

面向拆装训练的输油泵机组虚拟维修系统设计

蔡凯骏¹, 张伟明¹, 赵鸿飞², 师磊²

(1. 陆军勤务学院 油料系, 重庆 401331; 2. 中国人民解放军 31605 部队, 南京 211100)

摘要:针对输油泵机组实装维修训练存在的不足, 设计了面向拆卸与装配训练的输油泵机组虚拟维修系统; 通过现有主流虚拟现实仿真平台的对比, 选择以 Unity3D 作为开发平台; 分析了系统功能、性能需求, 设计了系统多层次模块化训练体系结构; 针对系统交互式拆装的核心功能需求, 提出一种高效的鼠标拖拽时零部件位置更新算法; 为在虚拟训练过程中正确模拟机械零部件拆装的顺序关系, 研究了基于事件驱动编程的设计方法; 对系统实际运行参数进行测试, 满足各项功能、性能需求; 系统不仅满足输油泵机组维修训练的需求, 还可为其它产品的维修拆装训练系统设计提供参考。

关键词:虚拟维修; 输油泵机组; 拆卸与装配; 事件驱动

Design of Petroleum Pumping Unit Virtual Maintenance System for Disassemble and Assemble Training

Cai Kaijun¹, Zhang Weiming¹, Zhao Hongfei², Shi Lei²

(1. Department of Oil, Army Logistics University of PLA, Chongqing 401331, China;

2. No. 31605 PLA Troops, Nanjing 211100, China)

Abstract: Aiming to overcome the deficiency of petroleum pumping unit real-equipment based training, design the virtual maintenance system for disassemble and assemble training. Through comparison of existing VR simulation platforms, chose Unity3D as the system develop platform. The functional and performance requirements of the system was analyzed, the multilevel modular framework of system was designed. To realize the system key requirements—interactive disassemble and assemble, the part position update algorithm was proposed. To simulate the parts' assembly and disassembly order in virtual training, the design method based on event-driven was studied. The system practice running parameters meet the functional and performance requirements. It can not only satisfy the demand of petroleum pumping unit maintenance training, but also provide reference for disassemble and assemble training system design of other products.

Keywords: virtual maintenance; petroleum pumping unit; assembly and disassembly; event-driven

0 引言

输油泵机组是输油管线系统的核心设备, 主要包括发动机、离心泵和增速器等, 在石油化工和军事供油领域具有重要应用。正确、熟练的维修活动是保持泵机组较高可用度的关键, 其中拆卸与装配是所有维修活动中普遍存在的步骤, 因此针对使用人员的维修拆装训练尤为重要。传统维修拆装训练主要依靠实装进行, 受场地、时间、费用等因素限制, 训练效果受到一定影响。随着虚拟现实技术的不断发展与成熟, 其在维修领域的应用研究引起了广泛关注^[1]。Abate 等^[2]开发了面向航天领域的虚拟训练系统, Jayaram 等^[3]研究了人体工效学虚拟评估系统, 杨新红等^[4]分别开发了车辆虚拟实验系统。但是, 上述系统针对产品的结构展示与操作, 受训者主要采用观看以及少量操作的

方式参与训练。而针对输油泵机组维修拆装的虚拟训练系统需要受训者利用鼠标等设备进行大量操作以达到较好的训练效果, 系统的交互性明显增强。邹俞等^[5]、崔庆春等^[6]分别基于 Kinect 和 OSG 开发了沉浸式的交互式训练系统。但是沉浸式系统的造价高昂, 操作精度受设备影响。与之相比, 桌面式系统成本低、易于开发且操作方便, 适合面向拆装训练的虚拟维修系统。因此, 本文研究了面向输油泵机组虚拟拆卸与装配训练的关键技术并设计了相应的桌面式虚拟维修系统。

1 仿真平台对比

仿真平台是虚拟维修训练系统实现拟实视觉效果和人机交互的基础。当前图形程序接口 OpenGL 以及面向 Web 三维虚拟仿真的 Java 3D、VRML 等可用于构建虚拟维修仿真平台^[7], 但是利用上述编程语言开发效率较低, 难度较大。现有的成熟商用虚拟现实软件由于易学易用易获取且性能较好, 被广泛应用于虚拟维修训练系统的设计与开发^[8-13], 总结各软件平台的功能、特点及其适合应用领域如表 1 所示。

