

# 飞行体实时飞行显示系统的实现

The Realization of Real-time Monitoring System for Flying Object

权大俊 鲍 珊

(92941部队 葫芦岛125001)

**Abstract** From information visualization angle, this paper puts forward a solution project to real-time monitoring system for flying object, discusses methods, ways and technique in whole. During systematic realization, the key questions with technicality is discussed in detail too. Furthermore, systematic realization provides reference to a certain extent for physics information visualization and simulation fields.

**Keywords** Flying object, 3D graphics, Information visualization

## 1 前言

飞行体实时飞行显示系统,是根据所获得的飞行体的运动参数(包括空间位置、速度及姿态参数),以图形方式实时地显示飞行运动过程的系统。该系统不受天气、气候及超视距的影响,提供给科研及指挥人员生动直观的信息,帮助其及时了解和掌握发生的现场情况,并进行实时监控和决策。它对卫星控制、飞机试飞及导弹试验等领域具有广阔的应用前景和应用推广价值。该系统按照预先建立的三维场景和飞行体模型及天空背景,利用飞行体运动参数控制飞行体在场景中的行为,最终生成用户所观察到的场景图像。如配合音响系统则可大大增强视听觉的冲击力。

在显示系统中飞行体的相对运动是通过飞行体周围的场景背景的变化表现出来,这需要预先建立能够准确反映背景信息的三维场景模型。在系统运行过程中,要求对飞行体模型及复杂的三维场景模型进行实时渲染。这对系统硬件的图形处理能力提出了很高的要求,同时需要高效的渲染算法、对所获得数据的准确处理方法及合理的天空背景表现技巧。从而不仅能够保证系统的实时同步,也能够很好地解决在动画平滑连续性和动画高品质图像之间的矛盾。

## 2 系统结构

显示系统的实现可分为四个部分:场景建模、天空背景处理、数据处理及场景驱动,其基本框架如图1所示。

场景建模是将所要显示的场景对象通过数学方法表达成存储在计算机内的三维图形对象的集合。场景建模一般可采用 MultiGen Creator 作为建模工具,通过人机交互的方式完成场景模型的建立。也可自行开发场景模型组织工具,将使用 3Dmax 等其

它工具建立的个体模型组织成所需的场景模型。

天空背景是根据飞行体在空中运动的速度来控制改变天空背景相对应的纹理,表现飞行体飞行时其周围背景天空随着飞行体运动所产生的变化效果,模拟产生出背景天空中的太阳、云彩等。

数据处理包括滤波处理及坐标变换等。其中滤波处理的目的是消除数据中的随机干扰噪声和减小数据跳变的影响,可以通过传统的处理算法得到完整连续的运动参数。坐标变换完成从飞行体在某种坐标系上的运动坐标到显示坐标系上坐标的转换。

场景驱动是三维视景驱动程序根据所建立的场景模型、飞行体模型及飞行体运动参数来生成实时视景。视景驱动的结果由图像方式输出。用户可观察到飞行体运动状态,场景中的飞行体和天空背景是由经数据处理后的飞行体运动参数来控制。

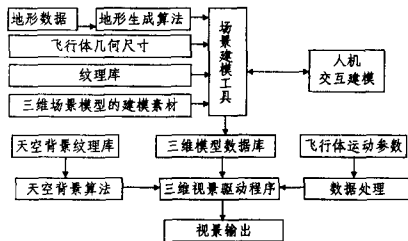


图1 飞行体实时飞行显示系统结构示意图

## 3 系统实现中的主要技术

### 3.1 基于纹理的天空背景

在场景建模中,天空背景以三维方式绘制起来比较困难,同时不富有动态变化,渲染效率也随之相应地降低。纹理映射是将来自现实世界的图像绘制到物体的表面,使物体的外观看起来与现实世界中

的物体相似或相同。利用纹理映射技术正好可以免除上述的缺点,既大大增强图像的真实感又得到很好的表现力。通常用于纹理映射的图像是建立在纹理坐标系中的二维图像。纹理坐标系中的两个坐标轴  $S$  和  $T$  定义了一个二维纹理空间,因此一个纹理映射就是定义在沿纹理坐标系两个坐标轴上的函数。如果此函数是与飞行体运动相关的函数,那么,纹理映射表现出来的是随飞行体的运动变化而产生的天空背景的变化。

实现基于纹理的天空背景的基本思想如下:首先构造一个与应用程序窗口客户区大小一样的平面,不应对它进行模式变换;接着指定一个代表天空背景的纹理图像,为避免循环单调效果的产生,选择纹理图像的大小尽可能大于视窗大小;然后根据飞行体运动速度来调节纹理坐标。

