无人机载MiniSAR实时成像处理GPU异步优化

袁溆东[1],雒梅逸香[1],王智超[2],谭佳伟[3],王　峰[1]
(1.复旦大学 信息科学与工程学院,上海 200433;2.32033部队,海南 海口 570100;3.31453部队,辽宁 沈阳 110000)

摘　要: 合成孔径雷达(SAR)以其全天候、全天时的工作特性及其分辨率不随平台高度变化的成像特性,已成为航天遥感、目标检测领域重要的传感器之一。SAR算法复杂度往往与成像分辨率呈正相关,其中计算量问题成为雷达成像实时性的一大挑战。无人机载MiniSAR具有小型化、低功耗、灵活性强和隐蔽性强等优点,其小型化使设备计算能力受限,加剧了复杂度与分辨率之间的矛盾。图形处理单元(GPU)和多线程技术发展迅速,为无人机载MiniSAR实时成像提供了平台。本文根据实时处理机数据流和GPU异构系统的特点,提出了一种GPU异步优化方案,该方案可明显提高中央处理单元(CPU)与GPU之间的并行工作效率,节约大部分的数据存取开销。实验结果证明:GPU的成像效率是单CPU系统的12倍左右,在此基础上,使用GPU异步优化方案后效率可继续提升15%左右。本文提出的设计思路可显著缓解无人机载MiniSAR的实时成像计算压力。

关键词: 合成孔径雷达(SAR);无人机(UAV);实时处理机;图形处理单元(GPU);异步优化
中图分类号: TN 92　　**文献分类号:** A　　**DOI:** 10.19328/j.cnki.2096-8655.2023.04.002

GPU Asynchronous Optimization for UAV-Borne MiniSAR Real-Time Imaging Processing

YUAN Xudong[1], LUOMEI Yixiang[1], WANG Zhichao[2], TAN Jiawei[3], WANG Feng[1]
(1.School of Information Science and Technology, Fudan University, Shanghai 200433, China; 2.32033 Troops of PLA, Haikou 570100, Hainan, China; 3.31453 Troops of PLA, Shenyang 110000, Liaoning, China)

Abstract: Synthetic aperture radar (SAR) has become one of the important sensors in the fields of aero-space remote sensing and target detection. It can work all day in all weather conditions, and has imaging characteristics that the resolution does not change with the height of the platform. The complexity of the SAR algorithm is often positively correlated with the imaging resolution, and the amount of calculation has become a major challenge for the real-time radar imaging. Unmanned aerial vehicle (UAV)-borne MiniSAR has the advantages such as miniaturization, low power consumption, strong flexibility, and strong concealment. Its miniaturization limits the calculation power of the device, which exacerbates the contradiction between complexity and resolution. The rapid development of graphics processing unit (GPU) and multi-threading technology provides a platform for UAV-borne MiniSAR real-time imaging. According to the characteristics of real-time processor data flow and GPU heterogeneous system, this paper proposes a GPU asynchronous optimization method. This method can significantly improve the parallelism between the central processing unit (CPU) and GPU, and can save most of the file access time. The experimental results demonstrate that the imaging efficiency of GPU is about 12 times that of a single CPU system. The efficiency can continue to increase by about 15% after using the GPU asynchronous optimization method. This design idea can greatly relieve the calculation pressure of UAV-borne MiniSAR real-time imaging processing.

Key words: synthetic aperture radar (SAR); unmanned aerial vehicle (UAV); real-time processor; graphics processing unit (GPU); asynchronous optimization

收稿日期:2023-03-23;**修回日期:**2023-04-28
基金项目:国家自然科学基金(61991421、61901122);上海市自然科学基金(20ZR1406300);上海航天科技创新基金(SAST2019-073)
作者简介:袁溆东(1998—),男,硕士研究生,主要研究方向为SAR成像优化算法。
通信作者:王　峰(1984—),男,博士,副研究员,主要研究方向为空间电磁信息智能感知。

0 引言

合成孔径雷达(Synthetic Aperture Radar, SAR)有着全天候、全天时的工作特性以及分辨率不随平台高度变化的成像特性[1],是航天遥感、目标检测领域重要的传感器之一。SAR系统的平台通常是卫星、飞机、导弹、工程车、热气球等。随着电子技术和航空技术的发展,无人机以其小型化、低功耗、灵活性强和隐蔽性强等特点倍受关注[2-4],常被用于雷达数据的采集。实时无人机载成像系统的研究前景广阔,实时性不仅有利于系统及时保存成像数据,且有利于无人机探测感知一体化的设计[5-6]。

庞大的计算量是实现实时无人机载MiniSAR的一大挑战,当前的研究分为基于成像原理的算法优化[1]和基于硬件结构的算法优化[7]。前者改变计算量和计算方法,可减少算法复杂度,但往往会伴随着成像质量下降的风险。后者不改变计算量和计算方式,在提高计算效率的同时可以保证成像质量。本文工作围绕后者进行展开。

硬件的选择需要结合算法特点,相比于中央处理单元(Central Processing Unit,CPU)与现场可编程门阵列(Field-Programmable Gate Array, FPGA),图形处理单元(Graphics Processing Unit, GPU)更适合用于SAR成像算法。一方面,将CPU与GPU进行对比[8],虽然CPU中的单个核心频率要比GPU高得多[9],逻辑预测的能力要比GPU快得多,但核心数量远不及GPU,GPU可并行执行的线程数量远超于CPU,使GPU的并行计算能力大大优于CPU,对于并行度高的SAR算法,使用GPU的计算方式进行优化,可以极大加快运算的速度,因此GPU比CPU更适合承担SAR实时系统的计算任务;另一方面,将FPGA与GPU进行对比。FPGA擅长处理通信密集型运算[10-11],GPU擅长处理计算密集型运算[11-14],而SAR属于计算密集型运算而非通信密集型运算,所以GPU相比于FPGA更适合用于SAR成像的处理。

