

文章编号 1004-924X(2009)01-0225-06

航空成像像移补偿的并行计算

李 仕^{1,2}, 张 葆¹, 孙 辉¹

- (1. 中国科学院 长春光学精密机械与物理研究所, 吉林 长春 130033;
2. 中国科学院 研究生院, 北京 100039)

摘要:基于图像处理器(GPU)平台提出了维纳并行滤波算法,解决了航空成像的像移实时补偿问题。介绍了原有像移补偿算法及其 PC 运行平台的不足,立足于现有图像处理器通用计算技术对原有像移补偿算法做多线程并行计算的改进。针对图像处理器硬件并行架构,优化设置算法多线程访问,提高了各个线程的访问速度。该环节的改进甚至能将算法效率提升到原来的 3 倍。借助图像处理器的并行运算优势,单帧 1024×1024 灰度图像的恢复处理时间只需要 8 ms,整个算法的运行速度达到原先的 20 倍,能够完全满足高分辨率航拍视频图像的像移模糊实时恢复的需求。

关键词:航空成像;并行计算;维纳滤波;图像恢复;图像处理器

中图分类号:TP391.4 **文献标识码:**A

Parallel restoration for motion-blurred aerial image

LI Shi^{1,2}, ZHANG Bao¹, SUN Hui¹

- (1. *Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun 130033, China;*
2. *Graduate University of Chinese Academy of Sciences, Beijing 100039, China*)

Abstract: In order to improve the speed of restoration algorithm, a parallel Wiener filtering method based on Graphic Processing Unit(GPU)platform is presented to restore motion-blurred aerial image degraded in a deterministic way by motion or vibration. The shortages of the original algorithm using Wiener filtering and original PC run-time platform are introduced. On the basis of the new General Purpose GPU (GPGPU) technology, the original algorithm is divided into thousands of single algorithm threads to be computed in parallel. According to the special simultaneous operating mode of GPU hardware, a way in which the algorithm threads access the data on the GPU global memory is specially configured to improve the accessing speed, the algorithm efficiency by the special configuration can even be improved roughly 3 times that by original configuration. With the parallel computing ability of GPU, the new algorithm can restore 1024×1024 gray image in 8 ms per frame. The experimental result shows the new algorithm based on GPU reaches approximately 20 times that of original algorithm based on CPU of personal PC, which can completely be applied to the real time restoration of high resolution motion-blurred aerial image.

Key words: aerial imaging; parallel computing; Wiener filtering; image restoration; Graphic processing Unit(GPU)

收稿日期:2008-04-16;修订日期:2008-06-06.

基金项目:国防科技预研基金资助(No. 1040603)

1 引言

当载机的飞行高度与飞行速度比值达到一定条件^[1-2],航空成像的像移现象将严重地影响成像的质量。经过各种算法的验证及对比评测,作者在前期工作中使用加窗维纳滤波^[2]实现了对由像移造成的运动模糊的精确恢复。针对算法运算速度的实时性问题,先后使用一维维纳滤波^[3]及实数傅里叶变换^[4]等方法对算法进行加速优化,并取得了一定成果。为进一步提高图像恢复算法的运算效率,希望能对算法进行并行运算的改进。而当前 CPU 的浮点处理性能及 CPU 的数据带宽在一定程度上限制了算法并行处理的效果,常见的嵌入式并行处理方案也过于复杂,性能效果及成本考量均不十分理想。

美国 Sandia 国家实验室的 Kenneth Moreland 研究员及新墨西哥州大学的 Edward Angel 教授在 2003 年 SIGGRAPH 会议上发表了关于在 GPU 上运行 FFT 算法的文章^[5],GPU 厂商 nVidia 公司也在 2007 年 Super Computing 大会上宣称:GPU 的计算性能及数据带宽大约是同期 CPU 的 10 倍。基于以上因素,本文对 GPU 编程进行了深入研究,将原先基于 CPU 的像移模糊图像恢复算法移植到 GPU 平台上,充分运用 GPU 高数据带宽的并行运算能力,实现图像恢复的并行处理。

2 像移模糊恢复算法

图像像移模糊恢复的理论基础是基于信息光学的传递函数理论^[6]所导出的数学模型:模糊图像 $g(x,y)$ 由清晰图像 $f(x,y)$ 与点扩散函数 $h(x,y)$ 相卷积再加上噪声 $n(x,y)$ 所得,用公式表述如下:

$$g(x,y) = f(x,y) * h(x,y) + n(x,y). \quad (1)$$

在已知点扩散函数 $h(x,y)$ 的情况下,要通过数学方法从式(1)中求得清晰图像 $f(x,y)$,经实验证明行之有效的方法是将图像的空域信息转换到频域,然后通过维纳滤波算法(见式(2))进行求解^[1-4]。

$$F(u,v) = \frac{H(u,v)^*}{H(u,v)H(u,v)^* + k} G(u,v). \quad (2)$$

从式(2)注意到频谱 $F(u,v)$ 、 $G(u,v)$ 、 $H(u,v)$ 在进行维纳滤波时的各对应点的下标是相同的,且各频谱数组的相邻数据在滤波的时候是相互独立的。假设一个线程在一个周期内可以完成一个式(2)所示的频域数据计算,且并行的线程有 $M \times N$ 个,则在一个周期内可以同时完成 $M \times N$ 个式(2)所示的频域数据计算。

