

基于预测和双缓冲区的直升机视景仿真

周秀芝, 孙忠云, 曹建平

(海军航空工程学院 青岛校区科研处军用虚拟仿真研究与训练中心, 山东 青岛 266041)

摘要: 为了解决直升机视景仿真中的飞控与视景不同步和降落问题, 首次提出了一种基于预测和双缓冲的直升机视景系统数据同步的解决方法; 该方法首先将直升机的姿态信息进行采样处理, 使之与视景仿真频率匹配, 然后将处理后的姿态信息分别存储于两个缓冲区中, 一个缓冲区的数据驱动场景进行渲染, 另一个缓冲区的数据被用来预测下一帧飞机的位置姿态, 将预测到的数据发送给视景进行碰撞检测, 最终引导直升机正确着陆; 实验结果显示, 该方法产生的数据比不做处理的数据更接近真实数据, 能够满足视景仿真数据的实时性需求, 目前该方法已成功应用在直升机通用着舰训练系统中。

关键词: 视景仿真; 预测; 双缓冲区; 降落

Helicopter Visual Simulation Based on Prediction and Double Buffer

Zhou Xiuzhi, Sun Zhongyun, Cao Jianping

(Qingdao Branch of Naval Aeronautical Engineering Institute, Qingdao 266041, China)

Abstract: A solution of helicopter visual system based on prediction and double buffer is proposed in this paper, which can solve two problems, one is the problem of asynchronism between flight control and visual simulation, the other is landing. The program sampled attitude information of helicopter in order to match frequency of the visual simulation firstly, and then stored it in two buffers. Data in one buffer driven scene rendering, another was used to predict the plane's location of the next frame which is sent to the visual simulation for collision detection and eventually lead helicopter landing. The experimental results show that the processed data is closer to the real data than the original data and can meet the demand of real-time visual simulation, and that the method has been successfully used in general helicopter carrier landing system.

Keywords: visual simulation; prediction; double buffer; landing

0 引言

现代直升机以其独特的环境适应性在军事和民用领域中得到了广泛应用, 我国目前正在大力发展直升机, 直升机的数量在迅速增加, 需要培养大量的直升机飞行员。采用模拟训练的手段可以提高效率、节省经费、保证安全。视景仿真系统在飞行模拟训练中为飞行员提供逼真的视觉感官体验, 提高模拟训练的逼真度, 是飞行训练模拟器的重要组成部分。与固定翼飞机不同, 直升机的起飞降落可以在机场、平地、楼顶平台、舰面等众多地点实现, 不局限于机场跑道, 因此对视景中的细节要求更高, 同时要求视景要与飞控紧密配合, 将直升机与场景中地形和目标的交互信息回馈给飞控, 以配合直升机在不同的地点降落。目前国内的大部分直升机视景仿真采用的是 MultiGen—Paradigm (后与 Engenuity 和 TerreX 公司合并成 Presagis 公司) 的 VegaPrime (以下简称 VP) 作为视景驱动软

件^[1-6], 还有一部分采用 FlightGear^[7]。在直升机视景仿真的报道中, 大部分的文章着重讲述了场景的搭建和控制, 关于飞控数据的处理并不多^[8-10], 而在工程实践中必须解决飞控仿真与视景渲染不同步的问题。

为了解决直升机视景仿真中飞控与视景不同步和降落问题, 本文首次提出了一种基于双缓冲区和预测机制的直升机着舰视景系统的解决方法。该方法首先将直升机的姿态信息进行采样处理, 使之与视景仿真频率匹配, 以达到飞控仿真和视景仿真的时间同步性, 然后将处理后的姿态信息存储于两个缓冲区中, 一个缓冲区的数据传送给视景进行更新; 一个缓冲区中的的姿态信息来预测飞控仿真的下一帧数据, 然后传送给视景进行碰撞检测, 最终使直升机正确降落。接下来首先对飞控与视景不同步的原因进行分析, 并给出相应的解决方法, 最后给出实验结果。

