

基于 RTX 和反射内存的红外景象投射系统实时性设计

苏德伦^{1,2}, 张金生¹, 廖守亿¹

(1. 第二炮兵工程大学, 陕西 西安 710025; 2. 解放军 96111 部队, 陕西 韩城 715400)

摘要: 红外景象投射是红外成像半实物仿真的关键技术, 实时性是其重要性能指标。提出了一种红外景象投射系统设计方案, 系统由红外图像生成和红外图像转换两部分组成。图像生成系统基于通用 GPU 实现, 图像转换基于电阻阵列实现。根据仿真应用对像元规模和帧频的需求, 采用反射内存网络实现图像数据传输和帧同步。通过红外图像帧周期测试, 评估系统的实时性。测试结果表明, 在 Windows 系统下红外图像生成帧周期不稳定, 会出现随机性大幅度偏差。提出采用 RTX 的扩展的实时性设计, 给出详细实现方案。试验结果表明: Windows 系统中断响应的不确定性是影响图像帧频稳定的主要原因, 基于 RTX 的实时性设计可以保证图像生成在帧同步控制下稳定地以 200 Hz 帧频运行。

关键词: 红外景象投射; 实时系统; RTX 实时扩展; 反射内存; 帧同步

中图分类号: TP391.9 **文献标志码:** A **文章编号:** 1007-2276(2013)05-1117-06

Real-time design of IRSP system based on RTX and reflective memory

Su Delun^{1,2}, Zhang Jinsheng¹, Liao Shouyi¹

(1. The Second Artillery Engineering University, Xi'an 710025, China; 2. 96111 Army of PLA, Hancheng 715400, China)

Abstract: Infrared scene project (IRSP) is a key technology for infrared imaging hardware in the loop (HWIL) simulation. Real-time quality is very important for IRSP system. A design of IRSP system was presented. The IRSP system was composed of infrared image generation subsystem and infrared scene transformation subsystem. The two subsystems were based on general GPU and resistor array device respectively. Reflective memory network was used for image data transfer and frame synchronization to meet the demand of image size and frame rate for HWIL simulation. The real-time quality was evaluated by testing IR image frame period. The result indicates that significant errors occur unpredictably under Windows system and the frame period is unstable. A new real-time design was introduced based on RTX extension, and the detailed scheme was presented. Test results of the real-time design indicate that the reason for significant frame period error is the unpredictable interrupt response delay under Windows system and the real-time design under RTX extension can ensure a stable frame rate of 200 Hz under the control of frame synchronization.

Key words: IR scene projection; real-time system; RTX real-time extension; reflective memory; frame synchronization

收稿日期: 2012-09-07; 修订日期: 2012-10-11

基金项目: 中国博士后科学基金(2012M512150)

作者简介: 苏德伦(1981-), 男, 在站博士后, 主要从事控制系统仿真与系统分析方面的研究。Email: sdl_saec@163.com

0 引言

随着红外成像技术在军事上的广泛应用,在系统的研制和测试中对红外景象投射系统的需求愈加迫切。红外景象投射系统通常作为红外目标模拟器与仿真计算机、飞行转台和光学系统一起组成半实物仿真系统,用于红外成像精确制导武器系统的设计验证与产品测试,是整个系统的关键部件。红外景象投射系统由两部分组成:红外图像生成系统和红外图像转换系统。如果把红外景象投射系统比作一套 PC 机系统,红外图像生成系统相当于主机,红外图像转换系统相当于显示器。红外图像生成系统生成表征红外辐射亮度的灰度图像,红外图像转换系统将灰度图像变换为中波(3~5 μm)或长波(8~12 μm)红外辐射图像。红外辐射图像经由光学系统被投射到被测系统的入瞳处,能对被测系统的光学系统和成像系统进行完整的测试,是目前红外成像系统最有效的测试与仿真手段。图像尺寸和帧频是红外景象投射系统的两个重要指标,它们反映了图像的精细度和动态特性的优劣。目前,对于高动态对象的仿真应用,要求图像尺寸达到 256×256,帧频达到 200 Hz。

