

Jetson TK1 平台实现快速红外图像背景预测算法

吴 鑫, 张建奇, 杨 琛

(西安电子科技大学 物理与光电工程学院, 陕西 西安 710071)

摘 要: 红外弱小目标的探测与跟踪对运算硬件和算法的性能提出较高的要求。针对传统背景预测算法串行运算耗时较长的问题, 以及经典的通用 GPU(Graphic Processing Unit)体积与功耗过大难于整合到红外设备中的问题, 提出在嵌入式 GPU 平台 NVIDIA Jetson TK1 中实现并行分离卷积的方法, 利用 CUDA(Compute Unified Device Architecture)实时执行背景预测算法, 实现了在嵌入式 GPU 平台上高效的红外背景预测算法。实验结果表明, 在保证正确预测背景的前提下, 利用小体积、低功耗的嵌入式 GPU 平台可以将运算性能提高到串行运算的 15 倍以上。

关键词: 红外探测; 快速运算; 背景预测; Jetson TK1; CUDA

中图分类号: TP391.4 **文献标志码:** A **文章编号:** 1007-2276(2015)09-2615-07

Efficient infrared image background prediction with Jetson TK1

Wu Xin, Zhang Jianqi, Yang Chen

(School of Physics and Optoelectronic Engineering, Xidian University, Xi'an 710071, China)

Abstract: For infrared small target detection and tracking, it requires very high efficiency of both hardware and algorithm. Since the classic background prediction algorithm is a serial one, which is very time consuming. Considering that common GPUs(Graphic Processing Units) are big in size and too power consuming to be integrated into an infrared device, the implement background prediction algorithm was proposed with separable convolution template method on the embedded GPU platform, named NVIDIA Jetson TK1. Taking advantage of CUDA(Compute Unified Device Architecture) programming language to execute background prediction algorithm in parallel, an operable and high performance result on board was achieved, which gained a 15x speedup comparing to the serial way with a CPU.

Key words: infrared detection; high performance computation; background prediction; Jetson TK1; CUDA

收稿日期: 2015-01-11; 修订日期: 2015-02-14

基金项目: 国家自然科学基金(61301290)

作者简介: 吴鑫(1986-), 男, 博士生, 主要从事 GPU 高性能运算与仿真方面的研究。Email: plwuxin@163.com

导师简介: 张建奇(1960-), 男, 教授, 博士生导师, 博士, 主要从事光电成像系统方面的研究。

0 引言

弱小目标的探测与跟踪是红外技术的重要应用之一,红外图像的背景预测是实现红外弱小目标检测与跟踪的经典算法之一^[1]。红外图像背景预测针对单幅或连续红外序列图像进行模板卷积运算,达到抑制背景噪声并突出目标信号的目的。近年来,背景预测的算法得到了很多扩展和应用,如 K-最邻近决策法^[2]、梯度信息融合法^[3]、多尺度面元法^[4]、双边滤波法^[5]等,有效地降低或消除了背景噪声对目标的影响,为红外目标的探测和跟踪奠定了基础。由于红外图像的背景预测涉及到全局模板卷积算法,所以运算的负载较大,一般比较耗时。针对背景预测算法性能低下的问题,基于 GPU(Graphic Processing Unit)的并行实现是一种有效的解决方案^[6]。在 GPU 中采用分离式卷积模板可以大幅提高背景预测的运算性能,尤其当背景预测模板的尺寸较大时,基于 GPU 的并行红外图像背景预测可以比传统的串行算法节省百倍以上的时间。然而,传统的 GPU 运算平台都必须依托于计算机系统级的资源与功耗管理,具有较大的体积和过大的功耗。因此,传统的 GPU 平台无法灵活地应用于轻便型红外设备中,如导引装置、机载测量装置等。从而,利用 FPGA(Field-Programmable Gate Array)将背景预测算法移植到小型电路中的方法被提出^[7],但是基于 FPGA 的实现需要较高的开发成本和编程能力,使其不易实现红外设备一体化的整合。因此,如何高效便利地实现红外图像背景预测方法成为红外弱小目标探测与跟踪技术在实际应用中的主要瓶颈之一。最近,继 GPU 的高性能并行运算在天文、遥感、通信和多媒体等诸多领域获得显著发展之后,NVIDIA 公司于年初推出了下一代嵌入式 GPU 微型电脑:Jetson TK1^[8]。NVIDIA Jetson TK1 的推出正式开启了嵌入式 GPU 时代的新纪元,Jetson TK1 上搭载了四核 Cortex-A15 架构的 ARM 处理器和具有 192 个 NVIDIA KeplerTM 架构运算核心的 Tegra K1 GPU,并且 Tegra K1 可以由传统 NVIDIA GPU 的编程语言 CUDA (Compute Unified Device Architecture)驱动^[9]。这使得 Tegra K1 成为目前最先进的移动处理器的同时又具备了良好的开发环境支持。所以,嵌入式 GPU 必将引发计算机视觉、机器

