除底层数据与高层应用之间的耦合。随着系统的使用,会产生设备升级、模型文件格式更改等底层数据变化问题,而此设计的底层数据变化不会影响到高层应用;同样随着用户需求的变化,高层应用也存在很多不确定性因素,此设计的高层应用变化对系统的影响只局限于应用层;随着图形引擎与碰撞检测等技术的发展,在引擎层中可以更换系统的图形引擎、物理引擎,更改数据库访问组件、升级硬件驱动。

1.3 硬件构建

爆炸装置排除仿真训练侧重虚拟操作,系统选用了交互操作性强、开发成本低的桌面型虚拟现实系统的硬件组成^[7],其硬件组成结构如图 2 所示,主要包含以下设备:

- (1)高性能个人计算机,采用主流配置高性能个人计算机,并要求有一款支持三维显示的专业级图形显卡。
- (2)立体显示器,使系统能够实现桌面三维立体效果。

(3)能获取全方位旋转信息的位置跟踪器,模拟操作时跟踪虚拟排爆工具的位置与姿态。

(4)数据手套,虚拟手操作时,获取手指状态信息。

(5)立体眼镜,观察三维立体显示模式。

(6)音效输出设备,输出系统排爆中产生的一系列音效。

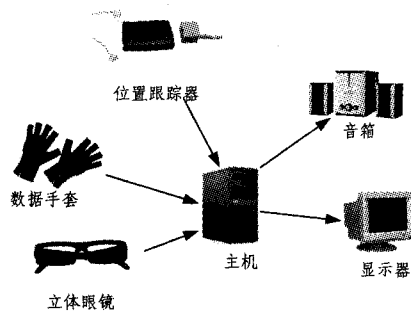


图 2 系统硬件结构示意图

2 核心模块软件设计

根据面向对象的思想将系统划分为交互界面、爆炸装置、排爆工具和辅助信息 4 个核心模块,其结构如图 3 所示。交互界面依赖于爆炸装置、排爆工具和辅助信息模块,为用户提供可视化的图形界面。爆炸装置、排爆工具与辅助信息作为独立模块存在,相互之间不存在直接依赖关系,其业务逻辑关系由交互界面负责维护。

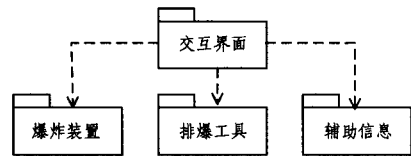


图 3 系统核心模块结构

2.1 爆炸装置模块

将爆炸装置视为一组可以拆卸的部件拼装而成,故划分为装置类与节点类,且它们是整体与部分的关系,两者之间以组合方式关联。考虑到装置与节点在虚拟场景中均可被抓取移动,故从位置姿态变换类 OSG 中派生出装置类 CSGWorld、节点类 CSGNode 和几何数据类 CSGShap。CSGShap 类负责读取装置三维模型以及负责装置三维模型的一些图形显示特性。在交互界面模块中设计了 CSGOSGCore 类,其通过与 CSGWorld 的组合关系实现与爆炸装置模块的依赖。

2.2 排爆工具模块

爆炸装置排除过程中需用到很多工具,系统只对常用工具进行模拟,并运用简单工厂与模板方法设计模式^[8]对工具类进行设计,能够容易地对排爆工具类进行扩展。排爆工具模块划分为排爆工具相关类、位置跟踪器相关类、数据手套相关类。

排爆工具相关类,包括抽象工具基类 CSGTool 及其子类。采用模板方法设计模式设计,业务逻辑均在基类 CSGTool 中指定,而具体实现方法由其子类实现。运用简单工厂设计模式,将排爆工具的创建交给 CSGToolFactory 类负责,并在其中添加新工具类型对象创建方法,为系统增加一种新的排爆工具。

位置跟踪器相关类,包括 CSGTrackerUpdateCallback 类与 CSGTracker 类。CSGTrackerUpdateCallback 类派生自