文中分析了 Windows 环境下基于光纤反射内存网络的红外景象投射系统^[1-2]存在的实时性问题。提出基于 RTX 实时扩展的设计方案,满足 200 Hz 帧频下 256×256 红外景象投射对图像数据传输和帧同步的实时性要求。

1 实时性问题

1.1 红外景象投射系统结构

由于 X86 架构微型计算机(PC)计算能力和通用图形卡(GPU)图形加速能力的提升,红外图像实时生成系统早期所采用的专用计算机系统设计方案已由 PC 架构通用 GPU 的设计方案所取代^[3-4]。在红外图像转换技术领域,电阻阵列和红外数字微镜器件(DMD)是应用最为成熟的技术。电阻阵列由于具有较宽的温度范围、较高的工作频率和完全自主的知识产权^[5],因此是高动态条件下红外景象投射的首选器件。红外景象投射系统的组成及结构如图 1 所示。

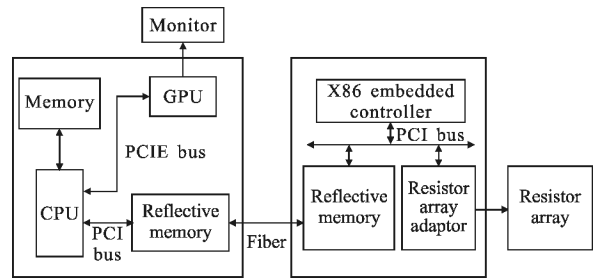


图 1 红外景象投射系统结构
Fig.1 Block diagram of IR scene project system

图像生成系统和图像转换系统通过反射内存光纤网络连接,实现数据传输与帧同步。反射内存数据传输率最高可达 174 MB/s,多模光纤传输长度可达 300 m^[6]。可以满足 200 Hz 帧频下 16 位灰度等级 256×256 红外图像数据的传输数据需求,如公式(1)所示。

$$256 \times 256(\text{pixel}) \times 200(\text{Hz}) \times 2(\text{Byte}) \leq 25.6 \text{ MB/s} \quad (1)$$

光纤数据传输具备较好的抗干扰特性并且宜于图像转换器在飞行转台上安装。

1.2 图像数据传输与帧同步的实时性问题

1.2.1 系统工作过程

图像生成系统工作在 Windows XP 操作系统,采用 Vega SensorVision 红外模块生成红外图像,利用 OpenGL 接口将灰度图像数据从显卡缓存中取出,图像经过处理后经由反射内存网络传输给图像转换系统。帧频控制信号由图像转换系统生成,通过反射内存中断实现与图像生成系统的帧同步。系统的工作流程如图 2 所示。

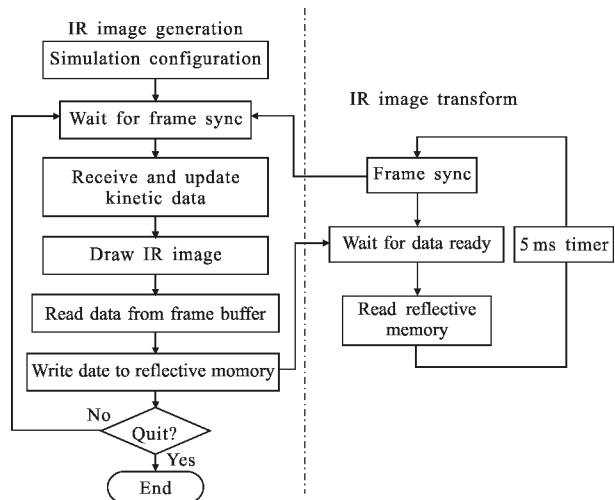


图 2 红外景象投射系统工作流程
Fig.2 Work flow of IR scene project system

红外图像转换系统由嵌入式控制系统组成,运行 VxWorks 实时操作系统。5 ms 定时信号由电阻阵列驱动控制卡上的晶振产生,通过 PCI 总线中断嵌入式控制器发送反射内存中断,产生高精度的帧同步信号。

1.2.2 系统实现

红外图像生成软件运行在 Windows 平台,基于 VC6.0 和 Vega 开发。红外图像转换系统每进入一个 5 ms 定时周期,立刻通过光纤发送反射内存中断给图像生成系统。图像生成系统接收到反射内存中断后,触发 PCI 总线中断,控制图像生成,从而实现两个子系统的同步。为了保证图像生成系统精确地运行在 200 Hz 帧频下,采取如下几项优化措施:

- (1) 关闭垂直同步,避免受到监视器刷新率的影响;
- (2) 为提高 PCI 总线中断的响应速度,剪裁 Windows 系统的后台任务,只保留系统运行必须的基本服务 (Event log, Plug and Play, RPC, DCOM Server Process Launcher);
- (3) 优化仿真对象的几何形体模型,减少顶点数量,缩短图像生成时间;
- (4) 反射内存读写采用 DMA 模式,降低图像数据传输时间。

1.2.3 实时性评估

系统实时性测试利用 CPU 计数器接口实现,这种方法是在 Win32 环境下比较精确的时间测试方式^[1]。以帧周期测试为例,其测试流程如图 3 所示。

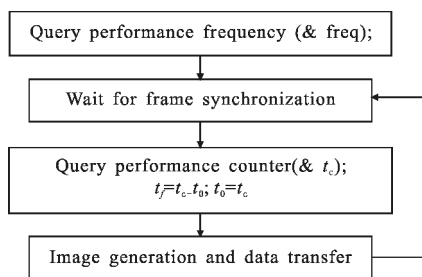


图 3 帧周期测试方法

Fig.3 Frame period test method

选择 HP xw9400 图形工作站作为图像生成系统的硬件平台,配置如表 1 所示。

表 1 图形工作站配置参数

Tab.1 Graph work station configuration

Parts	Parameter
CPU	Dual AMD operon2222 dual-core processor 3.0 GHz
Chipset	NVIDIA nForce professional 3 600 and NVIDIA nForce 2 050
Memory	4GB ECC DDR2-667 MHz SDRAM
GPU	NVIDIA quadro FX5500 1GRAM

仿真对象为由 730 个三角形组成的战斗机模型,背景为晴朗无云的天空。图像生成帧数不低于 2 000 帧,对红外图像绘制时间(t_{img})、图像数据写反射内存时间(t_{fm})和帧同步周期(t_{fm})进行了测试,测试结果如表 2 所示。

表 2 实时性测试结果(毫秒)

Tab.2 Real-time quality test result(ms)

Item	Max	Min	Mean	Std
t_{img}	1.170 6	0.586 7	0.644 3	0.046 0
t_{fm}	0.895 5	0.841 3	0.864 9	0.046 0
t_{fm}	8.723 0	2.993 0	5.000 6	0.099 6

测试结果表明:得益于 HP xw9400 图形工作站全长 PCI 总线 (66 MHz 时钟, 66 位数据宽度) 的优势,图像数据传输的带宽达到了 155 MB/s,接近反射内存的最大数据传输率;仿真目标几何形体优化后,在 GPU 加速下,图像绘制时间最大不超过 1.2 ms;虽然图像生成和数据传输的总耗时完全满足 5 ms 的时间要求,但帧同步周期测试结果仍出现了大幅跳变。从帧同步实现的方法和原理分析,问题的主要原因定位于 Windows 下中断响应延迟的不确定特性。

2 基于 RTX 实时扩展的设计

2.1 RTX 实时扩展

RTX(Real-Time Extension)作为 Windows 操作系统内核体系的延拓,修改并扩展了整个硬件抽象层,实现了独立的内核驱动模式,形成了与 Windows 操作系统并列的实时子系统(RTX Subsystem, RTSS)。RTSS 利用抢占式和时间片轮转算法使得线程上下文切换时间保持在 500 ns~2 μ s 范围内,实现了确定性的线程调度。RTSS 提供了高精度的精确时钟和定

时器, 时钟分辨率可达 100 ns, 定时周期最低可达 0.1 ms。利用共享内存机制, RTSS 可与 Win32 进程进行通信和数据交换。在单处理器环境下, RTSS 线程优先级高于所有 Win32 线程; 在多处理器环境下, RTX 可以单独占用一颗处理器资源, 从而提供更高的实时性^[7]。