人控制、信号处理及自动化等领域的新一轮技术革命。利用嵌入式 GPU 进行高效红外图像处理的应用国内外尚未见报道。鉴于此,文中提出一种新的嵌入式 GPU 架构下的红外图像背景预测方法,利用模板分离卷积的一般形式对红外图像进行快速背景预测,并通过实测和仿真红外图像实验验证其可行性与高效性。

1 分离式卷积背景预测原理

根据热力学第一定律,天空和海洋等自然背景中的能量一般呈连续分布,从红外图像上来看往往是相对平滑的。但是,人造目标如飞机、舰船等与背景的能量交换较少,尤其是移动目标无法在短时间里与背景进行充分的能量交换,所以在红外图像中此类目标会表现出突出的能量起伏。背景预测的方法利用目标和背景能量分布的特点,使用周围背景点预测模板中心点的能量灰度,从而,目标所在的区域会产生较突出的预测误差。最终将预测图像与原图像相减得到残差图像,达到抑制背景、突出目标的目的。

红外图像将目标与背景信号建立如下模型^[1]:

$$y(i, j) = x_b(i, j) + x_t(i, j) + n(i, j) \quad (1)$$

式中: $y(i, j)$ 为图像上任一像素点的辐射能量; $x_b(i, j)$ 、 $x_t(i, j)$ 和 $n(i, j)$ 分别为背景的辐射能量、目标的辐射能量与随机噪声的能量。

最基本的背景预测模型为:

$$\hat{y}(k) = \sum_{l \in \Omega} w(l) y(k-l) \quad (2)$$

式中: $\hat{y}(k)$ 为预测后的信号空间; $w(l)$ 为预测权重系数,组成了背景预测的模板矩阵; Ω 为整个图像空间范围; $y(k-l)$ 为原始红外图像信号空间。不同形式的 $w(l)$ 和不同的 $w(l)$ 计算方法形成了不同的背景预测方法,模板分离卷积的方法假设 $w(l)$ 是一个满足高斯分布的空域模板,为了清楚地表达高斯分布的模板分离方法,将公式(2)写成二维空间的形式:

$$\hat{y}(i, j) = \sum_{n=1}^{H_0} \sum_{m=1}^{W_0} \varpi(m, n) \cdot y(i-m, j-n), \quad i \in [1, w], j \in [1, h] \quad (3)$$

式中: H_0 和 W_0 为卷积模板 $\varpi(m, n)$ 的空间尺寸; m 和 n 为卷积模板中的二维索引; i 和 j 分别为红外图像宽 w 和高 h 的索引。 $\hat{y}(i, j)$ 代表被估计像素值, $y(i-m,$