1 飞控与视景不同步的原因分析

直升机视景仿真中, 飞控系统计算出飞机的位置, 通过网络传输到视景系统, 驱动视景模型库建立战场环境。各系统之间的关系如图 1 所示, 视景控制程序收集飞控和其他模块与视景系统进行交互的各种信息, 主要包括直升机和其他目标的姿态位置信息, 并把这些信息转换成相应的协议发送到视景渲染模块, 将视景渲染模块反馈回来的碰撞信息等进行协议转换后发送给飞控解算; 视景控制主

收稿日期: 2017-11-06; 修回日期: 2018-01-08。

基金项目: 国家自然科学基金重大研究计划资助 (91538201)。

作者简介: 周秀芝(1977-), 女, 博士, 研究生, 副研究员, 主要从事虚拟现实, 视景仿真, 图像处理方向的研究。

孙忠云(1974-), 男, 硕士, 副教授, 主要从事军事仿真, 虚拟现实方向的研究。

程序主要负责对场景中的模型和特效等进行统一的控制和管理；视景渲染模块主要负责接收视景驱动发送过来的信息，并在场景中作出相应的渲染和改变，同时将需要回应的信息发送给视景控制模块。

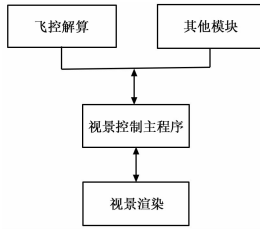


图 1 直升机着舰中各系统关系图

飞控解算出的飞机六自由度信息一般会通过网络数据传输给视景模块，由于网络堵塞或者延迟等原因，会出现漏帧的情况。假设飞控的解算频率和视景的刷新频率完全一致，那么在 T_n, T_{n+1}, T_{n+2} 时刻，视景端应该收到 F_n, F_{n+1}, F_{n+2} 帧的数据，但由于网络原因，只收到了 F_n, F_{n+2} 帧的数据，这种情况下会导致场景出现卡顿，尤其是飞机做大坡度飞行时，使用者会感觉到场景的抖动。

飞行仿真比较复杂，一般是根据自己的时钟和仿真步长进行多步长推进的方式进行计算。一般飞行仿真的内部解算采用 10 ms 的周期进行。视景系统是通过数据来驱动视景模型的，根据视景每帧获取的实体状态进行等步长解算。视景更新频率为 60 Hz，也就是以 16.7 ms 为步长进行解算。这样就存在刷新频率和数据帧不同步问题，而且帧周期也不一样，加上每帧计算所花费时间不一样，再加上网络数据传输延时，由此引起数据读错帧。在视景解算时就会发生连续两帧数据读取是同样数据，或者有些帧数据未被读取等现象（跳帧）^[8]。如图 2 上图所示，在飞控和视景分别以 100 帧/秒和 10 帧/秒运行的情况下，在理想情况下，当视景渲染 1, 2, 3, 4, 5, 6, 7, 8 帧时，所用到的飞控数据分别为 1, 2, 4, 5, 7, 9, 10, 12 帧。这样飞控与视景渲染达不到同步，在场景渲染时会出现抖动等情况，影响用户的使用；同时飞控系统需要视景反馈回的碰撞信息进行解算，视景在 1, 2, 3, 4, 5, 6, 7, 8 帧所检测的碰撞信息分别反馈到了飞控的 1, 2, 5, 7, 9, 10, 12, 13 帧，这种非实时反馈的碰撞信息会严重影响飞控的计算结果，导致飞控出现不可预料的错误。

2 飞控与视景不同步的解决方法

通过第一节分析可以知道，飞控与视景不同步主要是由两个原因引起的：一是网络传输所导致的漏帧和跳帧；二是飞控的计算频率与视景的刷新频率不一致所导致的错帧。针对这两种问题，采用不同的方法予以解决：一是在视景控制端利用已知数据对没有收到的飞控数据进行模拟推演，保证数据的完整性；二是利用双缓冲区来解决错帧问题。