GPU擅长处理具备以下3种特点的计算任务[15]:1)计算需求量大,其算法可以保证任务占满GPU内部的计算资源,防止资源浪费;2)计算并行程度高,其算法可以保证GPU开启更多的并发任务,提高GPU各时刻计算核心的活跃度;3)吞吐量的重要性大于延迟,其算法可以降低GPU对单线程处理速度的要求,而让GPU着重于单位时间内的任务处理量。实时SAR成像算法同时满足以上3个条件。1975年,英特尔(Intel)创始人Gordon Earle Moore提出了摩尔定律[16]。1999年,英伟达(NVIDIA)半导体公司首次提出了GPU的概念[17]。近20年来,尽管面对诸多挑战,GPU的发展仍旧突飞猛进,实时SAR结合GPU的研究前景辽阔。

算法的选择需要结合实际场景。1991年,Cafforio等[18]在处理地震信号时,根据信号时频域和频谱带宽的特点,采用了波数域的算法进行信号分析。目前,SAR系统逐步微型化[19],可被搭载于无人机。为缓解无人机平台的计算压力[20],FMCW波形逐步取代了LFM波[21-22],距离徙动算法(Range Migration Algorithm,RMA)也随之被广泛使用[23]。相比于后向投影算法(Back Projection Algorithm, BPA)算法,RMA精度相当且计算更快,适合用于无人机载的场景。因此,本文的核心算法选择RMA。

本文通过结合SAR成像计算特点和GPU硬件结构及其计算特性,设计出了一种无人机载MiniSAR实时处理机,并针对该处理机提出了一种多CPU线程的优化方案,可以在不改变算法原理和硬件结构的条件下缩短15%的计算时间,最后使用FUSAR-Ku系统进行了实验验证。本文内容组织如下:第1节介绍了FUSAR-Ku成像算法原理以及运动补偿原理和GPU异步系统结构;第2节介绍了实验方法以及实验结果;第3节对实验结果进行了总结。

1 FUSAR-Ku系统与GPU数据流

实验设备是复旦大学电磁波信息科学教育部重点实验室的FUSAR-Ku系统[24-25]。综合考虑成像精度、设备性能,FUSAR-Ku系统的核心算法折中选择RMA。

1.1 成像算法

FUSAR-Ku系统成像部分是在RMA基础上进行改进的,RMA处理流程如图1所示。RMA要在波数域进行插值操作,具体步骤为脉冲压缩、二维快速傅里叶变换(Fast Fourier Transform, FFT)、运动补偿、距离徙动校正、方位向脉冲压缩、二维逆傅里叶变换(Inverse Fast Fourier Transform,IFFT)。

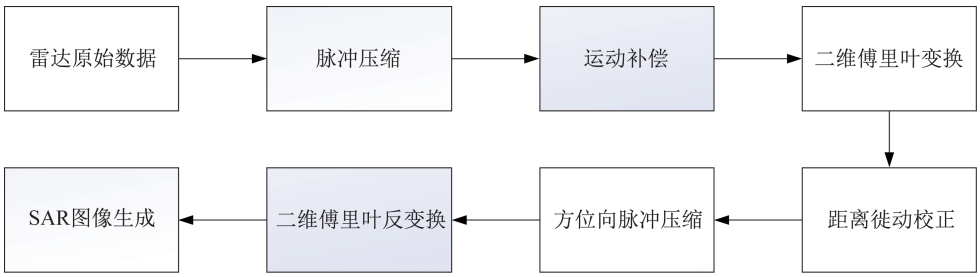

图 1　RMA 处理流程

Fig. 1　Processing flow of the RMA

但 RMA 的 STOLT 插值无法满足无人机平台运动补偿的要求，为此，对其进行改进得到了分段孔径成像（Segment Aperture Imaging，SAI）算法[23]。该算法基于 RMA 在波数域实现 RCMC 与运动补偿，将原算法分段处理，并且分解为级数相乘的滤波器以完成成像。

1.2　GPU 异构数据流

英伟达公司针对 GPU 提出了统一计算设备架构（Compute Unified Device Architecture，CUDA）。CUDA 是由 CPU 和 GPU 组成的异构平台，CPU 为主机端（Host），适合处理逻辑运算，GPU 为设备端（Device），适合处理并行化运算。CPU 和 GPU 之间依靠 PCIE（Peripheral Component Interconnect Express）总线通信。CUDA 编程的异步执流程如图 2 所示，具体流程如下：1）CPU 和 GPU 两端分别分配内存；2）将 CPU 端的数据拷贝到 GPU 端；3）CPU 呼叫核函数，GPU 执行核函数；4）将结果拷贝回 CPU 端。在核函数 N 处理阶段和 CPU 处理阶段 $N+1$ 之间，可选择同步处理或者异步处理。

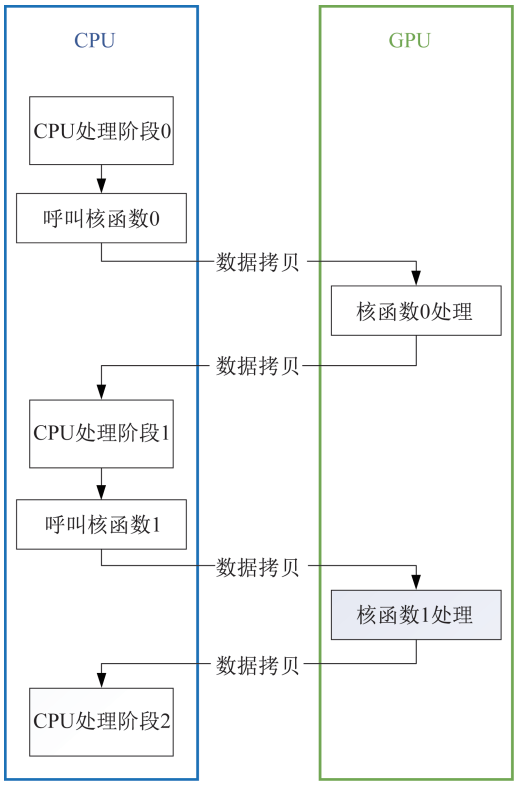

图 2　CUDA 异步执行流程

Fig. 2　Asynchronous execution flow of the CUDA

2　FUSAR-Ku 实时处理机

无人机载 SAR 实时成像系统难以在保证其成像质量的同时保证实时性，使得雷达成像的实时性依赖于硬件的性能。为充分发挥 GPU 硬件性能，本文根据异步处理时序的特点，在 FUSAR-Ku 系统的基础上，设计并优化了 FUSAR-Ku 实时处理机。FUSAR-Ku 实时处理机如图 3 所示。