调节天空背景纹理坐标的过程是:将飞行体速度向垂直于视轴方向的平面内进行投影得到水平和垂直两个方向上的速度,经分别对两个速度进行归一化处理,再乘以适当的比例系数,得到两个与飞行速度密切相关的参数,其中由水平方向上的数据来控制  $S$  纹理坐标,而用垂直方向上的数据控制  $T$  纹理坐标。在初始化时提供一个交互界面,可选择代表不同天气气候的纹理图像,效果会更加理想。在场景驱动过程中要注意渲染顺序,天空背景纹理的渲染应得到优先处理,否则会出现天空背景遮挡其它场景等不可预料的现象。

### 3.2 场景建模

系统场景包括飞行体、地表景物、天空背景及地形。通过二维纹理实现天空背景上面已有论述,系统中所需的飞行体及地面景物根据几何尺寸按照一定的比例可以用三维编辑器来建立其模型,这种几何建模适用于结构简单物体的构造,而对于象地面这种具有复杂细节表面的物体来说,这样的几何建模是一项费时费力的工作。同时建立起的地形模型能够准确反映真实的地形地貌,要求场景中的地形与真实地形一致,对地形模型的精度要求也比较高,所以完全由几何建模实现是不太现实。因此,采用突出特征,不失精度的折衷方案,即用 DEM 等地理数据来生成地形的粗略三维模型,通过交互编辑细化特征目标或区域。为了提高建模的效率也可考虑适当采用纹理。这种方案的具体步骤如下:

(1) 确定所要显示的区域,将其分成便于处理的若干个部分,如图2所示。

(2) 分别选择相应区域的 DEM (GIS, DTM) 等地理数据,根据不同情况选择一种方法(如窗口拟合法、移动拟合法及线性内插法等)对地理数据进行内插处理,直至满足精度要求为止。生成三维地形模型并将其分别存储起来。

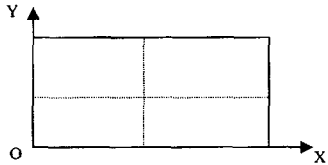


图2 显示区域表示图

(3) 依据可供参考的地形有关资料(如地形的真实图像),利用三维编辑工具对三维地形模型进行编辑修改,以便表现出地形的细节特征,如指定纹理、将地表景物模型与地形模型的合并等。

(4) 将通过(1)、(2)、(3)完成的地形模型组织成一个完整的模型。

### 3.3 坐标变换

要将现实世界中具有物理意义的物体正确显示在监视器中供用户观察则需要一系列变换。在这诸多变换中,自世界坐标系到屏幕二维坐标系的变换由专用图形处理引擎来完成,系统所关心的是真实坐标系到世界坐标系的变换,其它一些变换在有关文献中有相应的介绍。

为了计算方便和观察直观,选用如图2所示的世界坐标系。 $X$  轴代表东西方向, $Y$  轴代表南北方向,该坐标系为右手直角坐标系。设飞行体在某坐标系中的坐标为  $(t, \alpha, \beta, \gamma)$ , 其中  $t$  为时间,则需要把  $(t, \alpha, \beta, \gamma)$  变换至 OXYZ 坐标系中的  $(t, x, y, z)$ 。假如原先坐标系为非直角坐标系,还得将  $(t, \alpha, \beta, \gamma)$  变换至中间直角坐标系中来,然后通过平移和旋转完成中间直角坐标系到世界坐标系的变换。在真正显示驱动之前要对  $(t, x, y, z)$  做比例变换。设系统要求精度为  $d$  (点/m), 在比例变换中  $d$  就是比例系数。同时,建模时也要严格按照此比例系数来保持整个场景中的对象之间的几何关系。无论是怎样做缩放处理,系统实现中模型和飞行体运动数据要保持统一的标准。另外,当所显示处理的地形大小范围过大时注意考虑地球曲率的影响。

## 4 显示系统的软件实现

当完成了整个场景建模和主要技术问题得到解决后,可用视景驱动程序对场景和飞行体进行显示,即根据三维模型文件中存储的图形数据和飞行体实时运动参数,实时地生成在监视器或投影设备上显示的二维图像。本系统在 Windows NT 环境下,以 Delphi 和 OpenGL 作为开发工具,实现了飞行体实时飞行显示系统。该系统中视景驱动程序的流程如下:

(下转第109页)

