

文章编号: 1672-2892(2010)04-0415-05

## 一种基于 GPU 的高效合成孔径雷达信号处理器

俞惊雷, 柳彬, 王开志, 刘兴钊, 郁文贤

(上海交通大学 电子系, 上海 200240)

**摘要:** 随着合成孔径雷达(SAR)应用的不断扩展, 其所需要处理的数据量也在不断增加, 传统的 SAR 信号处理器的处理速度成为其应用扩展的瓶颈。为了应对这些挑战, 需要高效的 SAR 信号处理器来加快计算速度。文章利用图形处理器(GPU)这一新颖高效的计算平台进行 SAR 信号处理, 利用 GPU 通用并行计算, 使用 CUDA 实现 SAR 成像算法, 充分发挥其计算能力。实验结果表明, 其处理速度是基于 CPU 的传统 SAR 信号处理器的 10 倍以上。它为在未来 SAR 信号处理中可能出现的问题提供了一种可靠的方法。

**关键词:** 合成孔径雷达; 信号处理器; 图形处理器; 通用计算; CUDA 编程模型

**中图分类号:** TN958

**文献标识码:** A

## A highly efficient GPU-based signal processor of Synthetic Aperture Radar

YU Jing-lei, LIU Bin, WANG Kai-zhi, LIU Xin-zhao, YU Wen-xian

(Department of Electronic Engineering, Shanghai Jiao Tong University, Shanghai 200240, China)

**Abstract:** With its extending applications, Synthetic Aperture Radar(SAR) is required to process more and more data. The processing speed of traditional SAR processor turns out to be a bottleneck of the extension of SAR applications. Highly efficient SAR signal processors need to be explored towards these challenges. Recently, Graphics Processing Unit(GPU) is developing at top speed, which is a novel and promising computation platform for efficient SAR processors. A GPU-based SAR signal processor is introduced in this paper. The implementation of this processor takes full advantage of GPU's computation capability, using general purpose computations to implement the SAR imaging based on CUDA programming model. As the result shows, this processor is more than ten times as fast as traditional CPU-based SAR processor. It provides a promising way to solve the problems of SAR signal processing in the future.

**Key words:** Synthetic Aperture Radar; signal processor; GPU; general purpose computations; Compute Unified Device Architecture(CUDA)

随着合成孔径雷达(SAR)应用的扩展, 要求 SAR 系统可以产生令人满意的雷达图像, 能够在严酷的运行环境下具有较高的分辨力和稳定性<sup>[1]</sup>。

为了应对这些新的挑战, SAR 信号处理不得不增加复杂性, 但为了生成实时的 SAR 图像, 必须减少处理时间, 而且新的图像要比传统的图像大得多。因此, 急需一种高效的 SAR 信号处理方法。

最近, 图形处理器(GPU)正在高速发展, 对 SAR 信号处理器来说, 这是一种新颖的、有前景的高效计算平台。GPU 是为解决 3-D 时代的复杂图形处理计算而产生的。基于 GPU 实现的通用计算具有高效率、低功耗的特点。

在过去几年, GPU 和 CPU 相比, GPU 的浮点计算性能和存储器带宽有大幅度的增长。图 1 比较了各种主流 GPU(ATI 和 NVIDIA)与英特尔的 CPU 的浮点计算能力<sup>[2]</sup>。在达到相同浮点计算性能和存储器带宽的情况下, GPU 比 CPU 耗费更少的资金和电力<sup>[3]</sup>。

最近几年, 开发人员使用诸如 Cg(Computer Graphics), GLSL(GL Shading Language)和 HLSL(High Level Shading Language)等高级着色器语言进行 GPU 编程来实现通用计算, 但这需要开发人员熟悉大量的硬件知识和

GPU 的应用程序编程接口。CUDA 的发布改变了这一情况。CUDA 是 C 语言的扩展,它为开发人员提供了一种熟悉的 GPU 编程模型,不需要了解特定的 GPU 硬件和编程设计接口就可以在 GPU 上实现通用计算。

在 GPU 上实现 SAR 成像算法的前期工作可以在参考文献[4]和文献[5]中找到。在文献[5]中只是简单概述,并没有实现的细节和结果分析。在文献[4]中,Blom 和 Follo 在 GPU 上实现了后向投影算法。

