

文章编号: 1672-2892(2010)06-0652-04

基于可编程显卡的信息化雷达终端显示系统

张 辉, 孙立国, 李世丹, 王德生

(清华大学 电子工程系, 北京 100084)

摘 要: 高速发展的可编程图形处理单元(GPU)为信息化、软件化雷达终端系统的发展提供了新的技术平台。本文应用 GPU 强大的并行处理能力和极高的计算效率, 对原有雷达终端显示系统进行改进, 大大降低了 CPU 的占用率, 减轻了系统的负担, 取得了可以与专用雷达图像显示硬件卡相媲美的显示效果。

关键词: 着色程序; 雷达终端; 图形处理单元

中图分类号: TN957.7

文献标识码: A

Informationalized radar terminal display system based on programmable GPU

ZHANG Hui, SUN Li-guo, LI Shi-dan, WANG De-sheng

(Department of Electronic Engineering, Tsinghua University, Beijing 100084, China)

Abstract: The rapid development of programmable Graphics Processing Units(GPU) provides a new technical platform for the development of informationalized software radar terminal system. In this study, the powerful parallel processing capability and high computational efficiency of GPU were applied to improve the display system based on CPU, which greatly reduced the CPU utilization, reduced the burden of the system and obtained excellent display comparable with the hardware implementation of radar image display.

Key words: shader program; radar terminal system; Graphics Processing Unit

多年来计算机图形处理器以大大超过摩尔定律的速度高速发展, 图形处理器的迅速发展使 GPU 具有流处理、高密度并行运算、可编程流水线等特性, 从而极大地拓展了 GPU 的处理能力和应用范围。作为一种新的计算平台, GPU 正被越来越多的研究开发人员所采用。

目前国内雷达终端系统基本上都依赖于专门的硬件处理板, 信息化、软件化雷达终端系统技术的研究和进展相对比较缓慢, 而对于可编程显卡 GPU 的应用更是一片空白, 可编程 GPU 的出现和不断发展给雷达终端技术的软件化和信息化发展前景注入了新的活力和方向。图形处理器灵活的可编程特性以及强大的数据并行处理能力, 彻底解脱了中央处理器 CPU, 极大地降低系统的负担。因此, 国外的很多公司纷纷把技术方向转向了可编程 GPU, 例如, 美国的 CURTISS WRIGHT 公司推出了基于可编程 GPU 的雷达扫描变换软件包, 取得了远超过 CPU 实现的性能和效果。

本文结合国内外终端技术的发展状况, 提出了将可编程 GPU 的并行数据处理的强大能力通过可编程特性, 应用在信息化雷达终端显示系统中, 并针对 GPU 的架构和编程模型, 充分利用 GPU 的运算资源和极高的计算效率, 对原有的雷达终端显示技术进行改进。

1 可编程 GPU 与着色程序 Shader

早期可编程显卡的编程能力主要依靠低层次的汇编语言或硬件规格语言, 开发周期长, 难度大。简单、灵活和方便的编程模式一直是编程语言追求的目标, 显卡编程语言也不例外。可编程着色语言从诞生就被定位为显卡编程的高级语言, 它允许应用程序对图像处理管线的大多数重要阶段施以完全控制, 使软件开发人员不再局限于选择固化在芯片中的图像渲染算法和公式, 而是可以使用这种可编程能力来满足实际的应用需求^[1]。

目前比较流行的可编程着色语言主要有 3 种: 基于 OpenGL^[2]的 OpenGL Shading Language(GLSL)^[1], 微软公

司基于 Direct3D 的 High-Level Shading Language(HLSL)以及 NVIDIA 公司的 Cg^[3]语言。使用可编程着色语言编写的程序称为着色程序。着色程序分为 2 类:顶点着色程序(Vertex Shader Program, VSP)和片断着色程序(Fragment Shader Program, FSP),它们分别对应 GPU 上的 2 个组件:可编程顶点处理器和可编程片断处理器,也就是说顶点着色程序运行在顶点处理器上,片断着色程序运行在片断处理器上。

本文采用 GLSL 语言,也称作 GLslang,它是一个以 C 语言为基础的高级着色语言,由 OpenGL 体系结构审核委员会所建立,允许开发人员控制图形处理管线中最重要的阶段^[1]。

2 坐标扫描变换与顶点着色器

现在大部分的导航、搜索雷达仍使用旋转天线,当雷达回波被数字化采样后,雷达视频按极坐标(距离和方位)形式输出,因此视频必须被转换为直角坐标形式,才能在光栅扫描显示器上显示。坐标扫描变换作为雷达终端显控系统中的一个重要功能,其核心就是将输入的极坐标雷达数据转化为笛卡尔坐标系。