$j-n$)表示在卷积模板覆盖范围下的原红外图像的像素值。经典的背景预测模板都是高斯的或者类高斯的分布,如公式(4)和(5):

$$W_1 = \begin{bmatrix} 0.0158 & 0.0167 & 0.0158 \\ 0.0167 & 0.0177 & 0.0167 \\ 0.0158 & 0.0167 & 0.0158 \end{bmatrix} = \begin{bmatrix} 0.1258 \\ 0.1330 \\ 0.1258 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \quad (4)$$

$$W_2 = \frac{1}{288} \begin{bmatrix} 9 & 9 & 9 & 9 & 9 & 9 & 9 \\ 9 & 4 & 4 & 4 & 4 & 4 & 9 \\ 9 & 4 & 1 & 1 & 1 & 4 & 9 \\ 9 & 4 & 1 & 0 & 1 & 4 & 9 \\ 9 & 4 & 1 & 1 & 1 & 4 & 9 \\ 9 & 4 & 4 & 4 & 4 & 4 & 9 \\ 9 & 9 & 9 & 9 & 9 & 9 & 9 \end{bmatrix} \quad (5)$$

高斯类卷积模板如 W_1 可以根据高斯分布的二维形式分离成行向量和列向量,详细数学推导过程见参考文献[6],类高斯分布的卷积模板如 W_2 可以通过矩阵的线性叠加原理分解成可分离子矩阵的和。通用分解公式如下:

$$\begin{bmatrix} a_n & \cdot & \cdot & \cdot & \cdot & a_n \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & a_1 & a_1 & a_1 & \cdot & \cdot \\ \cdot & a_1 & a_1 & a_1 & \cdot & \cdot \\ \cdot & a_1 & a_1 & a_1 & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ a_n & \cdot & \cdot & \cdot & \cdot & a_n \end{bmatrix} = C_n a_n \begin{bmatrix} 1 & \cdot & \cdot & \cdot & \cdot & 1 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & 1 & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 1 & \cdot & \cdot & \cdot & \cdot & 1 \end{bmatrix} +$$

$$C_{n-1} a_{n-1} \begin{bmatrix} 0 & \cdot & \cdot & \cdot & \cdot & 0 \\ \cdot & 1 & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & 1 & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & 1 & \cdot & \cdot & \cdot & \cdot \\ 0 & \cdot & \cdot & \cdot & \cdot & 0 \end{bmatrix} + \dots +$$

$$C_0 a_0 \begin{bmatrix} 0 & \cdot & \cdot & \cdot & \cdot & 0 \\ \cdot & 0 & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & 1 & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & 0 & \cdot & \cdot & \cdot & \cdot \\ 0 & \cdot & \cdot & \cdot & \cdot & 0 \end{bmatrix} = C_n a_n \begin{bmatrix} 1 \\ 1 \\ \cdot \\ \cdot \\ \cdot \\ 1 \\ 1 \end{bmatrix}$$

$$[1 \ 1 \ \cdot \ \cdot \ \cdot \ 1 \ 1] + C_{n-1} a_{n-1} \cdot \begin{bmatrix} 0 \\ 1 \\ \cdot \\ \cdot \\ \cdot \\ 1 \\ 0 \end{bmatrix} + \dots + C_0 a_0 \cdot \begin{bmatrix} 0 \\ 0 \\ \cdot \\ 1 \\ \cdot \\ 0 \\ 0 \end{bmatrix}, C_{n-1} = (a_{n-1} - a_n) / a_{n-1}, C_n = a_n \quad (6)$$

式中: C_n 为模板卷积权重系数; a_n 为模板中同域元素。使用分离出来的行向量和列向量对红外图像进行背景预测虽然无法降低算法的复杂度 $O(w^2)$,而且还需要预先对卷积模板进行处理,但是经过分离的模板可以分别对行向量和列向量进行并行卷积,尤其适合 GPU 的多核多线程并发式高性能运算模式。