面向拆卸与装配的输油泵机组虚拟维修训练系统需要较

收稿日期:2018-10-29; 修回日期:2018-11-20。

基金项目:陆军勤务学院研究生创新基金项目(20151230)。

作者简介:蔡凯骏(1990-), 男, 重庆市人, 博士研究生, 主要从事虚拟维修、装备保障信息化方向的研究。

张伟明(1958-), 男, 河南郑州人, 博士, 教授, 博士生导师, 主要从事油气储运工程, 军事装备学方向的研究。

表 1 不同虚拟维修仿真平台对比

名称	特点及功能	适用领域	视觉效果	人机交互设计
Unity3D	高度优化的渲染和高效物理引擎,高度整合且可扩展的编辑器,支持 C#、Javascript 语言,交互式设计方便,灵活易操作,跨平台性好,获取成本低。	军事国防,维修仿真,游戏开发,教育培训	好	支持
Jack、DELMIA	可为制造流程提供完整的数字解决方案,包含人机工程相关知识库,可方便地生成人员维修姿态。	产品设计,维修性评估,产品制造流程仿真	一般	不支持
Vega	可利用图形化的工具快速构建系统。功能单一,需用 OpenGL 扩充功能模块。	场景展示,教育培训	较好	支持
EON、Virtools	图形化编程易学易用,提供内置功能节点,通过节点连线的方式实现逻辑关系,支持利用 VBScript 和 JavaScript 开发自定义功能节点。	场景展示,教育培训	一般	支持
3DVIA Composer	使用简单,灵活易操作,提供大量功能模块。	产品文档编制,教育培训	较好	支持

强的交互性和较好的视觉效果。通过对比可以看出，Unity3D 具有视觉效果好，交互式设计方便，灵活易操作等突出优势，故选取 Unity3D 作为系统开发平台。

2 系统需求分析与结构体系设计

2.1 需求分析

系统主要作用是使输油泵机组使用、维修人员掌握设备维修拆装的要求与方法，提高维修能力与效率，因此对系统提出功能和性能两方面需求。

功能需求主要包括：

- 1) 能对设备常见保养、故障维修进行训练，能通过键盘、鼠标等实现交互式拆装。
- 2) 采集操作过程数据并考评。性能需求方面，系统能在所有 windows 系统中使用。为保证适用性，系统在具有中低等配置的计算机（处理器 Inter Core i5 2410M，图形处理器 NVIDIA GeForce GT 550M，内存 2 GB）运行时，场景渲染频率不小于 30FPS（Frames Per Second）。

2.2 多层次模块化训练结构体系

本文参考了国内外通用虚拟训练（General Virtual Environment, GVE）体系理念^[14-15]，针对输油泵机组使用过程中的维修情况，提出如图 1 所示的多层次模块化训练结构体系。系统主要包括维修训练，技能考核两大部分。

1) 维修训练按照从简到难的顺序，包括子装配体拆装训练、设备保养、故障维修、完全拆解与装配 4 个内容。受训者通过鼠标点击拖拽与键盘操作相结合完成相应维修拆装任务，交互性较强。