2.1　实时处理机硬件框架

FUSAR-Ku 实时处理机硬件内部结构和数据处理流程如图 3 所示。DSP 表示数字信号处理器（Digital Signal Processor）；DDS 表示直接数字式频率合成器（Direct Digital Synthesizer）；SSD 表示固态硬盘（Solid State Disk）；SMA 表示 SMA 接口（SubMiniature Version A）；V 表示垂直方向极化（Vertical）；H 表示水平方向极化（Horizontal）；DA 表示数字信号转模拟信号（Digital-to-Analog）；AD 表示模拟信号转数字信号（Analog-to-Digital）；RF 表示射频（Radio Frequency）；RTK 表示实时动态差分（Real-Time Kinematic）。MiniSAR 数据采集系统主要由 FUSAR-Ku 系统、惯性导航系统和地面全球导航卫星系统（Global Navigation Satellite System，GNSS）基站 3 部分构成，其中 FUSAR-Ku MiniSAR 参数见表 1。

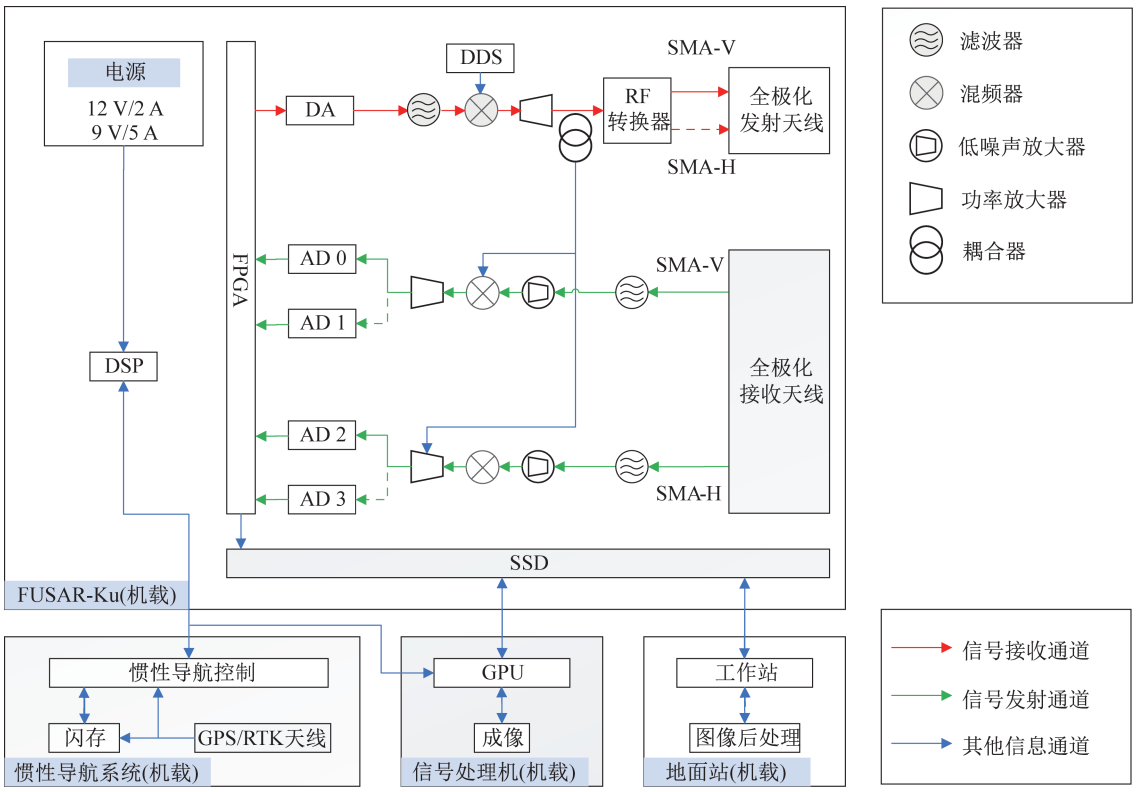

图 3　FUSAR-Ku 实时处理机

Fig. 3　FUSAR-Ku real-time processor

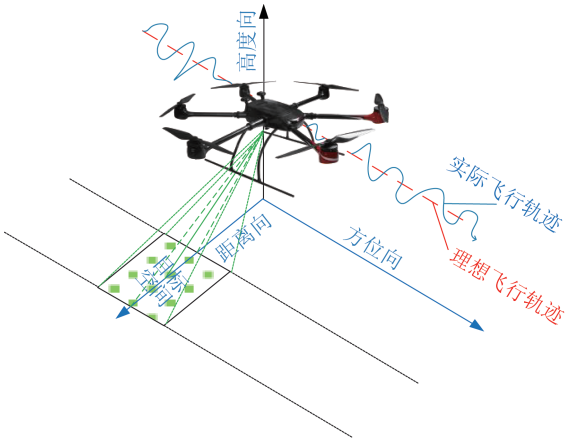

图 4　条带式 SAR 成像模式

Fig. 4　Imaging mode of the stripmap SAR

无人机载 MiniSAR 数据采集场景如图 4 所示，它的原始回波数据往往接近 TB 级别，难以被实时地传输至地面站，这也是该系统的受限之处。在 MiniSAR 数据采集系统上增加信号处理机，即可完成 FUSAR-Ku 实时处理机的硬件设计。FUSAR-Ku 实时处理机可将 TB 级别的雷达原始回波数据转化为 MB 级别的 SAR 成像数据，大大降低了地面站实时收取的难度。成像数据被传输到地面站后，