在本文中实现一种基于距离徙动校正算法(Range Cell Migration Correction, RCMC)的 SAR 信号处理器。仿真结果显示处理器生成的图像结果与理论值一致。即使实际的 SAR 数据比 GPU 的存储空间要大得多,但是通过实际的 SAR 数据的处理时间显示,这种处理器比基于 CPU 的 SAR 处理器要快 10 倍以上。相比之下,文献[4]假设了所有的相关数据(输入数据,中间数据,结果数据和输出数据)一次性放入 GPU 的存储空间中。因此,本文介绍的基于 GPU 的 SAR 处理器是十分高效且实用的。它提供了一种新颖的、有前景的高效计算平台,以解决未来 SAR 信号处理可能出现的问题。

## 1 CUDA 编程模型

在 CUDA 编程模型中, GPU 被视为一种多核处理器,适用于数据的并行计算,如 SAR 信号处理。

### 1.1 主机、设备与核

CUDA 认为 CPU 是主机(host),而 GPU 则是设备(device)。函数(function)被称为核,它由主机调用并在设备上运行。本文介绍的处理器由多个核实现。

### 1.2 线程组层次结构

CUDA 能够并行地执行和管理大量的线程。这些线程在 GPU 上执行,它们被组织为一个 3 层结构。线程被组织成为线程块,多个线程块在一个线程网格中运行。同一个块内的线程可以彼此协作,它们通过一些共享存储器来共享数据,并同步执行来协调存储器的访问<sup>[6]</sup>。由于一个块内的所有线程处于同一个处理器内核中,因此每个块内的线程数会受限于处理器内核的存储资源。

一个核可以由多个形状相同的线程块执行,因此,线程总数等于每个线程块的线程数乘以线程块数。网格内的线程块的数量总是由处理的数据量决定的,而不是由 GPU 硬件的处理器数量决定。

以 Nvidia Quadro FX3700 为例,每个块内最多有 512 个线程,每个网格内最多有 65 535 个线程块。

### 1.3 存储器层次结构

CUDA 线程可以在执行时访问多种存储器的数据。文献[7]介绍了 CUDA 的存储器属性。每个内核需要根据存储量、访问速度、是否为只读等属性来选择合适的存储器。

### 1.4 CUFFT 库

CUFFT(Compute Unified Fast Fourier Transform)是 CUDA 编程环境中所使用的一个 FFT(Fast Fourier Transform)库。CUFFT 在 GPU 上能发挥出高性能<sup>[7]</sup>。本文使用 CUFFT 库来进行 FFT/IFFT(Inverse Fast Fourier Transform)的计算。

## 2 基于 GPU 的 SAR 信号处理器实现

在本文中,使用纹理贴图(计算机图形学中的重要概念)的方法实现 RCMC<sup>[8]</sup>,在一些相关的论文中已作详细阐述,由于篇幅限制,故不再赘述。

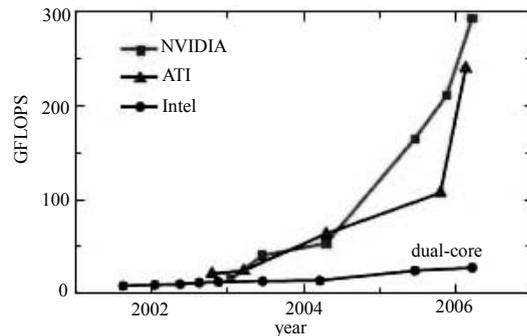


Fig.1 Floating-point operation per second for GPU(NVIDIA and ATI) and CPU(Intel)<sup>[2]</sup>

图 1 GPU(NVIDIA 和 ATI)与 CPU(Intel)每秒的浮点运算次数<sup>[2]</sup>

基于 GPU 处理器的流程图如图 2 所示。SAR 系统接收的数据称为信号数据，它的处理步骤如下。

1) 将信号数据从 CPU 拷贝到 GPU;

2) 通过多线程的匹配滤波器对信号进行距离向压缩。先作距离向 FFT，然后通过一个距离向匹配滤波器，最后作距离向 IFFT 来完成距离向压缩;

3) 通过方位向 FFT 将距离向压缩后的数据变换到距离多普勒域;

4) 在距离多普勒域利用纹理贴图实现 RCMC。经过处理后，不同距离的徙动曲线得到拉直，并与方位轴平行;

5) 方位向压缩由多线程的方位向匹配滤波器完成;

6) 方位向 IFFT 将数据变回到时域，并得到充分聚焦的复图像;