2) 技能考核考察受训者设备维修技能的掌握程度。获取、分析维修过程中各步骤完成时间、使用提示次数、错误操作次数等数据并进行打分。

3 交互式拆装

系统交互式拆装设计目的是使受训者经过自主判断后，可通过鼠标、键盘完成零部件的拆装、工具的拾取等操作，更好地模拟实际维修过程，主要功能如下：

- 1) 受训者移动鼠标光标至目标零件上的任意位置，按住左键并移动鼠标可实现拖拽效果。

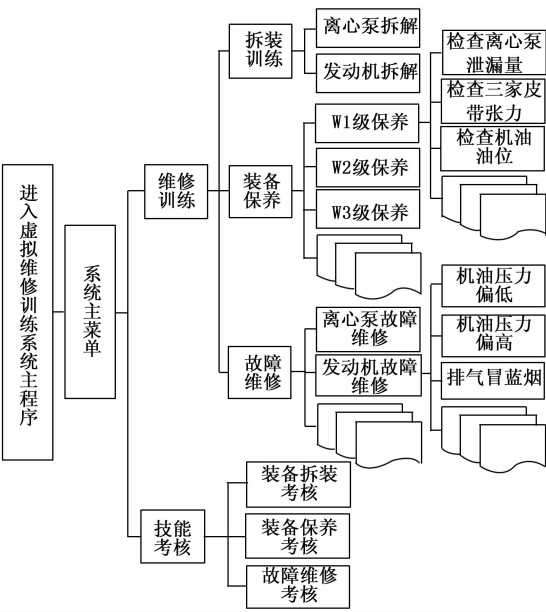


图 1 系统多层次模块化体系结构图

- 2) 零件在拖拽过程中始终沿着预定的拆装路径移动。
- 3) 零件移动时限于初始位置和目标位置之间。拖拽零件到达指定位置后，零件不能再拖动。

为实现上述功能，鼠标拖动时零件位置计算是关键，对此提出如下算法。

Step1：根据零件在世界坐标系的坐标值 P_{World} ，通过 WorldToScreenPoint 函数得到屏幕坐标系的坐标值 P_{Screen} 。

Step2：鼠标点击时计算鼠标光标位置与零件中心位置的偏差向量 d 。每次鼠标点击时 Unity 调用 OnMouseDown 函数，在该函数中将 P_{Screen} 中 Z 轴的值 $P_{Screen.z}$ 赋给 $M_{Screen.z}$ ，其中 M_{Screen} 为鼠标光标位置在屏幕坐标系的坐标值。通过 ScreenToWorldPoint 函数得到光标在世界坐标系的坐标值 M_{World} 。计算鼠标光标位置与零件中心位置的偏差向量 $d = P_{World} - M_{World}$ 。

Step3：计算零件完全随光标移动时在当前帧所在位置。Unity 提供的 OnMouseDown 函数在鼠标按下的每一帧调用。

由于鼠标按下时光标可能在零件上任意位置,而零件坐标为其中心点的位置,因此在每一帧更新零件位置时计算零件完全随光标移动时的位置 $P'_{World} = M'_{World} + d$, 其中 P'_{World}, M'_{World} 分别为当前帧零件和鼠标光标在世界坐标系的位置。

Step4: 使零件拖动过程始终沿着预定方向移动。计算光标当前帧的移动向量 $l = M'_{World} - M''_{World}$, 其中 M''_{World} 为前一帧光标的坐标。计算 l 沿预定方向向量 n 的投影向量 $l' = \left(\frac{l \cdot n}{|n|} \right) n$, 则零件当前帧位置为 $P'_{World} = l' + P''_{World}$ 。

Step5: 保证零件移动时限于初始位置和目标位置之间。计算零件当前帧位置相对初始位置的向量 $v = P'_{World} - P_{OriWorld}$, 其中 $P_{OriWorld}$ 为零件初始位置在世界坐标系的坐标值。计算 v 与 n 的点积 $c = v \cdot n$, 若 $c > 0$, 更新零件位置为 $P'_{World}, P'_{World} = P'_{World}$, 否则不更新零件位置。

Step6: 判断零件是否到达目标位置。若 P'_{World} 与目标位置 P_d 相同, 停止 OnMouseDown 函数的调用, 算法结束。否则, 若鼠标左键松开, 返回 Step2, 若鼠标左键仍处于按下状态, 返回 Step3。

图 2 为零件位置计算示意图, 图 3 为算法流程图。

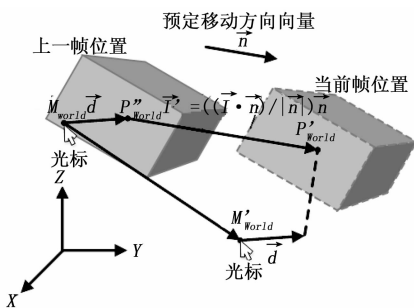


图 2 零件位置计算示意图

4 事件驱动设计

拆卸与装配过程中, 由于装配关系的约束, 待拆装零件满足一定的拆装顺序。因此对于某一零件, 需根据其它相关零件的状态以决定其是否可拆装。Unity 提供了 Update 函数在系统运行的每一帧调用一次。因此在该函数中利用条件判断语句可实现零件状态的实时监控, 是一种较为方便的方法。但是当某一维修过程涉及零部件数量众多时, 利用 Update 函数监控所有零部件状态会造成大量计算资源的浪费, 严重时造成运行画面的卡顿现象。对此本文研究了一种基于事件驱动的改进方案。