表 1　FUSAR-Ku MiniSAR 参数

Tab. 1　Parameters of the USAR-Ku MiniSAR

参数	参数值
中心/载波频率/GHz	15.2
带宽/GHz	12.0
方位向波束宽度/(°)	6
距离向波束宽度/(°)	20
下视角/(°)	0/45/50/60/65/70
脉冲持续时间/μs	5
脉冲重复频率/Hz	2 000
方位向采样率/MHz	25
平台相对高度/m	100/200/300

设备不易受到质量和尺寸的限制，算力问题得以解决，各种后处理得以实时实现。

无人机载重上限大约 7 kg，硬件选择应优先考虑体积小、质量小、计算性能好、功耗低的设备。英伟达 Jetson 系列边缘计算机符合体积和质量小的要求。Jetson AGX Xavier 如图 5 所示，其功耗范围为 10～30 W，相比于 Jetson TX2 其计算性能更好，相比于 Jetson AGX Orin 其功耗更低，因此被用作为实时处理机核心计算设备。数据流时序的设计需

要对 Jetson AGX Xavier 的计算能力和算法复杂度进行评估。

图 5　Jetson AGX Xavier[26]（图片来源：NVIDIA）

Fig. 5　Jetson AGX Xavier[26] (Image source: NVIDIA)

Jetson AGX Xavier 官方参数见表 2。硬件的测试结果见表 3。由表 2 可知，乘、加、减、乘加融合和赋值操作所消耗的时间相近，除法运算所需时间大约为乘法运算的 5 倍。乘、加、减和乘加融合皆为一次浮点运算。SAI 算法复杂度见表 4。

表 2　Jetson AGX Xavier 硬件参数

Tab. 2　Parameters of the Jetson AGX Xavier

设备	最大核心频率/GHz	核心数/个
GPU	1.21	512
CPU	2.03	8

表 3　Jetson AGX Xavier 各类单操作平均所需时间

Tab. 3　Average time required for each operation of the Jetson AGX Xavier

计算类型	双精度/(秒·次$^{-1}$)	单精度/(秒·次$^{-1}$)
乘、加、减、加乘融合	$8.157\,4\times10^{-11}$	$9.417\,7\times10^{-13}$
除法	$4.647\,9\times10^{-10}$	$4.243\,8\times10^{-12}$
赋值	$9.765\,6\times10^{-11}$	—

表 4　SAI 各计算操作次数

Tab. 4　Calculation operations of the SAI

计算类型	所需操作数
乘、加、减、加乘融合	$3N_aN_r\log N_a+75N_aN_r\log N_r+38N_aN_r+23N_a+3N_r$
除法	$34N_aN_r\log N_r+3N_aN_r+3N_a+2N_r$
赋值	$9N_aN_r+5N_a+3N_r$

由表 4 中可见，浮点运算复杂度可用 AN_a、N_r 表示，A 表示浮点运算复杂度的系数，其数值约为 1 170。

2.2　实时处理机软件框架

实时单 CPU 线程数据流如图 6 所示。图中流程分为 3 个部分：1）数据准备；2）数据预处理；3）主干计算。

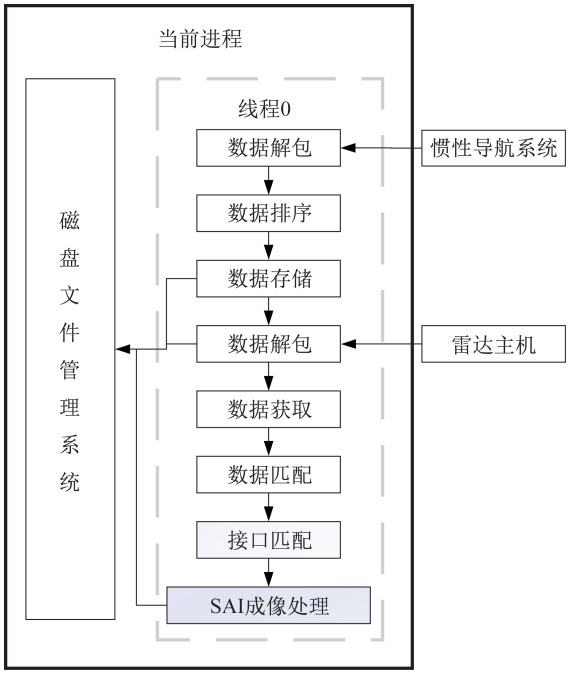

图 6　实时成像单 CPU 线程数据流

Fig. 6　Data streams of the real-time imaging on a single CPU

数据准备包括惯性导航系统和雷达主机的数据解包、排序以及存储，该部分由 CPU 负责；数据预处理包括数据和接口的匹配，该部分由 CPU 负责；主干计算是指 SAI 成像处理，该部分由 GPU 负责。

在数据预处理之前，CPU 需要耗费较多时间用于文件数据读取和存储，而此时 GPU 处于闲置状态，CPU 与 GPU 的并行工作效率几乎为零。为降低 CPU 数据准备和数据预处理开销，充分提高两者并行效率，可根据 CPU 与 GPU 异步处理流程，提出一种多 CPU 线程的异步优化方案。