选项均值:

$$A = [(\sum_{j=m}^{n+m} T_j) / n + (\sum_{j=n}^{n+m} T_j) / v] / 2 \quad (3)$$

采样数据标准差:

$$\delta = \sqrt{\sum_{j=m}^{n+m} (T_j - \bar{T})^2 / n} \quad (4)$$

正向允许范围:

$$\xi_1 = \bar{T} + k\delta \quad (5)$$

负向允许范围:

$$\xi_2 = \bar{T} - k\delta \quad (6)$$

数据升降斜率:

$$\eta = (T_i - T_j) / |i - j| \quad (7)$$

其中  $m, n, i, j, k, u, v$  为正整数, 并且  $m < n, u < v, i < j, T_i, T_j$  为采样数据。

另外还有全局最大值、全局最小值、局部最大值、局部最小值、超界数据个数、报警级别等统计数据, 还有一些焦炉烘炉专业领域的专用统计数据等。

图2是焦炉某纵向切面九个测点温度分布示意图。

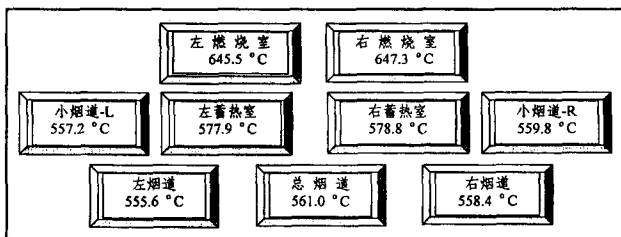


图2 焦炉某纵向切面测点温度分布图

**结束语** 本文所介绍的基于多传感器的现场总线技术的设计思想与方法已在某大型钢铁企业的焦炉烘炉监控系统中得到了成功的应用, 先后对三个大型现代化炼焦炉进行了成功的烘炉, 在每个烘炉过程80天左右的时间里, 都能连续稳定可靠地运行。本文所介绍的设计思想对远距离多测点多传感器监控系统具有积极的借鉴意义。

参加本项研究的还有江杰, 王宗彩, 孙殿迟, 金锦良, 虞明, 张向东, 孙旭峰, 王玮, 姚剑, 马秉瑞, 刘

瑞凤等同志。

## 参考文献

- 1 Xianhui Yang. The Field Bus Technology and Its Application. The Press of Tsinghua University, 1999
- 2 The Proceeding on Seminar of DCS Application Technology and Field Bus, 1997. 11
- 3 Forouzan B. Introduction to Data communications & Networks. McGraw Hill, 2nd Edition, 1997. 21~57
- 4 贺才兴. 大学数学手册. 上海交通大学出版社, 2000. 460~471
- 5 刘其真, 等. 一个大型实时数据自动检测系统图形界面的设计. 计算机工程及应用, 1999, 35(7): 51~53

(上接第98页)

(1)系统初始化. 设置图形运行环境, 提供能够选择飞行体型号、显示地域、天气气候的交互界面, 初始化各类参数。

(2)三维模型载入. 读取三维场景和飞行体模型文件, 根据其图形数据建立各类显示列表。

(3)天空背景生成. 构造二维平面, 读取相应的纹理图像文件。

(4)帧循环. 获得经滤波处理的飞行体运动数据参数, 并根据这些参数计算出天空背景纹理移动的速度, 同时对数据进行坐标变换, 由所得到的数据控制场景中的飞行体模型。接着对整个场景进行渲染。这样就完成了本次循环。考虑到系统实时性要求, 帧时间信号应由硬件中断来产生, 通过对时间硬件中断的响应来驱动帧循环, 以保证系统显示与整个系统的同步。由于在三维图形显示中, 速度与场景的复杂程度、系统硬件性能密切相关, 因此可以采用为同

一个纹理指定多幅图像的多重纹理技术, 适当缓解实时性和动画连续性的矛盾。

**结论** 飞行体实时飞行显示系统可以模拟显示飞行体飞行全过程。实时显示天空、海面、远山、道路、树林包括实时飞行的飞行体在内的视觉环境。目前该系统已用于导弹试验实时测控系统中, 并取得了良好的效果。系统在实时运行的时候还可以对所需要的数据存储, 利用存储数据又能够事后回放。系统在物理信息可视化方面做了些尝试, 对其它各类仿真也有一定的适用性, 并可用于城市规划的三维可视化、飞行体试验控制及现场决策等领域。

## 参考文献

- 1 苏虎, 周美玉. 面向对象的实时三维视景仿真系统的实现. 计算机工程与应用, 2002, 38(6)
- 2 吕建峰, 刘定生, 焦伟利, 李国庆. DEM生成算法研究. 中国图形学报, 2002. 5
- 3 彭晓明, 王坚. OpenGL 深入编程与实例揭秘. 北京: 人民邮电出版社, 1999