雷达坐标扫描变换一般分为前向扫描和反向扫描变换 2 种。在前向扫描转换中,极坐标距离方位(R, θ)数据通过使用基本的三角函数公式
$$\begin{cases} X = R \sin \theta \\ Y = R \cos \theta \end{cases}$$
 转换为直角坐标(X, Y),在这个转换过程中,会出现一些屏幕像素点没有被映射到的情况,特别是在显示远区的位置上,会有很多连续的点没有视频数据可画,从而在整幅平面位置指示器(Plan Position Indicator, PPI)显示画面的显示远区形成某种“花纹”,影响显示效果,一般把这种现象称为作图“死区”^[4],通常需要通过其他的手段来进行“补点”。

反向扫描转换通过逆向坐标映射来解决漏点问题,其核心思想就是把整个显示区域的每个像素的直角坐标(X, Y)通过反变换公式
$$\begin{cases} R = \sqrt{X^2 + Y^2} \\ \theta = \arctan(X/Y) \end{cases}$$
 映射回极坐标系(R, θ)中,通过最邻近或插值的方法来确保每个显示像素都有对应的采样值,从而确保漏点的情况不会出现。

扫描变换是一个运算量大且计算重复的过程。早期的雷达终端显控系统专门采用一个嵌入式单板计算机来完成扫描变换,现已装备的雷达终端系统基本上采用复杂可编程逻辑芯片(Complex Programmable Logic Device, CPLD)或 FPGA 来协助主处理器完成。多年以来,基于硬件实现的雷达坐标扫描转换是实现高质量、高分辨力雷达显示唯一现实的解决办法。但随着图形处理卡 GPU 并行数据处理能力的不断增强和可编程特性,结合多核中央处理器 CPU,已经彻底改变了这种状况。

可编程显卡的顶点着色器可用来指定一系列完全通用的,用于各个顶点及其数据上的操作,而且顶点着色程序比片断着色程序更适合做几何运算,故可以使用顶点着色程序来完成雷达坐标扫描变换。由于在前向扫描变换中“补点”需要对漏点的位置进行判断,而在 GPU 进行中数据流的分支和判断会大大影响数据的执行效率,因此采用反向扫描变换更适合发挥 GPU 的并行处理能力,顶点着色程序的处理过程如图 1 所示。

相比硬件扫描转换卡所具有的初始成本与维护成本高的缺点,基于顶点着色器的坐标扫描变换的软件化的解决方案为终端系统提供了很高的性能和较大的灵活性,从而可以满足不断变化的需求,并且以较低的成本,取得了更高的可靠性和易维护性。

3 余辉与片断着色器

余辉对于雷达 PPI 显示来说是必要的:目标在显示器上亮度强,衰减慢;噪声在显示器上亮度弱,衰减快,这使目标很突出。如果是运动目标,会产生拖尾效应,指示目标运动方向,使运动目标更容易被发现^[4]。

模拟余辉最好的方法是对整个雷达 PPI(Plain Position Image)显示区域内的每个像素进行衰减,但这样做(例如 1024×1024 分辨率,每次余辉需要进行 1024×1024 次减运算)将带来极大的运算量。因此传统的软件化模

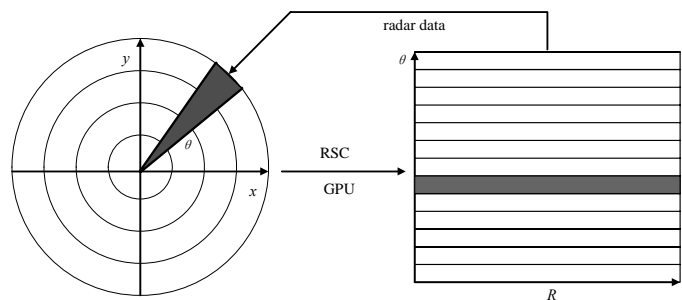


Fig.1 Process of vertex shader program

图 1 顶点着色程序处理过程

拟余辉方法是采用随机访问若干点的方法来对 PPI 显示区域进行余辉处理,只要在一定时间内保证能够使当前扫描线前端衰减为 0,就比较好地实现了余辉功能^[4]。但是这种软件化余辉实现技术,一方面 CPU 的占用率仍然很高,造成了不小的系统开销;另一方面会使 PPI 显示效果大打折扣,整个显示画面不平滑,很粗糙,出现沙粒感。