多 CPU 线程实时成像数据流如图 7 所示。当前进程分 3 个线程并发执行，采用时间片轮转调度策略。线程 0 查询磁盘文件中是否存在与雷达数据相匹配的惯导数据；线程 2 通过 PCIe 串口解包雷达主机数据；线程 1 完成数据预处理，并呼叫 GPU 进行成像。为提高系统效率，雷达数据解包后会被暂时存放于内存中的暂存区，暂存区内数据在成像完成后被销毁；为维持内存使用稳定，解包后的惯导

数据不停留于内存中,将立即以二进制文件的形式保存于磁盘中以等待数据匹配,数据匹配使用二分查找策略。为保证线程安全,互斥锁被用于临界区数据同步。

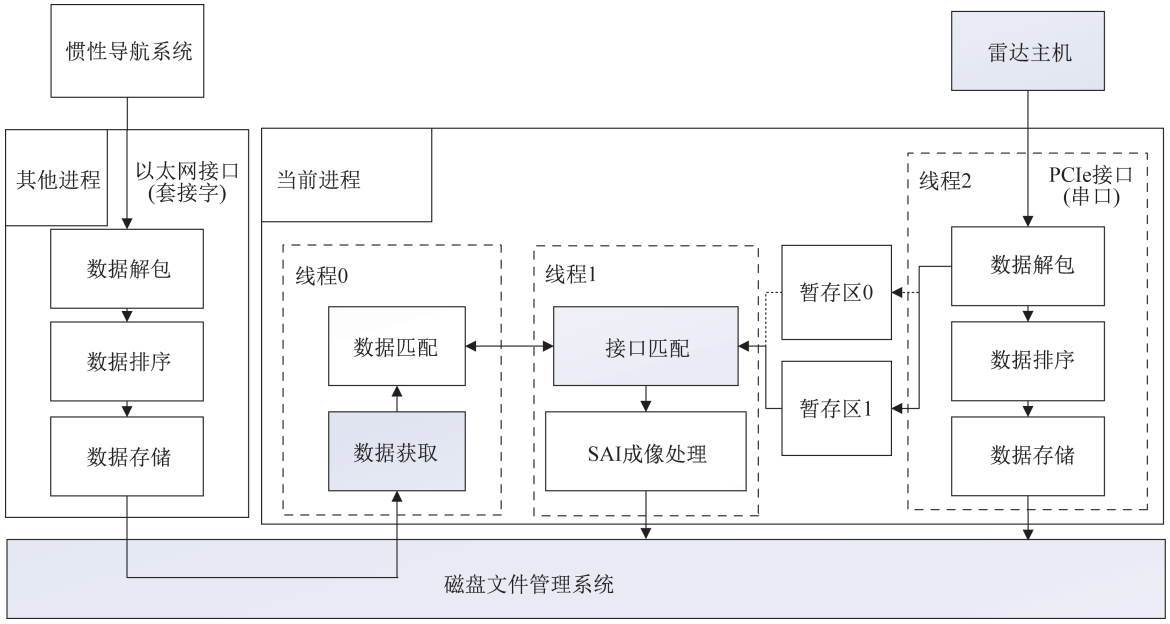

图 7　实时成像多 CPU 线程数据流

Fig. 7　Data streams of real-time imaging on multiple CPUs

该系统参考多流处理的加速思想,并设置了多个暂存区,当其中一个暂存区内的雷达原始数据被用于成像时,另一个暂存区会加载下一帧所需的雷达原始数据,可掩盖原始数据从磁盘到内存的拷贝延时。惯导系统无法保证时钟与雷达主机完全同步,其数据不能存放于暂存区,只能由其他进程通过以太网口收取数据并存放于磁盘中。

软件所需内存大约为原始数据的 5 倍。Jetson 设备中的 GPU 与 CPU 共用一块 16 GB 的内存,其中约 2 GB 用于保证软件以外的环境运行,因此软件在一个处理周期中最多能处理 2.5～3.0 GB 的原始数据。

成像时序会产生一帧图像的时延,如图 8 所示。软件在成像启动前会开辟好输入数据、计算中间变量和结果所有所需的内存空间,并在程序终止时进行统一销毁。因此,在“主干计算”的循环阶段,软件不会有任何开辟内存的行为,同时避免了内存泄漏问题。

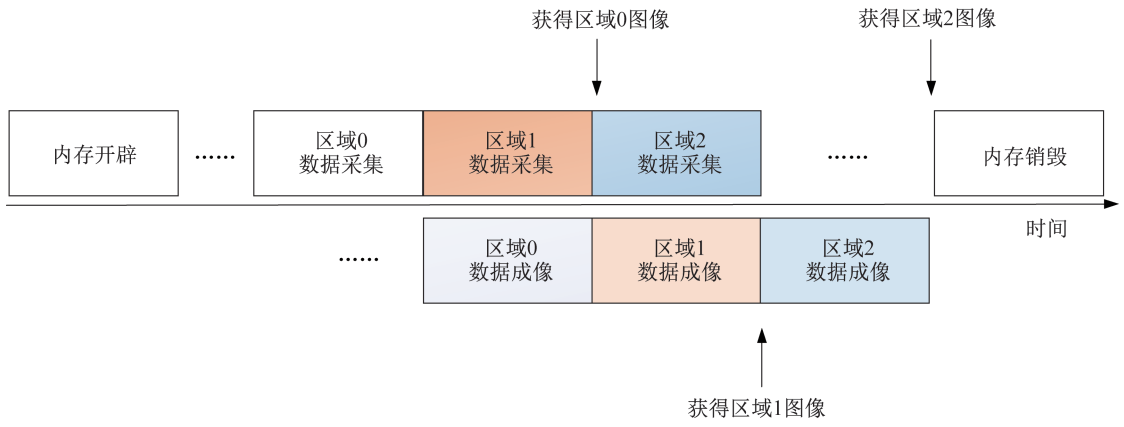

图 8　SAR 成像时序

Fig. 8　Time sequence of SAR imaging

2.3　实验评估指标

为直观展现出多 CPU 线程优化在无人机载 MiniSAR 实时成像中的优势，本实验设计了几种实验评估指标。实验评估指标有 SAR 单帧加速比 α_{SAR}、多 CPU 线程压缩率 μ_{mcg}、多 CPU 线程开销比例 γ_{ref}、异步计算并行率 ε_{het}。μ_{mcg} 可以显示出，多 CPU 线程优化后实时处理机成像时间缩短的时间比例；γ_{ref} 可以显示出，多 CPU 线程切换开销占据成像周期的比例；ε_{het} 可以显示出，CPU 与 GPU 同时工作的时间占据成像周期的比例。