OSG 节点回调类 NodeCallback, 该类在每一帧场景绘制过程中均会被回调, 在回调时更新位置跟踪器的数据, 实现位置跟踪器的输入交互。CSGTracker 类负责获取位置跟踪器硬件设备数据, 处理硬件采集的原始数据, 并为上层调用提供数据读取接口。

数据手套相关类, 包括 CSGGloveUpdateCallback 类与 CSGGlove 类。CSGGloveUpdateCallback 类派生自 OSG 节点回调类 NodeCallback, 在每一帧场景绘制中, 通过回调机制读取数据手套数据, 更新数据手套状态, 从而实现数据手套输入交互。CSGGlove 类负责获取数据手套硬件设备数据, 处理硬件采集的原始数据, 并为上层调用提供数据读取接口。

2.3 辅助信息模块

为了提高系统界面可视化效果, 在场景中增加了坐标轴显示与隐藏、地面显示与隐藏、灯光开启与关闭、被抓取节点名称提示、被抓取节点特效显示、节点间碰撞特效显现等辅助信息。在场景中设计了可通过虚拟手直接点选的工具栏类 CSGBaseBar、按钮类 CSGBarButton 和边框类 CSGBarFrame, 它们均派生自 OSG 位置姿态变换类 PositionAttitudeTransform, 以几种简单的虚拟手势操作来增强系统的可操作性。

3 关键技术

3.1 装置节点间的碰撞检测

为了提高爆炸装置排除过程中的真实感, 系统实现了爆炸装置节点间的碰撞检测。综合考虑爆炸装置部件以刚体为主及柔体的碰撞检测模拟计算代价大两个因素, 将所有柔体部件分切为小块刚体处理。如图 4 所示, 装置模型中胶带为柔体, 系统在建模时将其分切为胶带碎片进行模拟, 所有胶带碎片均为刚体。通过该方法, 系统只需要计算刚体间的碰撞检测。

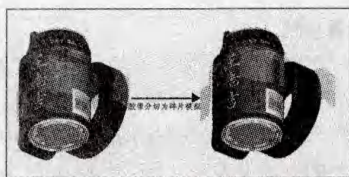


图 4 柔体分切为刚体碎片示意图

刚体间的碰撞检测在视景仿真领域已经是一项比较成熟的技术, 系统借助开源物理引擎 ODE^[9] 来计算刚体间的碰撞检测。在 ODE 中, 刚体 dBody 是个物理概念, 不包含几何数据, 所有位于同一碰撞检测空间 dSpace 中的刚体之间需要进行碰撞检测计算, 碰撞检测空间又位于 ODE 物理世界 dWorld 中。ODE 中刚体 dBody 的几何数据存储于 dGeom 中, ODE 物理引擎只涉及物理模型的计算, 并不能提供图形显示, 系统的图形显示采用了 OSG 图形引擎, 因此系统在模拟爆炸装置节点间的碰撞时需将 ODE 与 OSG 结合。系统根据 ODE 中 dWorld、dSpace、dBody 与 dGeom, 设计了爆炸装置结构, 如图 5 所示。

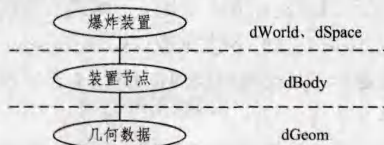


图 5 装置结构示意图

装置结构为一个三层结构: 第一层为整个装置, 对应于 dWorld 和 dSpace; 第二层为装置节点, 对应于 dBody; 第三层为节点几何数据, 对应于 dGeom。结合 OSG 图形引擎设计了爆炸装置类结构, 包括 CSGWorld 类、CSGNode 类和 CSGShape 类。CSGWorld 类负责维护 ODE 中物理世界 dWorld 与碰撞检测空间 dSpace, CSGNode 类负责维护 ODE 中碰撞刚体 dBody, CSGShape 类负责维护 ODE 中几何数据 dGeom, CSGNode 类和 CSGShape 类都保存碰撞检测空间 dSpace 的一个引用。