事件驱动的实现主要依赖以下 3 个核心部分:

1) 事件监听器: 事件监听器需要被告知事件是否发生, 因此将根据其它零件的拆装状态来决定自己是否可拆装的零件定义为事件监听器。

2) 消息传递器: 当某一事件发生时, 消息传递器发送相关消息至该事件的全部监听器。

3) 事件管理器: 设置为单例持久对象以提供全局访问行为。事件管理器连接监听器与传递器, 并接收传递器发

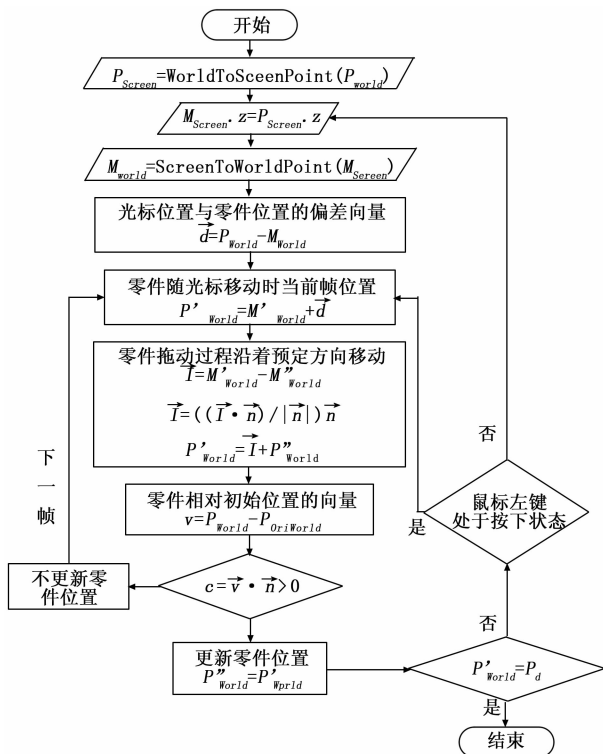


图 3 算法流程图

送的消息然后以事件的形式发送给监听器。

例如某拆卸过程中, 零件 A 和 B 拆卸后零件 C 才可拆, 实现这一关系的事件驱动原理如图 4 所示。

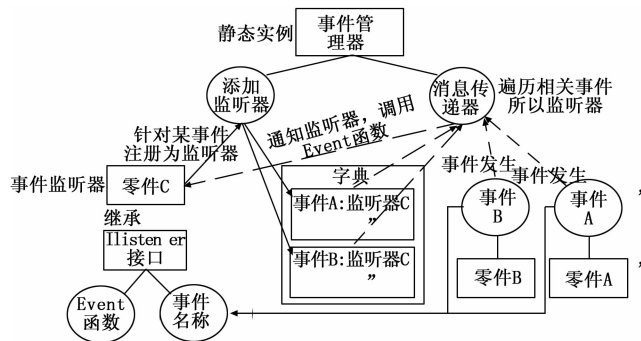


图 4 事件驱动原理

首先定义了 IListener 接口, 通过继承接口, 其它对象根据多态机制可调用接口函数, 因此任何类型对象均可设置为事件监听器。IListener 接口伪代码如下:

```
public enum EVENT_NAME { 事件 A, 事件 B, 事件 C, ..... }
public interface IListener
{
    Event (EVENT_NAME Event_Name, Component Sender, Object Param = null) {}
}
```

使零件 C 继承 IListener 接口, 则零件 C 可设置为事件监听器。

事件管理器主要包括添加监听器和消息传递器两个函

数。其中消息传递器的作用是将发生的事件通知相应监听器。事件管理器伪代码如下：

```
事件管理器
{
    定义 Dictionary< EVENT_NAME, List< IListener> > 类型的
    监听器数组 Listeners, 第一个元素为事件名称, 第二个元素为
    List 类型的监听器列表;
```

```
    添加监听器 (EVENT_NAME Event_Name, IListener Listener)
    {
        if (事件 Event_Name 已存在监听器列表)
            将 Listener 添加进列表;
        else
        {
            新建监听器列表, 将 Listener 添加进列表;
            将 Event_Name 与监听器列表添加进监听器数组 Listeners;
        }
    }
```

```
    消息传递器 (EVENT_NAME Event_Name, Component Sender, Object Param=null)
    {
        if (Event_Name 事件没有对应监听器)
            返回;
```

```
        遍历事件 Event_Name 的所有监听器, 调用监听器的 Event
        函数;
    }
```

```
    移除事件 (EVENT_Name Event_Name)
    {
        移除 Listeners 中的 Event_Type 项;
    }
```