7) 将复图像数据从 GPU 拷贝回 CPU。

与真实的 SAR 数据相比，GPU 的存储空间十分有限。以 Nvidia Quadro FX3700 为例，它的存储空间只有 512 MB。

本文提出的基于 GPU 的处理器可以在 CPU 的帮助下自适应调节数据量的大小。在处理前，CPU 会根据 GPU 能够一次性处理的最大数据量自动将原始数据切成数据块。然后，CPU 将每一块原始数据块拷贝到 GPU 中进行成像，成像后再将相应的图像块拷贝回来。CPU 会拼接所有的成像后图像块并最终生成整幅处理后的图像。所以，本文提出的基于 GPU 的处理器能够处理真实的 SAR 数据，其数据大小可达  $32\ 768 \times 32\ 768$ ，甚至  $65\ 536 \times 65\ 536$ ，能够满足实际所需。

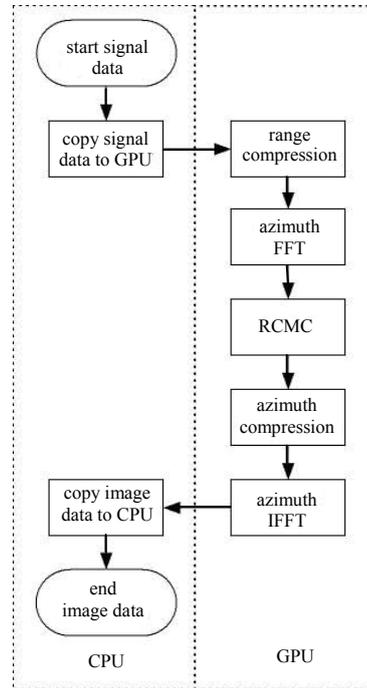


Fig.2 Block diagram of implementation of the GPU-based SAR processor  
图 2 基于 GPU 的 SAR 处理器的实现流程图

### 3 实验结果

为了检验本文提出的基于 GPU 的 SAR 处理器成像结果，在场景中放置 3 个目标，如图 3 所示，这 3 个目标在方位向位置相同，从 SAR 到目标 A、B 和 C 的最短斜距分别是 15 km、20 km 和 25 km。表 1 是仿真参数。

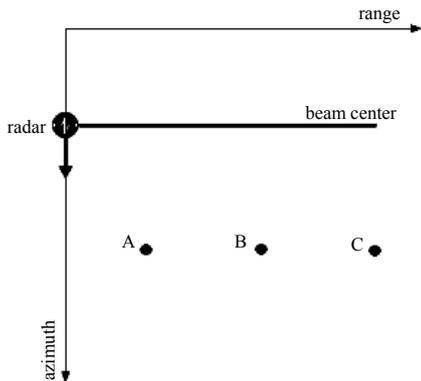


Fig.3 Positions of the three targets used in the simulation  
图 3 三个仿真目标的位置

表 2 是处理器的成像结果。距离向分辨率的理论值是 1.34 个距离向采样。由于 3 个目标的方位向分辨率理论值是不同的，因此表 2 中的方位向分辨率定义为：

$$R_{az} = R_{sim} / R_{th} \tag{1}$$

式中： $R_{az}$  是表 2 中的方位向分辨率； $R_{sim}$  是方位向分辨率的测得值； $R_{th}$  是方位向分辨率的理论值。从表 2 中可以看到，仿真数据的成像结果测得值与理论值一致。

表 1 机载 SAR 仿真参数  
Table 1 Airborne SAR parameters in the simulation

parameter name	value
radar centre frequency	1.27 GHz
transmitted pulse duration	2.5 $\mu$ s
range FM rate	8 MHz/ $\mu$ s
signal bandwidth	20 MHz
range sampling rate	30.8 MHz
Doppler bandwidth	250 Hz
PRF	469.5 Hz

表 2 仿真目标的成像结果  
Table 2 Imaging results of the simulated targets

	range resolution(samples)	azimuth resolution	range PSLR/dB	azimuth PSLR/dB
A	1.393	1.035	-13.98	-14.12
B	1.390	1.027	-13.89	-13.98
C	1.392	1.032	-13.92	-14.10