2.1 利用推演解决漏帧问题

飞控与视景之间的数据通过网络进行传输，网络传输会

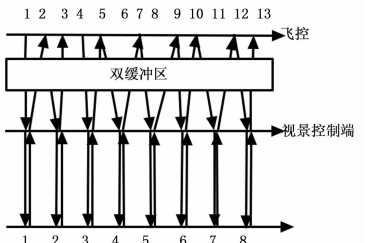
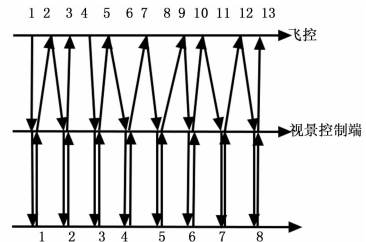


图 2 飞控与视景数据传输示例

导致数据丢失，而视景刷新时需要连贯一致的数据，否则会出现场景抖动的现象。为了得到与视景刷新同步的数据，同时又不能增加视景端的计算负担，因此需要在视景端需要利用已经收到的数据推算出丢失的数据，从而保证数据的一致性。假设视景在时刻收到的飞机六自由度数据为 $\{Lon_n, Lat_n, Alt_n, H_n, P_n, R_n\}$ ，其中 Lon_n, Lat_n, Alt_n 代表了飞机的位置信息，分别是飞机的经度、纬度和高度信息， H_n, P_n, R_n 表示飞机的姿态信息，分别是航向角、俯仰角和横滚角信息。视景端同时还会收到飞控传过来的飞机速度数据 $V_n = \{V_x, V_y, V_z, V_h, V_p, V_r\}$ ， V_x, V_y, V_z 分别是飞机速度的北向分量，东向分量和天向分量； V_h, V_p, V_r 分别是飞机的偏航角速度，俯仰角速度和横滚角速度。而在 t_{n+1} 时刻没有收到数据。此时视景端需要根据已经收到的飞机位置姿态信息和速度信息推算出飞机在 t_{n+1} 时刻的位置姿态信息。因为飞机的位置是用经度、纬度和高度来描述的，而飞机的速度信息却是以飞机起飞点为原点，北向为 X 方向，东向为 Y 方向，天空为 Z 方向的笛卡尔坐标系来描述的，因此需要进行坐标变换才能使二者的坐标一致。利用函数 $f(x)$ 表示从经纬度坐标到北东天坐标的变换，则在北东天坐标系下的飞机位置如公式 (1) 所示。

$$D_n = \{X, Y, Z, H, P, R\} = f(D_n) = \{f(Lon_n, Lat_n, Alt_n), H_n, P_n, R_n\} \quad (1)$$

用函数 $f'(x)$ 表示从北东天坐标系到经纬度坐标系的转换，则视景端在 t_{n+1} 时刻的飞机六自由度数据如公式 (2) 所示。

$$D_{n+1} = f'(f(D_n) + A_n) \quad (2)$$

2.2 利用双缓冲区解决错帧问题

飞控与视景是一个双向交互的过程，如果视景只是接收飞控数据进行飞机姿态的定位和显示，那么问题变的相对简单，可只对视景接收到的飞机姿态信息进行数据处理后送渲染端进行显示即可，在很多工程应用上都如此处理；而在飞控需要视景实时反馈的碰撞检测结果作为飞控的输

入量进行解算时, 整个过程成为一个需要实时反馈的系统, 这样视景系统不但需要处理接收到的数据, 还需要保证将当前飞机与场景中发生碰撞的信息实时返回。本文通过在飞控和视景控制端加入了双数据缓冲区来解决这个问题, 如图 2 下图所示, 在飞控的数据传送给视景之前, 先将其数据放入双缓冲区中, 在缓冲区中对数据进行处理, 使得飞控发送给视景的数据和飞控接收到的视景数据都满足要求, 能够同步。具体实现时, 如图 3 所示, 缓冲区 1 存放用来进行视景渲染的飞机信息, 通过插值计算使飞控解算频率与渲染频率匹配; 缓冲区 2 存放用来进行碰撞检测的飞机信息, 通过预测算法使飞控解算频率与渲染频率匹配。视景中放置两架飞机, 一架用来渲染, 一架“影子飞机”只用来进行碰撞检测的计算, 不在场景中进行渲染, 这样既可以保证在渲染时按照渲染端的频率进行更新, 又可以保证碰撞检测的实时性。