GPU 专为计算密集型、高度并行化的计算而设计的,它允许以更快的速度将图形图像渲染到显示帧缓存中。运行在片断处理器上的片断着色程序负责对光栅化后的每个片断进行独立的颜色计算,而且片断着色器相比顶点着色器的一个突出特点就是能够检索纹理。由于余辉需要对 PPI 显示区域内的每个像素的灰度值进行衰减,因此需要检索雷达视频纹理数据,故采用片断着色器来实现余辉。

在片断着色程序中余辉的处理过程见图 2,其中 radar value 为传进来的雷达纹理灰度值,decay value 为每次余辉衰减的灰度值,YhOut 是衰减后的雷达灰度值。

余辉的效果对比如图 3 所示。

4 多层图像叠加与片断着色器

在军事、航海等领域,都有在电子海图上叠加雷达一次视频及标绘 ARPA(Automatic Radar Plotting Aid)和二次目标信息的应用需要(见图 4)。如果单独使用电子海图或雷达,虽然可以在一定程度上满足应用需求,但它们都有各自的缺陷(电子海图无法显示动态目标,雷达则不能识别周围目标的属性),只有将电子海图与雷达信息叠加显示,才可实现两者的优势互补^[5]。因此实现图像的分层叠加显示技术对于信息化雷达终端系统是必备的。

由于雷达系统操作的实时性和快速响应,很多已装备的雷达终端系统大都采用专门设计的多层叠加显示硬件板,来支持分层的快速叠加显示,即所谓的“硬叠加”。这种硬件叠加方式虽然解决了快速叠加显示问题,但也无形加大了终端系统开发的难度,增加了设备的成本,提高了维护的费用。利用 GPU 的可编程特性和强大的计算能力,可以软件化实现图像的多层叠加显示,同时又不降低显示的质量。如前所述,片断着色器控制像素颜色的计算过程,因此可以使用片断着色器来进行图像的多层叠加显示。叠加操作不是简单的覆盖,它与 alpha 分量相关,alpha 分量定义了 2 个像素组合的透明度。在实际应用中,雷达一次信息是半透明叠加在电子海图上,透明度由 alpha 分量来指定。ARPA 和二次信息则完全叠加覆盖在雷达一次视频上。

实际计算过程如下:假设电子海图像素颜色的 3 个分量(红色,绿色,蓝色)为 (M_R, M_G, M_B) ,雷达一次视频颜色的 3 个分量(红色,绿色,蓝色)为 (R_R, R_G, R_B) ,alpha 为透明度,则二者叠加后视频的颜色为 $(M_R + R_R \times \alpha, M_G + R_G \times \alpha, M_B + R_B \times \alpha)$ 。关于 ARPA 和二