图 4 是 3 个仿真目标的成像结果。为了更加细致地观察成像结果, 图 5 和图 6 分别展示 3 个目标的距离像和方位像的剖面图。

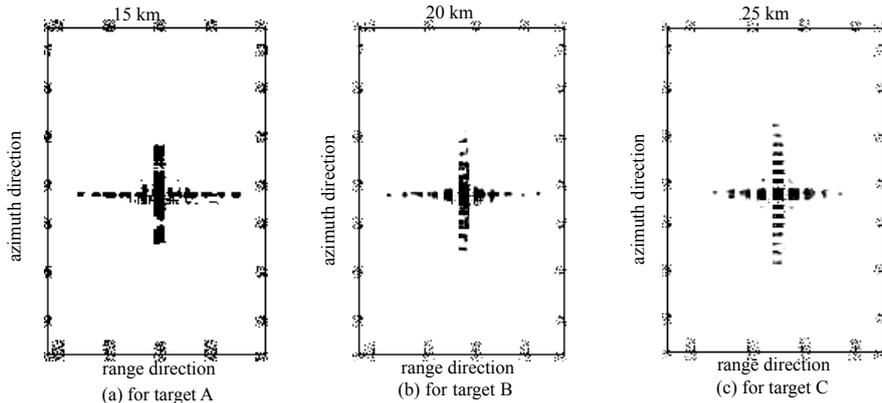


Fig.4 Imaging results of the simulations  
图 4 成像仿真结果

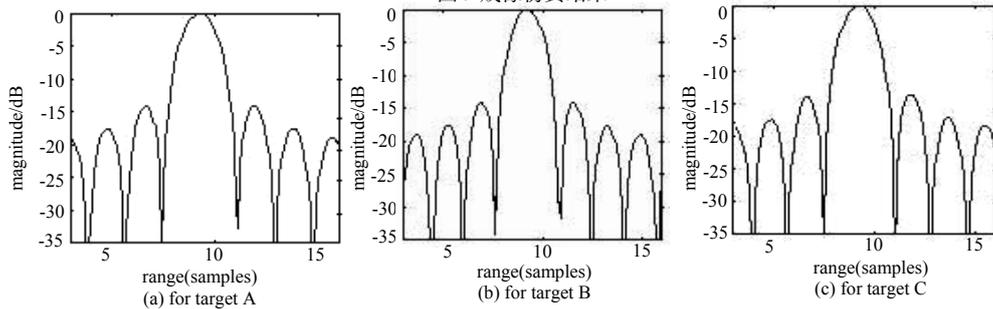


Fig.5 Range profiles of imaging results  
图 5 成像结果距离向剖面图

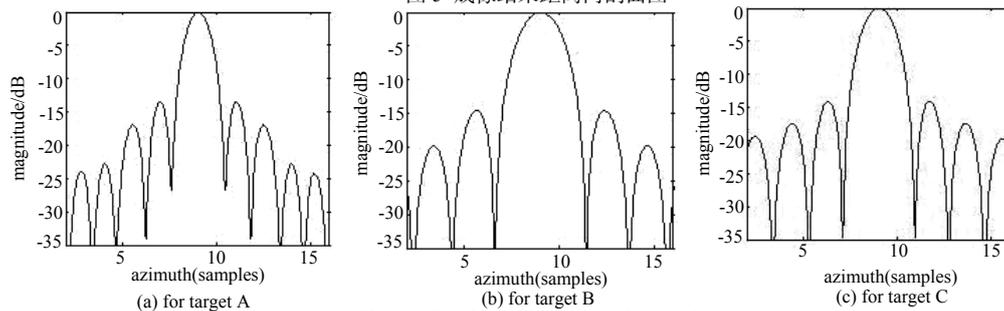


Fig.6 Azimuth profiles of imaging results  
图 6 成像结果方位向剖面图

图 7 是真实机载 SAR 数据的成像结果, 其大小为  $8192 \times 16384$ 。

#### 4 效率测试和讨论

本文提出的基于 GPU 的 SAR 处理器是十分高效的。表 3 比较了 GPU 和 CPU 在不同大小的输入数据下处理时间的测试结果。测试平台由 Intel 的 CPU Xeon E5140 和 Nvidia 的 GPU Quadro FX3700 组成。

在表 3 中, 条件 1 只使用了单核 CPU, 在条件 2 下, 利用 OpenMP 使用了 CPU 的全部 4 个核。条件 4 只列

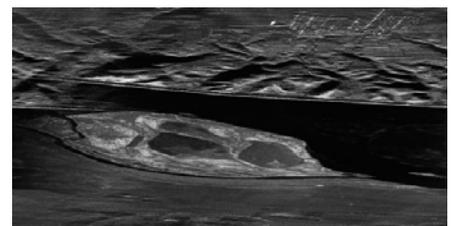


Fig.7 Imaging results of actual airborne SAR data  
图 7 真实机载 SAR 数据的成像结果

出了 GPU 上的处理时间, 而条件 3 列出了 GPU 上的处理时间以及 CPU 与 GPU 之间的传输时间。

根据第 2 节的讨论, 本文提出的基于 GPU 的处理器是根据原始数据大小而自适应的。因为 CPU 切割与拼接数据很快, 所以即使数据量远远大于 GPU 存储空间, 基于 GPU 的处理器仍然比基于 CPU 的处理器高效得多。实验结果表明, 本文介绍的基于 GPU 的处理器比基于 CPU 的处理器更加实用、高效, 在处理速度上快了 10 倍以上。