2 Jetson TK1 上实现红外背景预测

2.1 嵌入式 Tegra K1 的 CUDA 程序设计

Jetson TK1 是 NVIDIA 公司于 2014 年 3 月推出的一款微型式的超级电脑,具有与 NVIDIA 最新 KeplerTM GPU 一样的运算的架构。Jetson TK1 具备一块 Cortex A15 架构的四核 ARM 处理器,使其具有强大处理能力的同时功耗大大降低,最重要的是 Jetson TK1 上搭载了具有 192 个 CUDA 运算核心的嵌入式 GPU 可用于高性能并行运算^[10]。Jetson TK1 同时支持 NVIDIA 公司的 CUDA 架构编程语言,因而可以通过 CUDA 编程高效、便捷地完成 GPU 并行程序设计。

从图 1 可以看出, Tegra K1 具有与 NVIDIA Kepler 系列 GPU 相类似的内部结构,不同的是 Tegra K1 的二级缓存和内存较小。在使用 CUDA 对 GPU 编程中,分别将编译好的程序在主机端(Host)和设备端(Device)执行。主机端和设备端的数据交互都是通过 PCI-E 总线负责调度的。一般来讲,良好设计的 GPU 并行程序设计都会尽量让内核函数(Kernel)在设备端的执行占主导部分,而且执行的过程中不会

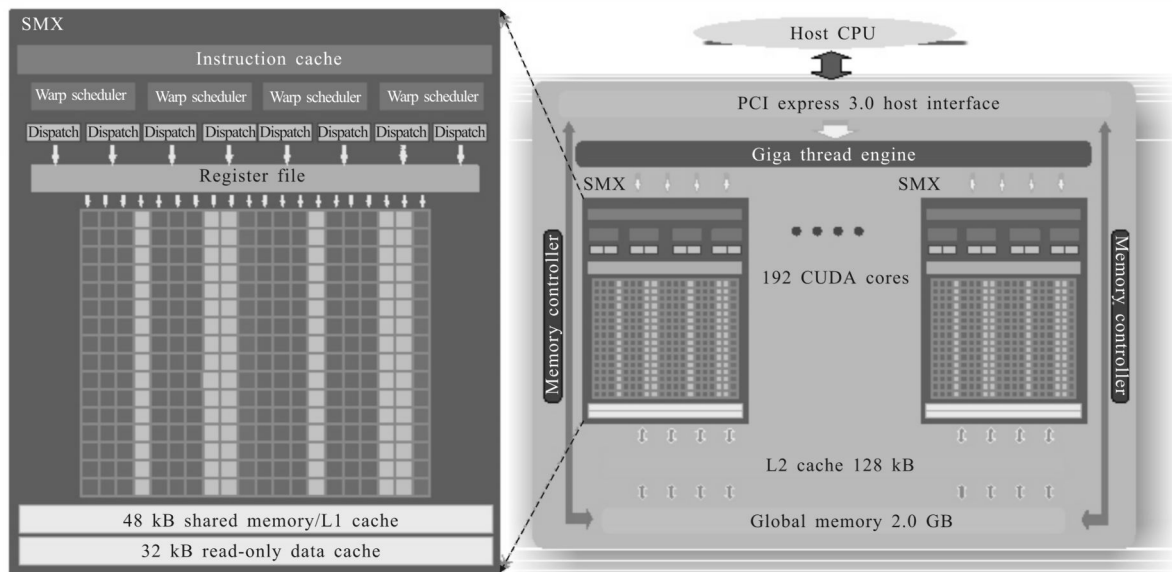


图 1 Tegra K1 GPU 内部结构示意图

Fig.1 Internal structure of Tegra K1 GPU

频繁和主机端进行数据交互，以达到减少输入输出数据传输耗时的目的。将只在程序开始和结束的时候与主机端进行数据交互的 GPU 程序设计称为纯设备程序，因为纯设备程序的并行性设计的优良程度直接决定整个并行运算性能的高低。