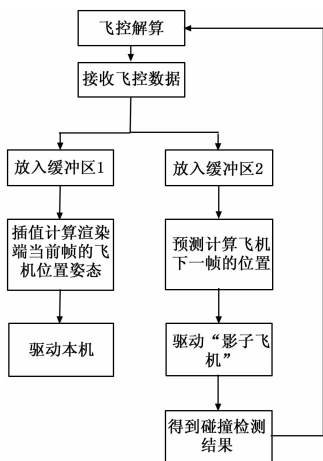


图 3 双缓冲区示意图

为了达到频率匹配, 视景和飞控中的数据都自带帧号, 若当前视景渲染到 n 帧, 飞控解算到 m 帧, 缓冲区 1 由一个 10 帧的队列组成, 所收到的数据分别为 $\{D_m, D_{m+1}, D_{m+2}, D_{m+3}, D_{m+4}, D_{m+5}, D_{m+6}, D_{m+7}, D_{m+8}, D_{m+9}\}$, 用来渲染的数据为 $\{DD_n, DD_{n+1}, DD_{n+2}, DD_{n+3}, DD_{n+4}, DD_{n+5}\}$, 则它们之间的关系为:

$$\begin{cases} DD_n = D_m \\ DD_{n+1} = C_1 D_{m+1} + C_2 D_{m+2} \\ DD_{n+2} = C_3 D_{m+3} + C_4 D_{m+4} \\ DD_{n+3} = D_{m+5} \\ DD_{n+4} = C_6 D_{m+6} + C_7 D_{m+7} \\ DD_{n+5} = C_8 D_{m+8} + C_9 D_{m+9} \end{cases} \quad (3)$$

其中: $C_1+C_2=1, C_3+C_4=1, C_6+C_7=1, C_8+C_9=1$ 。从公示 (3) 可以看出, 利用线性插值算法可从缓冲区 1 中计算出需要渲染的数据。这样可保证飞控发送过来的数据能够与视景的渲染速度相匹配, 如果视景渲染帧速或飞控的解算频率发生改变, 则相应修改公式 (3) 即可。为了保证数据的一致性, 缓冲区 1 中的数据采用滑动窗口进行, 按照“先进先出”的原则, 采用队列存储缓冲区 1 的数据。利用缓冲

区计算的时候, 使用者操作后的效果在视景上显现时会延迟 0.01 秒, 这种延迟不会影响使用者的主观感受。

缓冲区 2 用来计算当前帧的碰撞检测, 本文中采用的 LOS 碰撞检测, 该检测需要提供检测线的起点和终点形成检测线, 而检测线的起点和终点依附在飞机尾部上, 因此是随着飞机实时更新的。假设视景在第 n 帧接收到碰撞检测请求, 会在第 $n+1$ 帧将检测结果返回。如果将当前帧的碰撞检测请求直接发送给视景, 则结果需要在下一帧才能收到, 延迟了一帧。为了解决这个问题, 我们设立了第二个缓冲区。假设收到飞控的当前帧数据为第 i 帧, 则在第二个缓冲区中存放的五帧数据分别为 $\{D_{i-4}, D_{i-3}, D_{i-2}, D_{i-1}, D_i\}$ 。首先根据公式 (3) 计算出这五帧数据对应的三帧渲染数据 $\{DD_{n-2}, DD_{n-1}, DD_n\}$; 然后利用飞机当前帧的位置和加速度预测出下一帧飞机的位置姿态。公式如下:

$$DD_{n+1} = DD_n + C_0 V_n + C_1 V_{n-1} \quad (4)$$

其中: $C_0+C_1=1, V_n=DD_n-DD_{n-1}, V_{n-1}=DD_{n-1}-DD_{n-2}$ 。将预测到的数据发送给视景中的“影子飞机”, 然后将视景发送回来的碰撞检测结果作为输入量传送给飞控进行解算。