2.2 设计方案

根据前述分析, 红外图像生成帧周期不稳定的主要是由反射内存触发 PCI 总线中断时, Windows 系统响应中断延迟的不确定性引起的。因此, 要解决问题须将反射内存硬件的访问与控制纳入到实时子系统环境中, 对红外图像生成系统进行实时性改造。在红外图像生成系统中采用的反射内存为 GE FANUC VMIC-PCI5565, GE 官方未提供 RTX 驱动, 但提供了详细的硬件说明书和寄存器配置信息。基于硬件寄存器配置和原理开发反射内存的 RTX 驱动, 用于系统的实时模块设计与实现。

基于 RTX 实时扩展的红外图像生成系统软件由两部分组成: 图像渲染模块和图像传输与帧同步模块, 如图 4 所示。

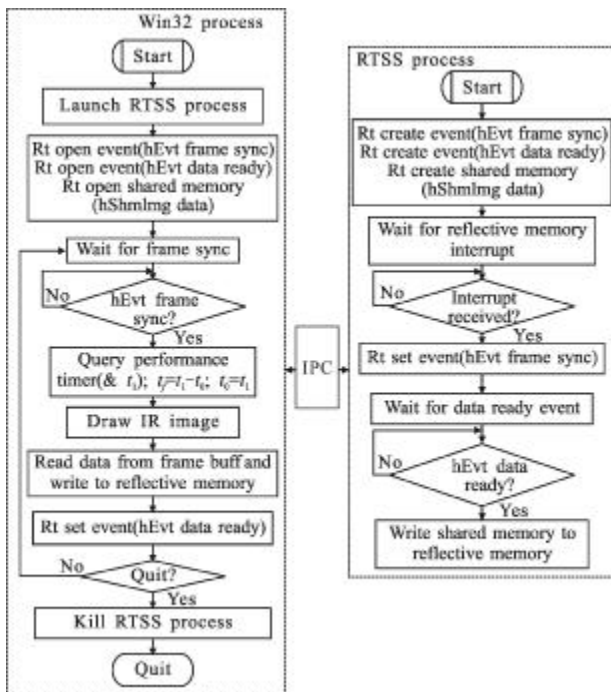


图 4 RTX 实时扩展流程图

Fig.4 Flow chat of RTX real-time extension

图像渲染模块是一个运行在 Win32 环境下的进程, 利用 Windows 平台下丰富的软件资源(如 Vega,

Creator)实现红外图像的渲染。图像传输与帧同步模块是运行在 RTSS 环境下的实时进程, 它通过 RTX 下的驱动接口实时控制反射内存卡。

红外图像转换系统的帧同步信号以反射内存中断的形式发送给红外图像生成系统, 反射内存 RTX 驱动接收到该中断后实时地触发 PCI 总线中断。图像传输与帧同步模块在 PCI 总线中断下触发帧同步事件, 图像渲染模块捕获到帧同步事件后开始绘制红外图像。红外图像数据绘制完毕后, 图像渲染模块从帧缓存取图像数据写入共享内存并触发数据准备好事件。图像传输与帧同步模块捕获到数据准备好事件, 将共享内存数据写入反射内存, 完成从帧同步到图像数据传输一个完整的帧周期。

2.3 反射内存的 RTX 驱动开发

RTX 是 Windows 平台的一个实时扩展子系统, 它的实时性通过建立自己独立的硬件抽象层(Hardware Abstraction Layer, HAL)来实现。RTX 的 HAL 实现对硬件资源管理, 其对中断的管理与 Windows 内核是完全隔离的。非实时内核(Win32 内核)和实时内核(RTSS)共存是 RTX 实时扩展的显著特点。反射内存 RTX 驱动开发应着重关注以下两个问题。

2.3.1 中断资源获取

PCI 总线具有中断共享的特点, 但是 RTX 下的设备不可与 Win32 下的设备共享 PCI 中断资源。因此, 反射内存所使用的中断资源必须没有被 Win32 内核下任何设备使用, 否则 RTX 设备将无法挂接中断。中断资源的使用情况可通过 Windows 下的设备管理器来查看。一般情况下, PCI 设备在计算机加电后就已经由 BIOS 自动分配中断资源, 因此, 所使用的计算机须提供足够的空闲中断资源供 RTX 使用。当分配给反射内存的中断已经被 Win32 下设备使用时, 可以通过更换反射内存的 PCI 插槽重新分配中断资源。若所有的 PCI 插槽都无法提供空闲的中断资源, 则应考虑更换硬件扩展能力更强的计算机。