3 维纳滤波并行算法

3.1 GPGPU 与 FFT 算法

GPU 用于通用计算 (General Purpose GPU),一般是使用多纹理技术,集合纹理环境参数和纹理函数来实现的^[7],它的工作模式如图 1 所示。数据以图像的形式经由图形 API 函数传入 GPU 的纹理区, GPU 通过渲染一个适当的几何图形来触发 GPU 的数据并行运算,并将渲染(运算)结果返回纹理区。第 n 次输出的数据作为第 $n+1$ 次的输入源,数据在纹理区内被不停地渲染,直到预设运算结束。

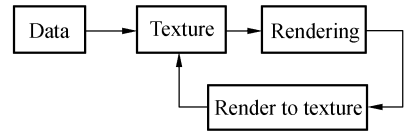


图 1 GPU 流式运算工作模式

Fig. 1 GPU data flow

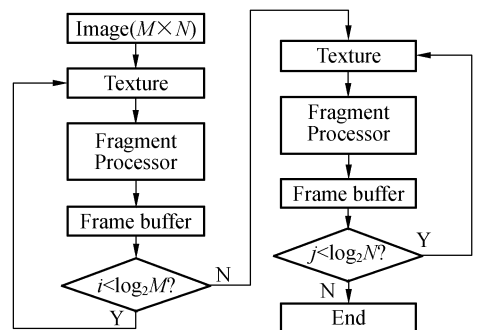


图 2 FFT 流式运算流程图

Fig. 2 Flow chart of FFT

采用分级蝶形算法实现的 FFT 算法,正适合用 GPU 这种流式着色运算来实现^[5,8]。以二维 FFT 为例:先画与屏幕平行的 $M \times N$ 四边形,让光栅处理器为四边形内的每个像素创建一个片

段(fragment),由此执行 fragment 程序。fragment 程序以单指令多数据流(SIMD)方式工作,由此实现 FFT 的并行处理。每执行一次 fragment 程序,即完成一次 FFT 算法中的行(或列)蝶形迭代。当完成 $\log_2 M + \log_2 N$ 次行(列)蝶形迭代后,存储区内的数据就是所求的二维 FFT 运算结果。详细算法流程参见图 2。

3.2 维纳并行滤波算法

将维纳滤波算法向并行运算改进的重点是怎样进行图像数据在各线程间的分配,更好地协调数据线程的运行,发挥多线程并行运算的优势。nVidia 公司在 2007 年推出的 GPU 开发包 Compute Unified Device Architecture(CUDA)对数据的处理方式类似图 1 所示的流式运算,它使用“线程”并行的概念实现全体像素点的同一处理。CUDA 的优势是能够清楚知道数据执行的先后顺序,并能对运算数据进行控制,降低维纳并行滤波算法开发的难度。

维纳并行滤波算法如图 3 所示,首先使用图 2 所示的 FFT 算法将模糊图像 $g(x, y)$ 及点扩散函数 $h(x, y)$ 转换成频域的 $G(u, v)$ 及 $H(u, v)$,然后根据数据存储的线性排列对 $G(u, v)$ 及 $H(u, v)$ 上的数据进行划分,不同的线程读取不同频域的数据,由每个线程独立地对线程内的数据进行式(2)的滤波。当所有线程上的数据滤波完毕后,再通过 FFT 的逆运算将数据 $F(u, v)$ 由频域图像转换成空间图像,实现图像的还原。

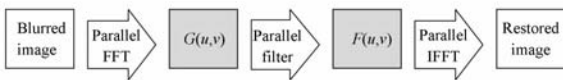


图 3 维纳并行滤波算法示意图

Fig. 3 Wiener parallel filter

CUDA 允许上千个线程在 GPU 上同时运行,为了让维纳滤波算法在 GPU 上充分地并行起来,理论上应该在 GPU 上开尽可能多的线程。但实际上随着线程数目的增加,每个线程所拥有的寄存器数目将减少,可能导致并行效率的下降。为更好地管理线程,本文将线程进行分块管理,并对应频域数据的划分,让每个维纳滤波线程都处理数据块上一定量的数据(如图 4 所示)。