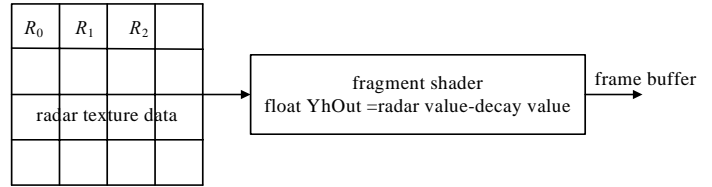


Fig.2 Process of persistence
图 2 余辉处理过程

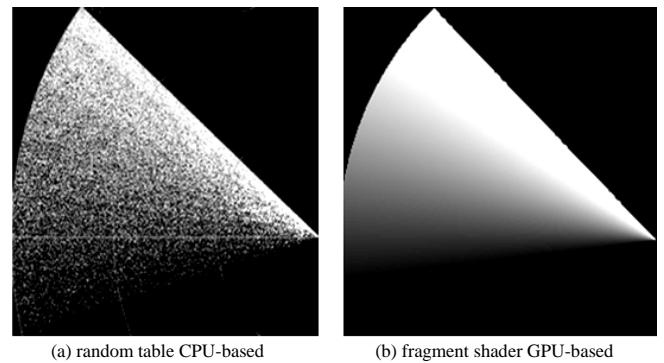


Fig.3 Comparison of the effect of persistence
图 3 余辉效果对比图

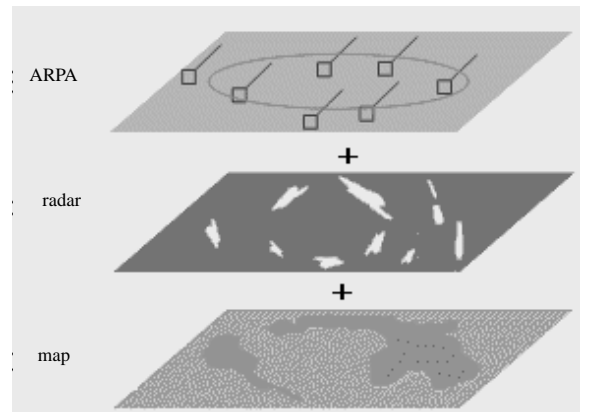


Fig.4 Schematic diagram of multi-layer stacking
图 4 多层叠加示意图

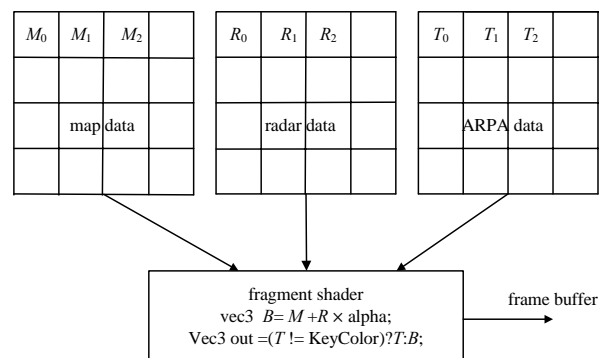


Fig.5 Process of multi-layer stacking
图 5 多层叠加显示处理过程

次信息的叠加则可以使用关键色透明技术,只要是关键色就透明,即显示海图和一次视频叠加混合后的颜色,如果不是关键色的,则覆盖。通过关键色来指示对应像素位置上是否需要标绘 ARPA 和二次信息,叠加的具体过程如图 5 所示: M 为传进来的底层电子海图纹理值, R 为雷达一次视频纹理值, T 为 ARPA 和二次目标信息, B 为底层地图和一次视频混合叠加后的颜色, out 为多层图像叠加后的最终结果。

由于叠加混合的过程是在 GPU 中运算完成的,不占用 CPU 时间,因而具有相当高的执行效率以及极佳的图像显示效果。

5 结论

开发计算机平台配置为:中央处理器 CPU 为 Intel Core 2 Duo 2.0 GHz,可编程显卡为 ATI Mobility Radeon X1400,显卡驱动支持 OpenGL 2.0。实验结果:当雷达分辨率为 1024×1024 ,PPI 显示转速为每圈 1 s 时,CPU 的占用率仅为 2%~3%。由此可见,采用可编程的渲染管线,可以充分发挥显卡的并行计算能力。

本文结合 GPU 的高密集并行运算和可编程流水线的特性,将 GPU 的处理能力和应用范围拓展到了信息化、软件化雷达终端领域。改进了基于 CPU 的软件化雷达终端显示系统,充分发挥显卡的渲染能力,大大降低了 CPU 的占用率,减轻了系统的负担,取得了可以与专用雷达图像显示硬件卡相媲美的显示效果。

参考文献:

- [1] Randi J Rost. OpenGL Shading Language[M]. 2nd Edition. New York:Addison Wesley Professional, 2004.
- [2] OpenGL Architecture Review Board,Dave Shreiner,Mason Woo,et al. OpenGL Programming Guide[M]. 6th Edition. New York:Addison Wesley Professional, 2007.
- [3] Mark W R,Glanville R S,Akeley Kurt,et al. Cg:A system for programming graphics hardware in a C-like language[J]. ACM Transactions on Graphics, 2003,22(3):896-907.
- [4] 王德生,赵利民,孙立国,等. 信息化、软件化、通用雷达终端的构建与实现[J]. 现代雷达, 2007,29(12):22-26. (WANG De-sheng,ZHAO Li-min,SUN Li-guo,et al. Study and Implement of Software Radar Terminal System Based on Information Packages[J]. Modern Radar, 2007,29(12):22-26.)
- [5] 刘毅,索继东,曹兰兰. 基于 GDI+的电子海图与雷达图像的叠加显示[J]. 航海技术, 2009(1):43-45. (LIU Yi,SUO Ji-dong, CAO Lan-lan. Superimposition Display for Electronic Chart and Radar image based on GDI+[J]. Marine technology, 2009 (1):43-45.)

作者简介:



张 辉(1987-),男,山西省怀仁县人,在读硕士研究生,主要研究方向为信号处理、雷达终端技术.email:zhanghui07@mails.tsinghua.edu.cn.

孙立国(1982-),男,山东省潍坊市人,博士,主要研究方向为数据融合、雷达终端技术。

李世丹(1983-),男,辽宁省大连市人,博士,主要研究方向为数据融合、雷达终端系统。

王德生(1946-),男,北京市人,博士生导师,主要研究方向为雷达信号处理和终端系统。