3.2 虚拟手抓取物体

虚拟手抓取物体, 是排爆模拟的核心技术, 旨在解决虚拟手抓取物体的方式与物体被抓取后跟随虚拟手运动的问题^[10,11]。系统场景图形开发基于 OSG 图形引擎, 借助 OSG 提供的相交性测试、矩阵变换等相关技术来解决。

3.2.1 虚拟手抓取方式

系统设计了两种抓取手势以弥补装置的组成部件体积不均、大小形状各异的缺陷, 如图 6 所示。对于体积较小的物体, 用点取物体手势抓取物体, 该手势食指伸直, 其余四指弯曲, 利用食指第一指节与物体相交进行抓取; 对于体积较大物体, 用抓取物体手势抓取物体, 该手势五指均弯曲, 利用食指第一指节或者第二指节与物体相交进行抓取。

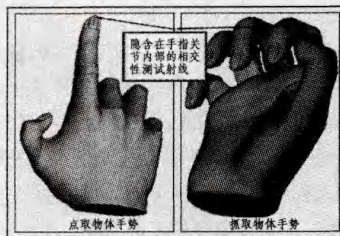


图 6 虚拟手抓取物体手势示意图

虚拟手与物体的接触判断, 利用了 OSG 底层提供的射线相交性测试机制。OSG 提供了射线类 osg::LineSegment 与交集访问器类 osgUtil::IntersectVisitor, 其使用步骤如下^[12]:

- (1) 创建一个 LineSegment 实例, 它使用两个 Vec3 实例来定义交集测试所用射线的起点和终点;
- (2) 创建一个 IntersectVisitor 实例;
- (3) 将 LineSegment 实例添加到 IntersectVisitor 实例中;
- (4) 初始化 IntersectVisitor 实例, 使其从场景图形中适当的节点开始遍历;
- (5) 获取交集测试结果的世界坐标。

根据以上原理, 系统设计了射线类 CSGCatchLine, 该类拥有 LineSegment 对象与 IntersectVisitor 对象, 以组合方式与虚拟手等排爆工具类进行关联。

3.2.2 被抓取物体跟随虚拟手运动

OSG 中场景图形以树形结构组织, 树节点坐标以一个四维矩阵描述, 如式(1)所示:

$$M = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \\ m_{41} & m_{42} & m_{43} & m_{44} \end{bmatrix} \quad (1)$$

每个子节点坐标值均相对于父坐标系。虚拟手节点与需

要抓取的装置部件节点在场景树中并不是直接父子关系,当虚拟手抓取到装置部件之后,要求被抓取部件跟随虚拟手运动,进行一系列矩阵变换计算后,求解正确的运动矩阵。场景树结构示意图如图7所示。

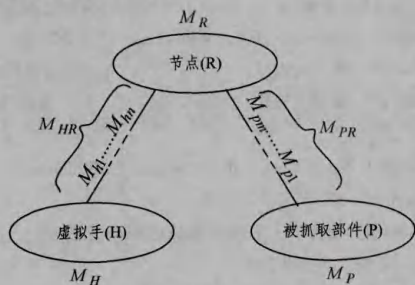


图7 场景树结构示意图

OSG 场景数据中只存在一棵场景树,虚拟手节点 H 与被抓取部件节点 P 必定拥有一共同祖先节点 R 。从节点 R 到节点 H 需要经历若干中间节点,通过遍历求得节点 H 相对于节点 R 坐标系的矩阵,记为 M_{HR} ,求解方法如式(2)所示。

$$M_{HR} = \prod_{i=n}^{i=0} M_i \quad (2)$$

同理,从节点 R 到节点 P ,节点 P 相对于节点 R 坐标系的矩阵记为 M_{PR} ,求解方法如式(3)所示。

$$M_{PR} = \prod_{i=m}^{i=0} M_i \quad (3)$$

根据以上计算得到了 H 与 P 在 R 坐标系下的矩阵表示,要求被抓取部件跟随虚拟手运动,将被抓取部件节点 P 的矩阵表示变换到虚拟手节点 H 的坐标系下,使得 P 节点相当于 H 节点的直接子节点,从而实现 P 节点跟随 H 节点运动,并保证其正确性。该变换的计算方法如式(4)所示。