设备端以管理多线程的形式进行并行运算，在 GPU 内部可以并发成千上万个线程，在 Tegra K1 内部可以支持 2 147 483 647 个线程，这些线程可以通过 CUDA 编程被划分到一个三层网络结构中，即网格(Grid)、线程块(Block)和线程(Thread)。网格和线程块都可以被设计成一到三维的结构，每个线程作为最小的处理单位拥有私有存储寄存器 and 局部存储器，而每个线程块可以共同访问共享存储器(Shared memory)，同一个线程块里的线程访问共享存储器与访问私有寄存器的效率是相当的，都远远快于访问全局存储器(Global memory)。因此，在程序设计时要针对不同应用进行在线程访问存储器上的性能优化。

另外，开发人员已经利用 CUDA 基于 GPU 在天文计算、分子动力学仿真、音视频编解码等众多领域做出了几十倍甚至至上百倍的性能提高^[4]，可以预计未来嵌入式的 GPU 在各个领域会有更广泛的应用。

2.2 背景预测的实现及运算效率的优化

Jetson TK1 上搭载的 GPU 是 NVIDIA KeplerTM 系列的产品，拥有所有 KeplerTM 系列 GPU 的所有

特性。文中针对其嵌入式特性重新设计红外图像背景预测算法的接口和流程，使得在 Jetson TK1 上的实现可以兼容外部串口图像数据的输入。同时，文中应用了多种 CUDA 并行程序设计的优化手段来充分利用嵌入式 GPU 的运算资源，主要的优化技术包括以下几种：其一，将可分离的卷积模板存放在 GPU 的常量存储器中，同时将红外图像数据分块载入共享存储器中，因此同一个线程块中的所有线程都可以并发的读取模板数据。实际上，线程是以每 32 个为一组(称为一个 warp)同时执行设备端的程序，全局存储器被用来读取或者存储数据时是以 16 个线程为一组(half-warp)并发完成的，因此输入的原始红外图像数据需要组织成内存对齐^[5]的方式被输入和输出，以能够最大化地使用全局存储器，并保证数据传输的耗时最小，文中使用 NVIDIA CUDA 运行库里的 cudaMallocPitch 函数来对齐数据；其二，通过分配不同的寄存器数目和一级缓存的大小来优化程序的执行性能，但是分配的方案没有固定的规律可循需要通过实验来决定，可以通过添加 cudaFuncCachePreferL1 这个 CUDA 库函数来选择是否使用更多的一级缓存，这也是基于 GPU 并行程序设计的主要优化手段之一；其三，优化设备端线程块的大小，即每个线程块中包含的线程数目，通常设定线程块的大小是 32 的倍数时，程序的执行效率较高，

因为 GPU 是以每 32 个线程一组同时执行一次命令使得以 32 的倍数个线程访问时达到最优的性能。

另外, 利用 Tegra K1 实现背景预测的过程中, 需要时刻检查并行运算与是否与传统串行运算方法的结果保持一致。按照背景预测的流程、数据的交互及处理方法将整个 Jetson KT1 上实现红外图像的背景预测过程描述如图 2 所示。