2.3.2 虚地址模式

设备驱动以 RTSS 动态链接库(RTSS DLL)的形式实现, RTSS DLL 不是一个真正的动态链接库而是一个 RTSS 进程, 它运行在内核空间因此可以向所有 RTSS 应用进程导出驱动接口。因此, 在调用 RTX 驱动接口前, 首先应保证 RTSS DLL 进程已经运行。

RTX 采用了和 Win32 相同的虚地址模式, 每个

RTSS 进程拥有独立的地址空间。因此,RTX 驱动程序应建立命名内核变量,并在驱动运行时就完成内核变量的创建。RTSS 应用进程在调用驱动接口前应初始化设备,通过内核变量名称来获取内核变量句柄,从而避免虚地址模式下跨进程接口调用问题。

2.3.3 RTX 驱动程序实现

RTX 提供了基于 VC6.0 的应用向导 (App Wizard),可自动生成 PCI 设备驱动程序框架,便于在 VC6.0 下开发 PCI 设备驱动。驱动接口的定义和功能实现只需按照反射内存的硬件说明和寄存器配置信息进行编程即可。

2.4 试验评估

为评估上述设计的实时性,在 RTX 实时环境下采用 VC6.0 编写图像传输与帧同步模块,在 Windows 下采用 VC6.0 和 Vega 编写图像渲染模块,组成红外图像生成系统。该系统的硬件平台为装有 VMIC-PCI5565 反射内存的 PC 机,运行 WindowsXP+RTX7.1 操作系统,采用自研的反射内存 RTX 驱动。

利用另外一台装有 VMIC-PCI5565 反射内存的 PC 机模拟红外图像转换系统,运行 WindowsXP+RTX7.1 操作系统。在 RTX 环境下,创建 5ms 精确定时器,在每个定时周期起始发送反射内存中断,模拟帧同步信号。

首先测试事件触发对实时性的影响,测试方法如图 5 所示。

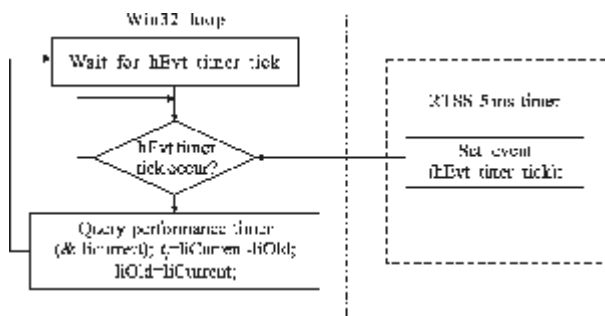


图 5 事件触发实时性测试方法

Fig.5 Real-time quality test of event trigger

在 RTX 环境下创建 5 ms 定时器,周期性触发事件;通过进程间通讯在 Win32 下循环捕获事件,测试循环周期(tf)。测试结果如图 6 所示。

测试结果表明:Win32 进程与 RTSS 进程间事件触发对定时精度带来的影响最大不超过 31 μ s,

Win32 下循环周期稳定。

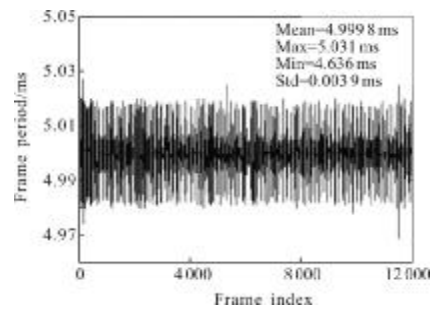


图 6 实时性测试结果

Fig.6 Real-time test result of event trigger

然后测试 RTX 实时扩展设计的性能。在模拟红外图像转换系统上创建 RTX 定时器,以 5 ms 周期向图像生成系统发送反射内存中断,模拟帧同步信号。采用 VC6.0 和 Vega 编写红外图像生成测试软件,如图 7 所示。