$$M_P' = M_P \cdot M_{HR}^{-1} \cdot M_{PR} \quad (4)$$

设节点 P 自身坐标矩阵为 M_P (即 OSG 中默认的相对于其直接父节点的矩阵),在节点 H 坐标系下的矩阵 M_P' ,先消去从节点 H 在节点 R 坐标系下的相对路径,即乘以 M_{HR} 的逆矩阵,然后附上节点 P 在节点 R 坐标系下的相对路径,即乘以 M_{PR} ,最后用得到的矩阵 M_P' 取代节点 P 自身矩阵 M_P 。当被抓取部件释放后,将该节点当前的矩阵 M_P' 以其自身矩阵 M_P 取代。通过上述方法实现被抓取部件跟随虚拟手运动以及释放被抓取部件。

3.3 装置部件排除顺序算法描述

爆炸装置排除训练的主要目标之一是要要求参训人员掌握装置结构,在排爆操作中必须以正确排除顺序对装置进行拆卸。系统设计了装置部件排除顺序判断功能,排除顺序判断算法描述如下:

(1)给每个节点添加两个属性:正确排除顺序、用户排除顺序,其中正确排除顺序数值来自系统数据库,用户排除顺序记录当前用户操作时排除该节点的顺序。

(2)正确排除顺序,由系统管理在装置录入系统时进行设定。例如,某装置拥有 A, B, C, D, E, F, G 7 个节点,其中 A, C, D 可以进行同时排除, B, E 可以进行同时排除, F, G 可以同时排除,而 A, C, D 必须早于 B, E 被排除, B, E 又必须早于 F, G 被排除。则可以为以上 7 个节点给定正确的排除顺序为: $A-1, B-4, C-1, D-1, E-4, F-6, G-6$,其中数字表示被排除顺序。

(3)用户排除顺序,在用户进行排除操作时由系统根据用

户操作给定。定义一变量记录已经被排除节点的数目,当某节点被排除时,将该变量的值赋给用户排除顺序。以上面步骤(2)中装置为例,节点排除顺序如表1所列。

表1 装置节点排除顺序

装置节点 (name)	正确排除顺序 (sysOrder)	用户排除顺序 (userOrder)
A	1	1
B	4	4
C	1	2
D	1	3
E	4	6
F	6	5
G	6	7

(4)正确性判断。在用户每次排除完一个节点后,遍历该装置所有节点,对正确排除顺序(sysOrder)与用户排除顺序(userOrder)进行比较。若存在 $userOrder < sysOrder$,则判定用户在排除该节点时顺序错误;若所有已排除节点均满足 $userOrder \geq sysOrder$,则判定所有已排除节点均按照正确顺序排除。例如,在表1中 F 节点 $userOrder < sysOrder$,则该装置排除失败。

(5)排除结束判定。若在排除过程中出现排除顺序错误,则排爆失败,结束排爆;若记录的已排除节点数目等于装置需要被排除节点总数目,并且未出现排除顺序错误,则排爆成功,结束排爆。

4 仿真效果

实验在 Intel Core i7-920 处理器、2G 内存、Nvidia Quadro 4600 显示接口卡、Windows XP Professional SP3 操作系统的计算机上进行。利用 Visual C++ 2005、3DS MAX 2009、Microsoft SQL Server 2000 等开发工具,采用 OSG、ODE 等技术,使用 OpenGL 图形 API 实现了爆炸装置排除视景仿真训练系统^[13],系统运行流畅,在爆炸装置部件 30 至 50 块范围内,图形渲染的帧率达到 60fps~140fps。系统运行部分效果如图8所示,其中给出了辅助信息使用效果、剪刀工具使用模拟效果、虚拟手抓取模拟效果、装置爆炸模拟效果。