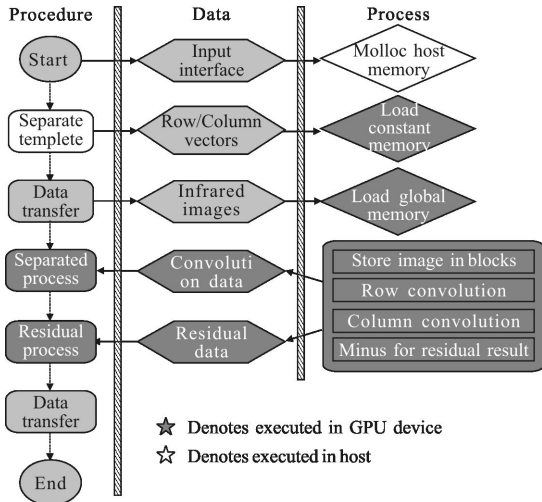


图 2 Jetson TK1 上红外图像背景预测流程图

Fig.2 Flowchart of infrared background prediction on Jetson TK1

在行卷积和列卷积的内核函数中, 两种卷积模板的尺寸索引是不同的, 列卷积是在水平方向上递增一个元素, 而行卷积是在垂直方向上递增一个行宽度的数据; 为避免发生访问共享内存的冲突而引发运算效率下降, 必须使用正确的二维数组索引, 实现细节见算法伪代码图 3(以行卷积内核函数为例)。

```

__global__ void RowsKernel(d_out, d_in, pitch){
//存放分块红外图像数据
__shared__ float s[ ][ ];
//对齐内存索引
const int x=blockIdx.x*b+threadIdx.x;
const int y=blockIdx.y*b+threadIdx.y;
d_out+=y*pitch+x;
d_in+=y*pitch+x;
//载入共享内存
for(遍历所有图像数据)
s_Data[ ][ ]=d_in[ ];
//对齐左右模板裙带空白数据
for(int i=0;i<ROWS_HALO_STEPS;i++)
s_Data[ ][ ]=x?d_in[ ];
__syncthreads();
for(遍历当前图像数据)
float sum=0
for(遍历卷积模板)
sum+=kernel*s_data[ ][ ];
d_out[ ]=sum;
}
}

```

图 3 行卷积内核函数伪代码

Fig.3 Pseudo code of row convolution in kernel function

3 实验与结论

文中使用的 Jetson TK1 设备的基本参数如下:

处理器名称 Cortex A15 MPCore™ ARMv8™

系统内存 2 GB DDR3L

存储器 16 GB eMMC

GPU 设备 NVIDIA Kepler™-based GeForce® 显

示卡

操作系统 Ubuntu 14.04 LTS

文中同时在 Jetson TK1 和 2.3GHz 的 Intel 四核 i7 处理器上测试了三组实测红外图像, 因为分享卷积的过程不涉及特殊函数(如 sqrt, pow 等), 不会影响 GPU 运算的精度, 所以实验得到的背景预测结果与基于 CPU 的串行程序完全一致, 如图 4 所示。其中 a 列为三组原始实测红外图像; b 列为 CPU 串行卷积算法的背景预测结果; c 列为 Jetson TK1 上运行分离卷积的背景预测结果; d 为预测结果残差图的三维图谱。从而, 可以证明基于 Jetson TK1 实现的并行红外图像背景预测算法的有效性。在确定并行实现的背景预测算法结果正确的基础上, 文中仿真了不同尺寸的图像数据, 对整个嵌入式 GPU 背景预测算法进行测试。同时为了测试嵌入式 GPU 的性能, 文中添加了与通用 GPU^[6]运算效率的对比。作为对比, 文中同时将传统串行背景预测算法在 Jetson TK1 的 ARM 处理器上执行时间一并列出, 如表 1 所示。