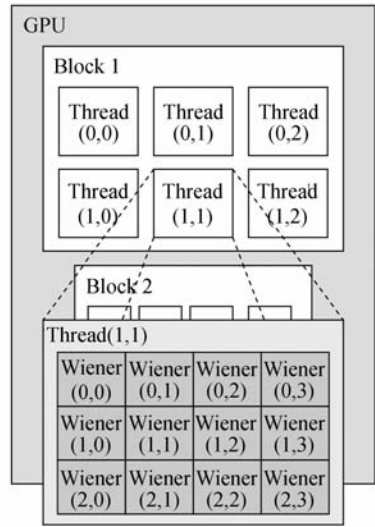


图 4 维纳滤波并行运算图解

Fig. 4 Illustration of parallel Wiener filter

3.3 并行算法优化

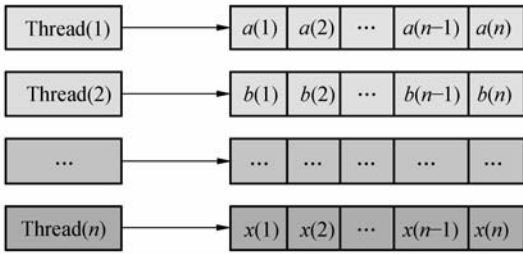
图像处理数据存放在图像显示卡的 DRAM 中,每个线程对数据的读写访问需要一定的时间间隔。设该访问时间为 t_r ,线程做数据地址寻址也需要一定的时间,设寻址时间为 t_d 。一般线程对数据读写所需的总时间为 $t_r + t_d$ 。当线程对 DRAM 内存上连续物理地址的数据做连续读取访问时,可近似认为寻址时间 $t_d = 0$ 。所以这里最有效率的数据存取方式是以连续的方式存取。

3.3.1 数据访问优化

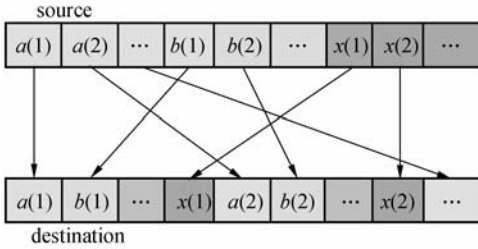
一般的数据连续存取方式是为每个线程划分一块连续的数据地址区,让每个线程能对各自辖区的数据块内容进行连续访问(如图 5(a)所示)。但是经研究发现 GPU 上线程的执行方式是:当一个线程在等待内存的数据时,GPU 会切换到下一个线程,以此提高 GPU 的执行效率。

在 GPU 的这种线程调配模式下,可近似认为 GPU 线程间的切换不需要时间。这时使用图 5(a)所示的数据存取方式,会使得原先连续存放的数据被非连续的存取(如图 5(b)所示),也就是说让一个线程连续地存取内存,在实际执行时反而不是连续存取。这时每个数据的存取均要重新寻址,随着数据量的增大,该方式的弊端边逐渐明显。

要让实际执行结果连续地存取,应该要让线程(1)读取连续地址的第一个数据,线程(2)读取第二个数据...依此类推(如图 6(a)所示)。这样



(a) 数据读取方式
(a) Data reading method

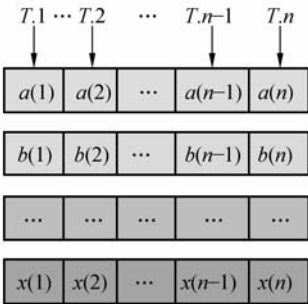


(b) 实际读取顺序
(b) Real reading sequence

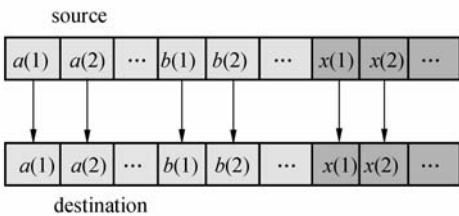
图 5 一般并行线程的数据读取方式

Fig. 5 General data accessing mode of parallel thread

读取出来的数据顺序就跟原先数据排列的顺序一致(如图 6(b)所示),可实现数据地址的连续访问。文中第 4 部分的实验数据表明这样的改进最多能将算法的执行效率提升到原来的 3 倍。



(a) 数据读取方式
(a) Data reading method



(b) 实际读取顺序
(b) Real reading sequence

图 6 改进后并行线程数据读取方式

Fig. 6 Improved data accessing mode

3.3.2 并行算法优化

连续内存块上数据 $G(k)$ 的序数 k 与将要访问它的线程的序号 i 间的关系由式(3)唯一确定:

$$i = k \% \text{threadNum}, \quad (3)$$

式中 $\%$ 为取余运算符, threadNum 为当前程序所开辟的线程总数。表述的意思是:当前数据 $G(k)$ 将要进入的运算线程的序号 i 是它的序数 k 被总线程数取余所得的余数。图 7 为经过线程数据访问优化的维纳并行滤波流程图。