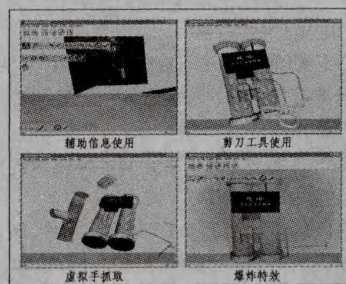


图8 仿真效果图

结束语 论文针对爆炸装置排除训练,借助现有虚拟现实技术设计并实现了一套爆炸装置排除视景仿真训练系统,描述了系统架构设计、系统硬件设计、系统核心软件设计,并详细阐述了在实现该系统过程中采用的关键技术。该系统达到了爆炸装置排除模拟训练效果,具有较高实用价值,可为爆炸装置排除视景仿真训练模拟的研究提供一定的参考。

参考文献

[1] Revealed D D, the headset that will mimic all five senses and

make the virtual world as convincing as real life [N]. Dailymail, 2009-03-05

- [2] 孙鹏,等. 基于 Vega 的高炮外弹道视景仿真系统设计与实现[J]. 弹箭与制导学报, 2011, 31(6): 215-218
- [3] 陈浩磊,等. 虚拟现实技术的最新发展与展望[J]. 中国科技论文在线, 2011, 6(1): 1-4
- [4] 王新建. 反恐活动中的若干爆炸问题研究[J]. 中国人民公安大学学报: 自然科学版, 2006(4): 68-70
- [5] 刘光然. 虚拟现实技术[M]. 北京: 清华大学出版社, 2011: 5-10
- [6] 孙建顺,等. 爆炸装置搜索视景仿真训练系统的设计与实现[J]. 成都信息工程学院学报, 2012, 27(4): 390-395
- [7] Sherman W R, Craig A B. Understanding Virtual Reality[M]. Publishing House of Electronics Industry, 2004: 290-296
- [8] Gamma E, Helm R, Johnson R, et al. Design Patterns, Elements of Reusable Object-Oriented Software [M]. Addison Wesley

Longman, Inc, 1995

- [9] 有人, 陈定云. 虚拟现实环境中的物理模拟及物理引擎应用研究[J]. 湖北工业大学学报, 2008, 23(2): 7-9
- [10] Masco M B. Grasping and moving objects in virtual environments; a preliminary approach towards a realistic behavior[C]// Toshio Fukuda ed. RO-MAN 94, Proc 3rd IEEE Int Workshop on Robot and Human Communication. Nagoya, 1994: 44-49
- [11] 李克彬,等. 虚拟手操作技术研究及实现[J]. 计算机仿真, 2007, 24(3): 227-230
- [12] Sullivan J. Naval Postgraduate School OSG Tutorial[Z]. Naval Postgraduate School, 2006: 259
- [13] 华泽玺,等. 基于粒子系统的爆炸效果仿真研究[J]. 计算机科学, 2012, 39(4): 278-281
- [14] 赵国锋, 陶文平. 可编程路由器的虚拟化技术研究[J]. 重庆邮电大学学报: 自然科学版, 2013, 25(1): 20-23

(上接第 301 页)

GA-PSO Hough 变换来探测平面表面。

3D 空间中的平面可以参数化为如下形式:

$$x \cos \alpha + y \cos \beta + z \cos \gamma = d \quad (4)$$

式中, $\alpha \in [0, \pi]$, $\beta \in [0, \pi]$, $\gamma \in [0, \pi/2]$ 表示平面的两条法向量之间的夹角与 3 个坐标轴, 同时:

$$\cos^2 \alpha + \cos^2 \beta + \cos^2 \gamma = 1 \quad (5)$$

$n = (\cos \alpha, \cos \beta, \cos \gamma)$ 为平面的法向量, d 为来自坐标系到平面的距离。式(4)显示在式(5)中的变量 γ 可以用 α 和 β 进行描述, 因此, 在平面公式中有 3 个独立的参数 α, β 和 d 。

GA-PSO HT 平面探测方法的步骤如下:

1. 适当地将参数空间 (α, β, d) 量化进 $P \times P \times Q$ 中, 将一个 (α, β, d) 的三维卷积数组初始化为 0。

2. 对图像中的每个点 (x, y, z) , 参数平面中所有的角二元组 (α, β) 表示为 $\{(\alpha_i, \beta_j) \mid i, j = 0, 1, \dots, p-1\}$ 。对每个二元组 (α_i, β_j) , 根据式(4)与式(5), 可以计算其对应的参数 d , 表示为 d_k 。然后, 在三元组 (α_i, β_j, d_k) 的卷积数组上增加一个量。

3. 在考虑通过所有点的所有角二元组后, 在卷积数组中拥有较高值的一个参数单元可能与图像中的一个平面相对应。

基于 GA-PSO HT 方法可能会存在如下不足:

a) 由于需要存储 $P \times P \times Q$ 个单元的三维卷积数组, 因此有巨大的内存需求。

b) 比较高的计算复杂度, 对于 $M \times N$ 图像进行 Hough 变换的计算复杂度为 $O(MNP^2Q)$ 。

c) 参数空间的量化步骤会影响计算精度, 较少的量化步骤可以提高精度, 但需要大量的存储与计算成本。

3.2 真实图像的执行过程

图 2 显示了原始图像与从图像中提取的边图像, 测试图像的尺寸为 768×768 像素。对于 GA-PSO HT 执行性能与其他 HT 算法的比较评估如表 1 所列。

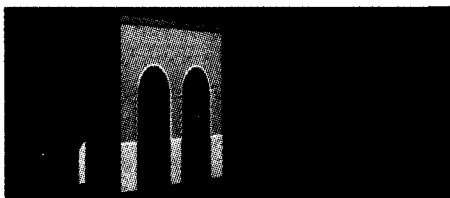


图 2 房子及其平面图像

表 1 GA-PSO HT 和 RHT 的比较

DEVIATION	GA-PSO RT(S)	RH(S)
10	0.2038	0.2593
50	0.6529	0.9368
100	1.2031	2.2575

结果显示, GA-PSO HT 与通常的 HT 算法在建筑物图像的边探测方面具有较高的性能与精度。

结束语 本文提出了一种基于 GA-PSO Hough 变换的建筑物平面检测方法, 该方法的主要优点是以前的技术需要监督, 而本文的算法是无监督学习, 因此对于大规模数字城市乃至数字地球的构建更具优越性。

参考文献

- [1] Brown L G. A survey of image registration techniques[J]. ACM Comput. Surv., 1992, 24(4): 325-376
- [2] Tanimoto S L. Template matching in pyramids [J]. Comput. Graphics Image Process, 1981(16): 356-369
- [3] Lai S-H, Fang M. Robust and efficient image alignment with spatially varying illumination models[C]// Proceedings of the Computer Vision and Pattern Recognition. 1999, II: 167-172
- [4] Olson C F, Huttenlocher D P. Automatic target recognition by matching oriented edge pixels[J]. IEEE Trans. Image Process, 1997(61): 103-113
- [5] Ballard D H. Generalizing the Hough transform to detect arbitrary shapes[J]. Pattern Recognition, 1981, 13(2): 111-122
- [6] Davis L S. Hierarchical generalized Hough transforms and line-segment based generalized Hough transforms[J]. Pattern Recognition, 1982, 15(4): 277-285
- [7] Cantoni V, Carrioli L. Structural shape recognition in a multiresolution environment[J]. Signal Process, 1987(12): 267-276
- [8] Lu W, Tan J L. Detection of incomplete ellipse in images with strong noise by iterative randomized Hough transform (IRHT) [J]. Pattern Recognition, 2008, 41(4): 1268-1279
- [9] Torii A, Imiya A. The randomized-Hough-transform-based method for great-circle detection on sphere[J]. Pattern Recognition Letters, 2007, 28(10): 1186-1192
- [10] Duquenoey E, Taleb-Ahmed A. Applying the Hough transform pseudo-linearity property to improve computing speed[J]. Pattern Recognition Letters, 2006, 27(16): 1893-1904