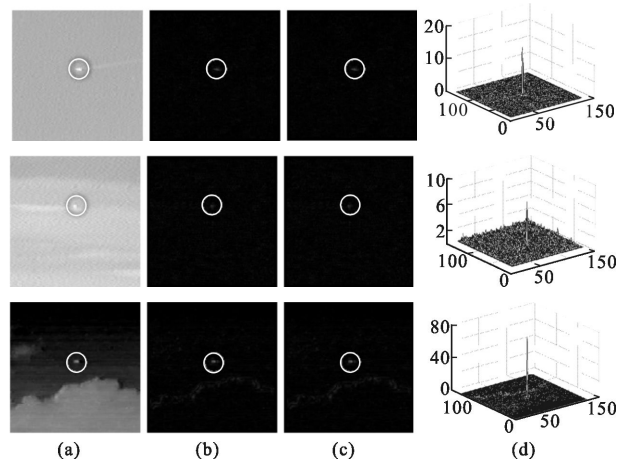


图 4 实验结果对比图

Fig.4 Experimental results for comparison

为了比较的公平性, 表 1 中笔者统计的基于 Jetson TK1 分离卷积的时间包括了原始数据的输入

过程和计算结果的输出过程耗时。实验结果表明,在嵌入式系统中使用 GPU 进行大规模的并行运算可以很大程度上提高红外背景预测算法的效率,尤其是当处理的图像数据比较大的时候效率提升效果更加明显,这就意味着随着红外光电成像器件的发展,红外图像的分辨率不断提高,利用 Jetson TK1 进行基于 GPU 的并行运算可以作为嵌入式设计的一项重要应用。同时,从表 1 中与 GeForce GTX480 通用 GPU 的运算效率对比,嵌入式 GPU 还存在一定的差距,原因在于它们之间 CUDA 内核的数量和时钟频率的差异,因此在实际应用中应当在 GPU 功耗大小及轻便程度与运算性能之间做出权衡。

表 1 红外图像背景预测算法效率对比

Tab.1 Performance comparison for different image

Image size	ARM serial convolution/ms	Jetson TK1 GPU parallel separated/ms	GeForce GTX480 parallel separated in Ref.[6]/ms
128×128	5.5	1.2	0.36
256×256	6.2	1.3	0.43
512×512	33.1	3.1	0.69
1 024×1 024	113.6	7.7	1.96
3 072×3 072	995.8	64.1	13.38

图 5 将 Jetson TK1 上的 ARM 处理与 Tegra K1 GPU 运行不同大小仿真图像时的效率进行了对比,可以看出基于 GPU 的并行计算在性能上有明显的优势,随着处理数据规模的增大时这种优势愈加明显。基于嵌入式 GPU 的并行实现与同平台的 CPU

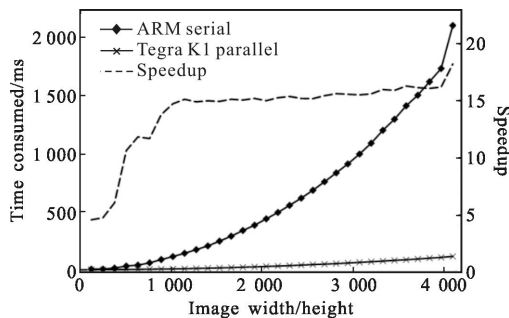


图 5 ARM 串行与 Tegra K1 并行运算性能对比

Fig.5 Performance of ARM serial and Tegra K1 parallel execution

串行相比,可以达到 15 倍以上的加速比。这样的结果使得在红外设备中整合 GPU 高性能运算能力而达到实时的处理成为可能。

一般来讲,经过优化的 GPU 程序的内核函数运算耗时应该占据大部分的总耗时间,通过统计内核函数运算时间和输入输出时间,可以判断输入输出部分是否是程序执行效率的瓶颈。图 6 表明文中实现的基于 Jetson TK1 的红外背景预测算法内核函数执行时间相对输入输出而言占据主要部分。尤其是图像尺寸大于 800×800 时内核函数执行时间占据总运行时间的 60% 以上。