各评估指标如式(1)～式(4)所示。

$$\alpha_{SAR}=\frac{T_{asc}}{T_{acg}} \tag{1}$$

式中：T_{acg} 为实时处理机完成一帧 SAR 成像平均用时；T_{asc} 为单核心 CPU 系统完成一帧 SAR 成像平均用时。

$$\mu_{mcg}=\frac{T_{ascg}}{T_{amcg}}-1 \tag{2}$$

式中：T_{ascg} 为单 CPU 线程实时处理机完成一帧 SAR 成像平均用时；T_{amcg} 为多 CPU 线程实时处理机完成一帧 SAR 成像平均用时。

$$\gamma_{ref}=\frac{T_{await}}{T_{amcg}} \tag{3}$$

式中：T_{await} 为图 7 中线程 1 完成一帧 SAR 成像等待互斥锁平均用时。

$$\varepsilon_{het}=\frac{T_{ah}}{T_{amcg}} \tag{4}$$

式中：T_{ah} 为 GPU 和 CPU 在多 CPU 线程实时处理机中共同主干计算所花总时间。

2.4　实时成像结果及分析

成像结果如图 9 所示，该伪彩图由多通道(HH、VH、HV、VV)SAR 成像结果经过极化定标和极化分解得到，成像区域为复旦大学邯郸校区，图中右侧为校内标志性建筑物光华楼。红色部分表示偶次散射，该散射分量主要由建筑物与地面构成的二面角贡献；绿色部分表示多次散射，主要由地表植被、树林等目标贡献。距离向分辨率为 0.08 m，方位向分辨率为 0.06 m，行高 8 199 个像素(655.92 m)，行宽 10 047 个像素(602.82 m)。

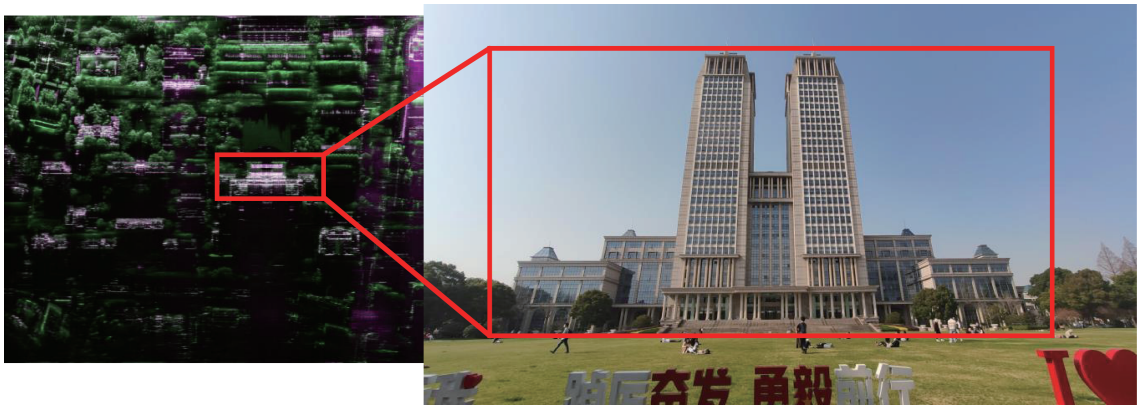

图 9　成像结果

Fig. 9　Imaging result

与 CPU 计算相比较，GPU 成像双精度平均误差约为 −70 dB，单精度平均误差约为 −30 dB。单精度计算效率高，但大大降低了成像质量。因此，实验均使用双精度进行成像处理。

本研究设计了多组实验对比，以下均为离线双精度成像的实验结果。考虑到 GPU 热启动计算效率要比冷启动高，实验只取 5 帧 SAR 成像后的结果。多 CPU 线程实验还考虑了线程块大小、数据量大小的影响。实验结果见表 5 与表 6。为使线程利用率为 100%，表 5 中线程块的线程数被设置为 32 的整数倍，1024 线程/块为 Jetson AGX Xavier 的软件上限。

在表 5 中：所有实验中数据 N_a 为 12 500，N_r 为 10 000；μ_{mcg}、γ_{ref}、ε_{het} 是多 CPU 线程实验组与单 CPU 线程实验组对照后得到的评估指标，不用于单 CPU 线程实验组性能评估；各组实验在编译期间并未分配任何本地内存；单 CPU 线程与多 CPU 线程实验在惯导数据存取、雷达原始数据存取、雷达成像阶

表5 不同线程块实验对比

Tab. 5 Experimental comparison of different thread block sizes

工作模式	成像周期/s	α_{SAR}	μ_{mcg}/%	γ_{ref}/%	ε_{het}/%
多CPU线程(32线程/块)	21.997 7	12.741 7	13.46	<0.01	22.90
单CPU线程(32线程/块)	24.958 6	11.230 1	—	—	—
多CPU线程(64线程/块)	23.953 0	12.191 7	12.32	<0.01	21.03
单CPU线程(64线程/块)	26.904 5	10.854 2	—	—	—
多CPU线程(128线程/块)	23.941 5	12.196 0	8.58	<0.01	24.91
单CPU线程(128线程/块)	25.994 9	11.232 6	—	—	—
多CPU线程(256线程/块)	23.043 5	12.198 0	12.83	<0.01	21.86
单CPU线程(256线程/块)	26.001 0	10.810 5	—	—	—
多CPU线程(512线程/块)	23.048 6	12.203 5	12.81	<0.01	25.73
单CPU线程(512线程/块)	26.000 3	10.818 1	—	—	—
多CPU线程(1024线程/块)	23.049 6	12.196 3	12.81	<0.01	21.78
单CPU线程(1024线程/块)	26.002 6	10.811 2	—	—	—