图 7 帧周期测试软件

Fig.7 Frame period test software

测试时长 300 s 总计 60 000 帧,测试结果如图 8 所示。其中有 4 处明显的跳变,跳变幅度最大的 A

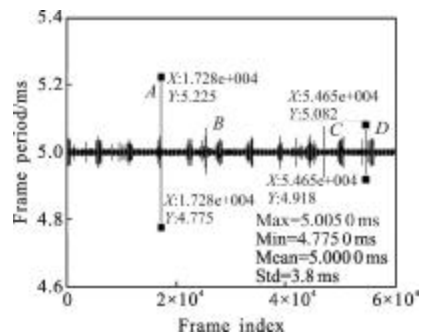


图 8 RTX 下帧周期测试结果

Fig.8 Frame period test result under RTX

点,周期偏差达到 225 μ s。其余 3 处偏差都不高于 100 μ s。多次测试结果的统计数据显示:在 RTX 实时扩展下采用反射内存中断传递帧同步信号,图像生

成帧周期的均值相对于理论值的偏差不超过 100 ns, 帧周期标准差不大于 10 μ s; 帧周期会出现随机性的跳变, 但最大幅度不超过 300 μ s。

3 结 论

通过对 Windows 环境下帧同步实时性的分析, 对比 RTX 实时扩展的试验结果, 可以看出: Windows 非实时环境下中断响应的不确定性是帧周期出现随机性大幅度偏差的主要原因。RTX 实时扩展方案的试验评估中, 用于帧周期测试的 CPU 计数器接口运行在 Windows 进程, 考虑到 Windows 本身的非实时特性和接口调用的延迟, 可推断 RTX 下帧同步的实时性应优于测试数据所呈现的结果。

研究表明: RTX 可以提供高精度的定时和可靠且稳定的中断响应延迟; 在 RTX 实时环境下基于反射内存中断的帧同步可以保证红外图像生成系统稳定地在 200 Hz 帧频下运行, 达到预期研究目的。

文中的研究成果不仅局限于红外景象投射系统, 也可为其它闭环半实物仿真系统的实时性设计提供参考。

参考文献:

- [1] Su Delun, Wang Shicheng, Liao Shouyi, et al. Dynamic infrared image generation based on CIG technology [J]. *Infrared Technology*, 2009, 31(9): 549-552. (in Chinese)
苏德伦, 王仕成, 廖守亿, 等. 基于 CIG 的动态红外图像实时生成系统[J]. *红外技术*, 2009, 31(9): 549-552.
- [2] Liao Shouyi, Su Delun, Zhang Jinsheng, et al. Real-time measures for image generation system of IR imaging target simulator [J]. *Infrared and Laser Engineering*, 2010, 39(3): 377-382. (in Chinese)
廖守亿, 苏德伦, 张金生, 等. 红外成像目标模拟器图像生成系统的实时性措施 [J]. *红外与激光工程*, 2010, 39(3): 377-382.
- [3] Lowry KG LaW. Development and integration of the army's advanced multispectral simulation test acceptance resource (AMSTAR) HWIL facilities[C]//SPIE, 2005, 5785: 174-183.
- [4] Eric M, Olson J D P. Utilizing commercial off-the-shelf PC graphics hardware for real-time hardware-in-the-loop scene generation[C]//SPIE, 2003, 5092: 221-230.
- [5] Cheng Zhengxi, Ma Bin, Liu Qiang, et al. 128 \times 128 resistor array with suspended micro-bridge fabricated through film transfer process [J]. *Infrared and Laser Engineering*, 2012, 41(3): 559-562. (in Chinese)
程正喜, 马斌, 刘强, 等. 薄膜转移工艺制备的规模高架桥式电阻阵列[J]. *红外与激光工程*, 2012, 41(3): 559-562.
- [6] Zheng Xiaobao, Wang Shicheng, Zhang Jinsheng, et al. Real-time infrared image generation system based on reflective memory network [J]. *Computer Application*, 2008, 28(z2): 344-346, 359. (in Chinese)
郑小宝, 王仕成, 张金生, 等. 基于反射内存网的红外图像实时生成系统[J]. *计算机应用*, 2008, 28(z2): 344-346, 359.
- [7] Ardenice, a Citrix Company. RTX7.1 SDK User Manual[M]. Version 1.0, 2007.