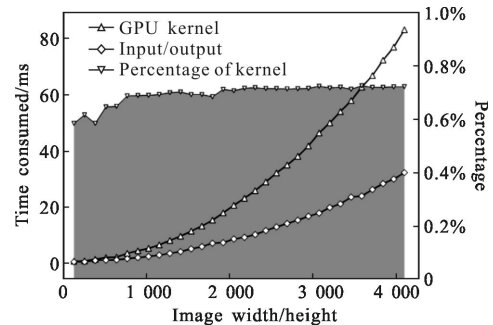


图 6 内核函数与输入输出耗时比较

Fig.6 Consuming time of input and output of the kernel function

4 结束语

文中利用分离式卷积的并行运算方法,提出在嵌入式 GPU 平台 Jetson TK1 上实现红外图像背景预测,利用 CUDA 编程实现并行的背景预测算法,达到在 Jetson TK1 上高效执行背景预测的目的。通过对比实验证明,文中提出的方法可以正确、高效的在嵌入式 GPU 平台上运行,与同平台的 ARM 处理器串行执行相比可以达到近 15 倍的加速比。随着 GPU 硬件技术的发展和 CUDA 程序设计的深入研究,以及红外器件整合度的不断提高,将 GPU 强大的高性能运算能力应用到具有实时处理能力的红外集成设备中已经成为可能,届时包括红外图像的背景预测在内的整个目标探测与跟踪过程可以更加高效、准确、实时地实现。

参考文献:

- [1] Xu Jun. Research on the detection of small and dim targets in infrared images [D]. Xi'an: Xidian University, 2003. (in

- Chinese)
- 徐军. 红外图像中弱小目标检测技术研究 [D]. 西安: 西安电子科技大学, 2003.
- [2] Chen Xiaosi, Cheng Zhengdong, Fan Xiang, et al. Infrared point target detection based on k-nearest neighbor algorithm [J]. *Infrared and Laser Engineering*, 2013, 42 (S2): 312–316. (in Chinese)
- 陈晓斯, 程正东, 樊祥, 等. 基于 k-最近邻的红外点目标检测方法[J]. 红外与激光工程, 2013, 42(S2): 312–316.
- [3] Wan Lei, Zeng Wenjing, Zhang Tiedong, et al. Real-time detection of marine infrared objects based on the fusion of gradient information [J]. *Infrared and Laser Engineering*, 2013, 42(1): 41–45. (in Chinese)
- 万磊, 曾文静, 张铁栋, 等. 基于梯度信息融合的海面红外目标实时检测[J]. 红外与激光工程, 2013, 42(1): 41–45.
- [4] Changcai Yanga, Jiayi Mab, Zhang Meifang, et al. Multiscale facet model for infrared small target detection [J]. *Infrared Physics & Technology*, 2014(67): 202–209.
- [5] Tae-Wuk Bae. Small target detection using bilateral filter and temporal cross product in infrared images [J]. *Infrared Physics & Technology*, 2011(54): 403–411.
- [6] Wu Xin, Zhang Jianqi, Huang Xi, et al. Separable convolution template (SCT) background prediction accelerated by CUDA for infrared small target detection[J]. *Infrared Physics & Technology*, 2013(60): 300–305.
- [7] Chen Yu, Yu Yan Xin, Zhao Ting, et al. The method of infrared point target detection and tracking based on DSP+FPGA [J]. *Applied Mechanics and Materials*, 2013(457): 1272–1277.
- [8] Nvidia. Bringing GPU-accelerated computing to embedded systems [EB/OL]. [2014-04]. http://developer.download.nvidia.com/embedded/jetson/TK1/docs/Jetson_platform_brief_May2014.pdf
- [9] Zhang Shu, Chu Yanli. GPU's High Performance Computing in CUDA [M]. Beijing: China Water Power Press, 2009. (in Chinese)
- 张舒, 褚艳利. GPU 高性能运算之 CUDA[M]. 北京: 中国水利水电出版社, 2009.
- [10] Nvidia. PM375 module specification [EB/OL]. [2014-05-02]. http://developer.download.nvidia.com/embedded/jetson/TK1/2014-03-24/JetsonTK1_Module_Specification_PM375_V1.01.pdf
- [11] Jason Sanders, Edward Kandrot. CUDA by Example: an Introduction to General-Purpose GPU Programming [M]. Boston: Addison-Wesley, 2010.
- [12] Shane Cook. CUDA Programming: A Developer's Guide to Parallel Computing with GPUs [M]. Waltham: Addison-Wesley, 2010.