段都有16.67%的概率出现1 s左右的波动,可能与各线程的波动相关,但不会在单线程上累加,成像周期也只存在1 s左右的波动。

在表6中:所有实验中线程块大小为32线程/块;实验中各处理阶段会1 s左右的波动。

根据硬件参数,假设CPU只使用单核心处理,可得到α_{SAR}的上限为305.182 3,其式如下:

$$\alpha_{max}=\frac{F_g\times N_g}{F_c\times N_c} \tag{5}$$

式中:F_g为GPU最大核心频率;N_g为GPU核心数量;F_c为CPU最大核心频率;N_c为CPU核心数量。

如图9所示,成像结果虽存在一定的运动模糊,但建筑物、道路、植被的轮廓依然清晰,符合成像要求。GPU双精度计算得到的成像结果误差为-70 dB。可见,其成像质量与CPU双精度计算基本无异。因此,GPU代替CPU承担SAR成像计算是完全可靠的。

多CPU线程实时处理机线程块大小的对比实验结果见表5。从μ_{mcg}和ε_{het}的实验结果中可以看

表6 不同数据量实验对比

Tab. 6 Experimental comparison of different amounts of data

工作模式	成像周期/s	α_{SAR}	μ_{mcg}/%	γ_{ref}/%	ε_{het}/%
多CPU线程(12 500×10 000)	22.019 3	13.298 2	13.45	<0.01	23.03
单CPU线程(12 500×10 000)	24.980 2	11.722 0	—	—	—
多CPU线程(6 250×10 000)	8.027 3	14.233 9	12.91	<0.01	21.73
单CPU线程(6 250×10 000)	9.063 4	12.606 7	—	—	—
多CPU线程(3 125×10 000)	3.039 4	22.280 1	2.24	<0.01	24.55
单CPU线程(3 125×10 000)	3.107 5	21.791 6	—	—	—
多CPU线程(12 500×5 000)	11.049 6	11.946 4	17.53	<0.01	9.35
单CPU线程(12 500×5 000)	12.986 8	10.164 4	—	—	—
多CPU线程(6 250×5 000)	4.018 7	16.649 5	24.12	<0.01	24.18
单CPU线程(6 250×5 000)	4.988 0	13.413 8	—	—	—
多CPU线程(3 125×5000)	1.044 0	33.345 9	6.91	<0.01	1.73
单CPU线程(3 125×5 000)	1.116 2	31.190 0	—	—	—
多CPU线程(12 500×2 500)	5.037 1	11.822 7	8.97	<0.01	1.31
单CPU线程(12 500×2 500)	5.488 8	10.849 8	—	—	—
多CPU线程(6 250×2 500)	1.961 9	15.683 3	1.83	<0.01	1.81
单CPU线程(6 250×2 500)	1.9978	15.4018	—	—	—
多CPU线程(3 125×2 500)	0.9785	16.7976	3.61	<0.01	1.77
单CPU线程(3 125×2 500)	1.0138	16.2120	—	—	—

到,多CPU线程优化可在GPU并行加速的基础上掩盖磁盘文件的存取开销,提高CPU与GPU之间的异步并行效率,将处理机整体性能提升了15%。纵观不同线程块实验,各组实验未使用本地内存,且线程块大小为32线程/块时,成像效率最高;线程块大小大于256线程/块时,成像效率也会有略微提升。线程块为32线程/块时,成像算法Warp活跃度较高,虽然块中线程少,但是Warp切换次数较少,单线程占有的资源较多,所以成像效率更高;线程

块增大到256线程/块时，任务调度的开销减轻，所以成像效率会有略微提升。因为存取缓存机制不稳定，各处理阶段存在1 s的波动。在运算资源方面，实验表明，系统在连续处理100帧成像数据时，并未发生实时内存占用增大的现象，表明该系统有效解决了内存泄漏问题。

表6展示了多CPU线程实时处理机不同数据量的对比实验结果。分析可知，处理机在处理小数据量时 α_{SAR} 最高可达到33.345 9，符合式(5)加速比最大理论。随着 N_r 数值的增加，α_{SAR} 会呈现下降趋势，数据量增大时，下降速度会减慢，而 N_a 数值增加时并不会出现这种状况。α_{SAR} 随 N_r 减小是因为插值中的二分查找和累加使 N_r 方向的数据局部性降低，且处理逻辑分支增多；数据量越大，下降速度越慢，这是因为小数据量的并行计算成分较低，计算效率主要受数据准备和数据预处理的影响。在处理机中，成像开销最大，其次是雷达原始数据的存取，多CPU线程实时处理机的效率提升主要得益于雷达原始数据存取的隐藏。如果多CPU线程实时处理机的工作效率与单CPU线程相同，ε_{het} 的值将与 μ_{mcg} 相近。实验结果显示，小数据量受到1 s波动的影响，不易看出 ε_{het} 与 μ_{mcg} 之间的关系，但当数据为 N_a 为12 500，N_r 为10 000时，ε_{het} 大约为 μ_{mcg} 的2倍，多CPU实验中雷达存取开销要比单CPU线程实验大整整一倍。说明各线程竞争CPU计算资源，工作分时段进行，符合时间片轮转调度策略特点，进一步证明了多CPU线程实时处理机未使用CPU多物理核心进行计算。