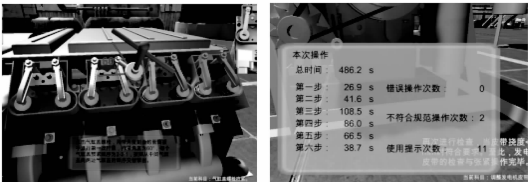
零件 C 通过事件管理器的添加监听器方法，针对零件 A 完成拆卸和零件 B 完成拆卸这两个事件注册为监听器。当零件 A 和零件 B 完成拆卸时，通过消息传递器通知零件 C，C 利用 Event 函数使其自身处于可拆卸状态。

相比利用 Update 函数系统每一帧对零件状态进行判断，基于事件驱动的方法只需在事件发生时将消息通知监听器，很大程度上节省了计算资源。

5 系统实现

根据前述相关技术，基于 Unity3D 软件设计并开发了输油泵机组虚拟维修训练系统。维修拆装训练过程如图 5 (a) 所示。维修拆装训练或技能考核科目结束后存储受训者各操作步骤的完成时间、错误操作次数、使用提示次数等数据如图 5 (b) 所示。

利用 Profiler 工具对系统性能进行测试。当系统运行时，Profiler 工具获取中央处理器、图形处理器的负载、帧



(a) 拆装训练 (b) 记录操作数据

图 5 系统实现

率等性能数据并提供图形化显示。系统硬件平台同 2.1 节，进行拆装训练时的性能数据如图 6 所示。图中横坐标表示帧，纵坐标显示了各帧的帧率、时间和计算开销，其中较大值反映了苛刻的运行环境，此时帧率较低，计算开销较大。受训者开始拆装训练时还未将视角移动至设备处，这时设备模型未在屏幕显示，系统帧率保持在 60FPS 以上，如图前半段所示。当受训者将视角移动至设备处并开始维修操作时，系统计算开销增大、帧率降低，如图后半段所示，但仍保持在 30FPS 以上，满足系统性能需求。系统在整个训练过程中运行流畅，维修模拟效果较好，验证了本文方法的有效性。

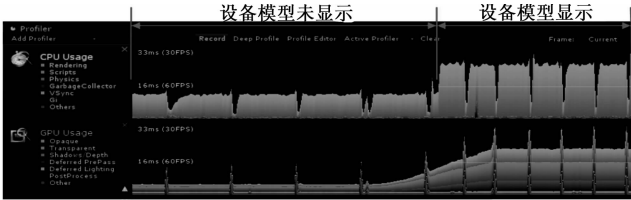


图 6 系统运行性能

6 结束语

1) 本文总结了现有主流虚拟维修仿真平台的特点及适用领域，分析了以 Unity3D 作为系统开发平台的优势。

2) 依据系统面向的对象和目的，分析了系统功能和性能需求。设计了系统多层次模块化体系结构，确定了主要包含的训练科目。

3) 提出了零件位置更新算法，实现了维修训练时鼠标拖动零件的交互式拆装功能。针对维修过程中零件拆装的先后顺序问题，研究了基于事件驱动编程的改进方法，节省了计算资源。

4) 经实际运行测试，系统满足各项功能和性能需求，维修模拟效果较好，验证了本文方法的有效性。系统不仅满足输油泵机组维修训练的需求，还可为其它产品的维修拆装训练系统设计提供参考。

参考文献：

[1] 王文举，李光耀. 虚拟维修仿真技术的研究与进展 [J]. 系统仿真学报, 2011, 23 (9): 1751 - 1757.
[2] Abate A F, Guida M, Leoncini P, et al. A haptic-based approach to virtual training for aerospace industry [J]. Journal of Visual Languages & Computing, 2009, 20 (5): 318 - 325.