3 实验结果

如果直接将飞控数据赋值给视景, 我们记录了视景渲染帧数和飞控计算帧数, 他们的关系如图 4 所示, 其中横轴是视景渲染帧数, 纵轴是飞控计算帧数。若两者的仿真频率完全匹配, 则该图中应该是一条直线, 而现在却是折线, 因此帧率是不匹配的, 而且由于帧率不匹配, 出现了错帧和漏帧, 图中圆圈所示是一处非常典型的错漏帧情况。

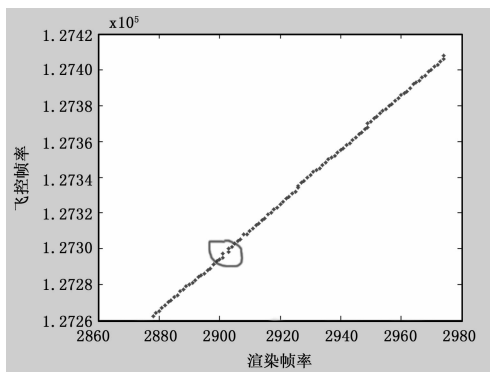


图 4 视景渲染帧数与飞控仿真帧数关系图

以飞行高度数据为例, 设置飞控仿真频率为 100 帧/秒, 视景渲染频率为 50 帧/秒, 得到根据视景渲染频率采样插值后的数据和没有采样直接接收的两组数据, 图 5 显示了这两组数据与直接按照 50 帧/秒的频率得到的飞控数据图, 其中横轴是仿真时间, 纵轴是分别是三组数据, 线段 1 是按照 50 帧/秒的频率得到的飞控数据 (即真实数据); 线段 2 是根据视景渲染频率采样插值后的数据; 线段 3 是没有采样直接接收的数据。图 5 上图是原始图, 下图是局部细节放大后的图像。从中可以看出, 线 2 比线 3 更接近真实数据, 即根据视景渲染频率采样插值后的数据比没有采样

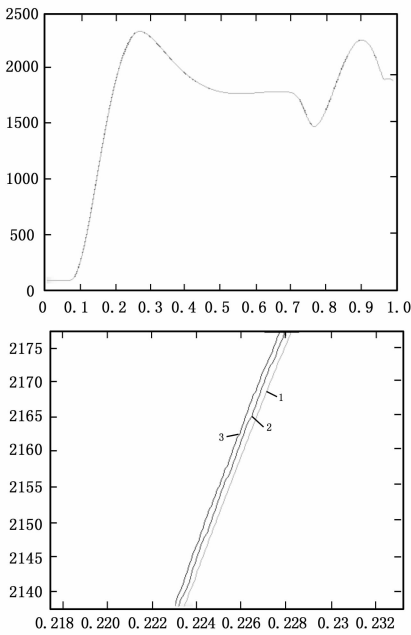


图 5 采样后数据、非采样数据和真实数据之间的对比

直接接收的数据更接近原始数据。以 50 个数据为一组，取 100 组，将这两组数据与真实数据之差的方差进行对比，如图 6 上图所示，其中曲线 1 表示根据视景渲染频率采样插值

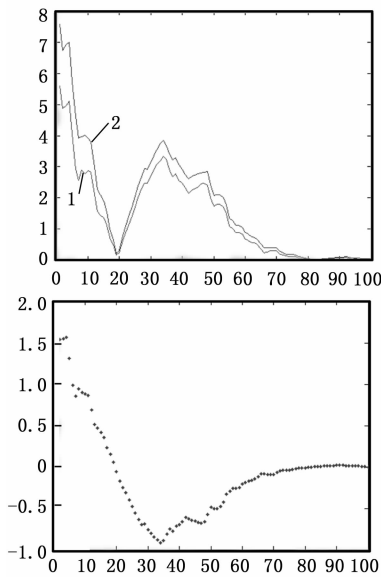


图 6 方差分析图

(上接第 160 页)

[10] 黄华晋, 秦建息. 基于三角定位算法的 WiFi 室内定位系统设计 [J]. 广西科学院学报, 2016, 32 (1): 59-61.