表 3 基于 CPU 和基于 GPU 的 SAR 处理器处理时间比较  
Table3 Comparison of processing time for SAR processors based on CPU and GPU

condition	data size				
	2 048×2 048	4 096×2 048	4 096×4 096	8 192×4 096	8 192×8 192
1	9.296	19.471	39.089	81.056	164.997
2	2.652	5.484	10.296	23.851	47.908
3	0.065	0.124	0.636	1.299	2.556
4	0.013	0.024	0.392	0.813	1.566
condition	data size				
	16 384×8 192	16 384×16 384	32 768×16 384	32 768×32 768	
1	336.978	670.497	1 347.226	2 702.689	
2	96.127	190.378	378.684	764.176	
3	5.298	11.467	21.346	44.962	
4	3.275	6.872	12.718	25.711	

## 5 结论

本文引入了 CUDA 这种并行计算架构, 利用 GPU 并行计算引擎来解决复杂计算问题。介绍了一种高效的基于 GPU 的 SAR 信号处理器, 该处理器与传统的 CPU 实现的处理器拥有同样的成像能力, 但耗费的时间与占用的资源更少, 它为未来解决 SAR 信号处理的问题提供了更广阔的前景。

致谢: 感谢中电集团第 38 研究所提供的机载 SAR 的原始数据。

### 参考文献:

- [1] Wang Kaizhi. High Resolution SAR Imaging Techniques under Squint Mode[D]. Shanghai:Shanghai Jiao Tong University, 2007.
- [2] Owens John D,Luebke David,Govindaraju Naga,et al. A Survey of General-Purpose Computation on Graphics Hardware[J]. Computer Graphics Forum, 2007,26(1):80-113.
- [3] Govindaraju Naga,Gray Jim,Kumar Ritesh,et al. GPU TeraSort:high performance graphics co-processor sorting for Large da-tabase management[C]// International Conference on Management of Data, Chicago:[s.n.], 2006:325-336.
- [4] Blom M,Follo P. VHF SAR Image Formation Implemented on a GPU[C]// Geoscience and Remote Sensing Symposium, Seoul Korea:[s.n.], 2005:3352-3356.
- [5] Nickolls John,Buck Ian,Garland Michael,et al. Scalable parallel programming with CUDA[J]. International Conference on Computer Graphics and Interactive Techniques, 2008,6(2):40-53.
- [6] Vasily Volkov,Brian Kazian. Fitting FFT onto the G80 Architecture[EB/OL]. [2008-08-14]. <http://www.cs.berkeley.edu/~kubitron/courses/cs258-S08/projects/reports/project6-report.pdf>.
- [7] Svetlin A,Manavski,Giorgio Valle. CUDA compatible GPU cards as efficient hardware accelerators for Smith-Waterman sequence align-ment[J]. BMC Bioinformatics, 2008,9(z2):S10.
- [8] Paul S Heckbert. Fundamentals of Texture Mapping and Image Warping[D]. USA:Berkeley,CA,University of California at Berkeley, 1989.

### 作者简介:



俞惊雷(1986-), 男, 上海市人, 硕士, 主要研究方向为 SAR 数据成像并行处理. email:yujing lei\_1222@yahoo.com.cn.

柳彬(1987-), 男, 上海市人, 博士, 主要研究方向为 GPU 的高性能计算、SAR 数据成像并行处理。

刘兴钊(1962-), 男, 四川省巴中市人, 博士, 教授, 主要研究方向为自适应信号处理、阵列信号处理、信号检测与估值等。

王开志(1977-), 男, 四川省绵阳市人, 硕士, 副教授, 主要研究方向为微波成像、微波全息图像、信号的时频分析与分数阶算子理论。

郁文贤(1964-), 男, 上海市人, 博士, 教授, 主要研究方向为低分辨雷达目标波形动态识别理论与系统技术、高分辨率雷达图像目标识别理论与技术。