3　结束语

本文提出了一种FUSAR-Ku实时处理机的多CPU线程优化方案。该方案解决了实时成像中的内存泄漏问题，确保了成像稳定性。在实时处理机中，GPU的成像效率是CPU成像效率的12倍左右，使用该方案后，成像效率还可以继续提升15%。由此可见，GPU异构系统相比于单核CPU系统具有更大的优势。在此基础上，实时处理机的多CPU线程工作模式相较于单CPU线程工作模式也有着明显优势。本文还详细介绍了该项工作的硬件框架和软件框架。硬件部分介绍了核心雷达主机的内部结构，测试分析了核心计算设备NVIDIA Jetson AGX Xavier的性能；软件部分分析了系统数据流，提供了一种提高CPU与GPU并行效率的方案。为后续实时处理机的开发奠定了基础，GPU与SAR相结合的前景依旧广阔。由GPU理论最大加速比可知，处理机的加速性能还有着一定的提升空间。未来的工作不仅会着重于实时处理机系统的开发，也会在GPU算法层面继续研究。

参考文献

[1] 保铮，邢孟道，王彤.雷达成像技术[M].北京：电子工业出版社，2005.

[2] FU C H, LI B, DING F Q, et al. Correlation filters for unmanned aerial vehicle-based aerial tracking: a review and experimental evaluation [J]. IEEE Geoscience and Remote Sensing Magazine, 2021, 10 (1): 125-160.

[3] OSCO L P, MARCATO J, RAMOS A P M, et al. A review on deep learning in UAV remote sensing [J]. International Journal of Applied Earth Observation and Geoinformation, 2021, 102: 102456.

[4] XIAO Z Y, ZHU L P, LIU Y M, et al. A survey on millimeter-wave beamforming enabled UAV communications and networking [J]. IEEE Communications Surveys & Tutorials, 2021, 24 (1): 557-610.

[5] GISDER T, HARRER F, BIEBL E. Application of a stream-based SAR-backprojection approach for automotive environment perception [C]// 2018 19th International Radar Symposium (IRS). 2018: 1-10.

[6] CORRADI F, FIORANELLI F. Radar perception for autonomous unmanned aerial vehicles: a survey [J]. System Engineering for constrained embedded systems, 2022: 14-20.

[7] SHI J, MA L, ZHANG X L. Streaming BP for non-linear motion compensation SAR imaging based on GPU [J]. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 2013, 6 (4): 2035-2050.

[8] WANG Y E, WEI G Y, BROOKS D. Benchmarking TPU, GPU, and CPU platforms for deep learning [J]. arXiv Preprint arXiv, 1907.10701.

[9] YUAN X, LUO M Y, LIU L J, et al. High-performance implementation of UAV MiniSAR imaging algorithm based on GPU [C]// 2021 7th Asia-Pacific Conference on Synthetic Aperture Radar (APSAR). 2021: 1-6.

[10] ZHANG Y, ZHU D, MAO X, et al. Multirotors video

synthetic aperture radar: system development and signal processing[J]. IEEE Aerospace and Electronic Systems Magazine, 2020, 35(12): 32-43.

[11] XIAO X, ZHANG R, YANG X, et al. Realization of SAR real-time processor by FPGA [C]//IGARSS 2004. 2004 IEEE International Geoscience and Remote Sensing Symposium. 2004: 3942-3944.

[12] ASANO S, MARUYAMA T, YAMAGUCHI Y. Performance comparison of FPGA, GPU and CPU in image processing[C]// 2009 international conference on field programmable logic and applications. 2009: 126-131.

[13] ZHANG Z, LIU P, WANG W, et al. High-performance password recovery hardware going from GPU to hybrid CPU-FPGA platform [J]. IEEE Consumer Electronics Magazine, 2020, 11(1): 80-87.

[14] HAWKINS B P, TUNG W. UAVSAR Real-time embedded GPU processor[C]// IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium. 2019: 545-547.

[15] OWENS J D, HOUSTON M, LUEBKE D, et al. GPU computing[J]. Proceedings of the IEEE, 2008, 96(5): 879-899.

[16] MOORE G E. Progress in digital integrated electronics [C]// Electron devices meeting,1975, 21: 11-13.

[17] 熊庭刚.GPU的发展历程,未来趋势及研制实践[J].微纳电子与智能制造,2020,2(2):36-40.

[18] CAFFORIO C, PRATI C, ROCCA F. SAR data focusing using seismic migration techniques[J]. IEEE Transactions on Aerospace and Electronic Systems, 1991, 27(2): 194-207.

[19] SPUDIS P, BUSSEY D, BALOGA S, et al. Initial results for the north pole of the moon from Mini-SAR, Chandrayaan-1 mission [J]. Geophysical Research Letters, 2010, 37(6):6204.

[20] YAN J, GUO J, LU Q, et al. X-band mini SAR radar on eight-rotor mini-UAV [C]// 2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS). 2016: 6702-6705.

[21] 冯利鹏,王辉,郑世超,等.FMCW体制毫米波SAR实时信号处理方法研究[J].上海航天(中英文),2022,39(3):116-121.

[22] 孔令振,王辉,郑世超,等.一种Ka波段多通道FMCWSAR数字接收机设计与实现[J].上海航天(中英文),2022,39(6):1-13.

[23] LUOMEI Y, XU F. Motion compensation for multirotors minisar system [C]// 2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS. 2021: 5143-5146.

[24] LUOMEI Y X, XU F. Segmental aperture imaging algorithm for multi-rotor UAV-borne MiniSAR [J]. IEEE Transactions on Geoscience and Remote Sensing, 2023, 61: 1-18.

[25] LUOMEI Y X, XU F, WANG F, et al. Demonstration of simultaneous localization and imaging with multirotor-borne MiniSAR[J]. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 2022, 15: 6548-6558.

[26] NVIDIA Corporation, Jetson AGX Xavier Developer Kit[EB/OL]. [2023-03-23]. https://developer.nvidia.com/embedded/jetson-agx-xavier-developer-kit.