[11] 郗万新. 基于安卓平台的一种 WiFi 精确定位系统的设计与实现 [D]. 南京: 南京邮电大学, 2014.

[12] 侯启真, 史秉鑫, 刘衍帆. 基于 RSSI 的 ZigBee 定位技术研究 [J]. 计算机应用与软件, 2016, 33 (4): 134-137.

[13] 杜月林, 石欣然, 王克寒. 基于 ZigBee 技术的室内定位

后的数据与真实数据之差的方差, 曲线 2 是直接接收的数据与真实数据之差的方差; 可以看出, 曲线 1 数据与真实数据之差的方差更小, 因此本文的采样算法是成功的。用同样的方法, 以 50 个数据为一组, 取 100 组, 计算预测到的飞控数据与真实数据之差的方差, 其结果如图 6 下图所示, 本文的预测数据越来越收敛与真实数据, 因此预测算法也是成功的。

4 结束语

该方法目前已应用于多个直升机模拟训练项目中, 有效解决了由于飞控与视景不同步所引起的场景抖动问题, 并且能够保证直升机在多个地点正常降落, 目前该方法已成功应用于通用直升机着舰系统, 取得了非常好的效果。下一步将在飞控数据预测中进一步优化算法, 使预测的数据更接近真实值。

参考文献:

[1] 申海荣, 王新民, 赵凯瑞. 基于 Vega_MFC 的直升机视景仿真设计 [J]. 计算机测量与控制, 2010, 18 (10): 2385-2390.

[2] 程海涛. 基于虚拟现实的直升机驾驶模拟器开发 [D]. 长春: 吉林大学, 2014.

[3] 赵永强. 直升机飞行模拟器视景仿真系统的设计与实现 [D]. 北京: 北京邮电大学, 2010.

[4] 魏靖彪, 郭广利, 黄海, 等. 直升机布雷视景仿真的建模与实现 [J]. 航空计算机技术, 2009, 39 (1): 78-83.

[5] 李石磊, 梁加红, 刘欣添, 等. 直升机飞控系统仿真平台开发 [J]. 计算机仿真, 2010, 27 (3): 64-68.

[6] 潘婷婷. 舰载机进近着舰航线设计及控制系统仿真 [D]. 南京: 南京航空航天大学, 2014.

[7] 潘浩曼. FlightGear 与 FPGA 的直升机飞行仿真系统研究 [D]. 南昌: 南昌航空大学, 2014.

[8] 徐勇波, 王行仁, 贾荣珍. 飞行仿真器联网仿真视景图像抖动研究 [J], 系统仿真学报, 2005, 17 (2), 414-416.

[9] 吕品, 张金芳. 基于 PC 机群的多通道视景仿真技术研究 [J], 计算机工程与应用, 2006 (26), 38-40.

[10] 高焯, 郑康平, 郭俊丽, 等. 飞行模拟系统中分布交互视景的设计与实现 [J], 指挥控制与仿真, 2013, 35 (5), 84-87.

[11] 系统算法研究及实现 [J]. 计算机技术与发展, 2014, 24 (7): 245-249.

[14] 卞合善. 基于蓝牙 4.0 低功耗室内定位研究 [D]. 北京: 北京邮电大学, 2015.

[15] Mario Munoz-Organero, et al. Using bluetooth to implement a pervasive indoor positioning system with minimal requirements at the application level [J]. Mobile Information Systems, 2012, 8 (1): 73-82.