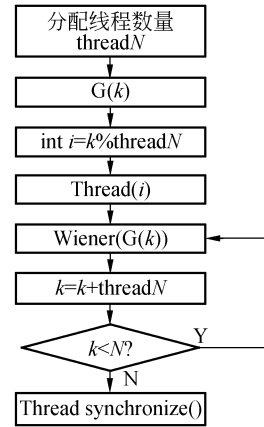


图 7 并行算法流程图

Fig. 7 Flow chart of parallel algorithm

4 实验结果

实验测试平台的 GPU 为 GeForce8800GTS (显存 500 M), CPU 是 P4 主频 3.0 G(单核), 内存 1 G。测试程序在 VC2005 环境下编译完成。针对实验用的 GPU, 本文开辟 8 块线程区, 每块拥有 256 个线程, 即 GPU 上总共有 2 048 个线程在并行执行维纳滤波。实验分别对分辨率为 256×256 、 512×512 、 $1\ 024 \times 1\ 024$ (图 8) 及 $2\ 048 \times 2\ 048$ 的图像进行像移模糊的恢复评价测试。图 9 为图 8 在水平、竖直方向均发生 10 个像移的情况下所得到的模糊图像, PSNR 为 20.89。图 10 为图 9 在维纳并行滤波后所得的恢复图像, PSNR 为 32.93。由图 10 及表 1 所得的像质评价数据, 可知维纳并行滤波算法的图像恢复质量与改进前的维纳滤波算法一致, 没有发生图像恢复质量下降的问题。

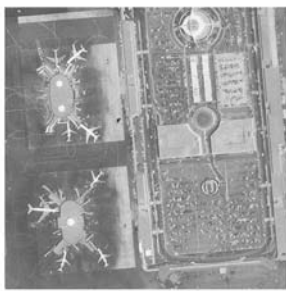


图 8 原始图像 (1 024×1 024)

Fig. 8 Original image (1 024×1 024)

表 1 并行算法 MSE、PSNR 值参照

Tab. 1 MSE and PSNR fore-and-aft restoration

图像	尺寸	MSE	PSNR
模糊(Airport)	1 024×1 024	528.86	20.89
恢复(Airport)	1 024×1 024	33.12	32.93



图 9 模糊图像 (1 024×1 024)

Fig. 9 Blurred image (1 024×1 024)



图 10 复原图像 (1 024×1 024)

Fig. 10 Restored image (1 024×1 024)

表 2 数据为对各分辨率图像分别使用基于 CPU 运算的维纳滤波算法①、基于 GPU 运算的维纳并行滤波算法②及经优化的维纳并行滤波算法③所得的时间测试结果。其中基于 CPU 运算的维纳滤波算法为文献[4]中经实数傅里叶变换改进

的维纳滤波算法。

表 2 中第 5 行数据显示,随着图像尺寸的增大,3.3 节所介绍的线程配置方法的优势越来越明显,当图像尺寸为 2 048×2 048 时,算法优化前后的效率相差甚至达到 3 倍。说明本文在对维纳并行滤波算法的各线程数据模块上的相应规划设置是有成效的。表 2 中最后一行数据表明,文中维纳滤波算法的并行运算改进效果明显,尤其在处理 1 024×1 024 图像时,效率提升到原有算法的 20.93 倍。

表 2 时间测试结果(单位:毫秒)

Tab. 2 Results of time test (unit: ms)

图像大小	256×256	512×512	1 024×1 024	2 048×2 048
原有算法①	6.93	40.31	165.96	721.53
并行算法②	0.599	3.01	11.51	136.89
优化的并行算法③	0.467	2.05	7.93	45.01
②/③	1.28x	1.47x	1.45x	3.04x
①/③	14.84x	19.66x	20.93x	16.03x

5 结 论

维纳并行滤波算法中使用的各线程数据访问方法将并行改进后的维纳滤波算法与 GPU 特殊的硬件并行架构很好地结合在一起,有效地发挥数据多线程并行处理的优势,使得算法的运算效率在算法并行改进的基础上再提高 28%~204%,有着一定的理论及实践价值。

维纳并行滤波算法在 GPU 平台上所取得的结果是惊人的:它能够在不到 8 ms 的时间内对一帧 1 024×1 024 分辨率的像移模糊图像进行像移模糊恢复,将原先算法^[4]的实验速度又提高了 20 倍,很好地解决了高分辨率航空像移模糊图像的实时恢复问题。

维纳并行滤波算法的运行是基于 GPU 处理器的,不用额外的辅助电路参与,整个算法方案简洁、灵活。该算法可以在笔记本电脑的独立图形显示卡上运行,不过笔记本电脑的硬件可扩展性较差,怎样将 GPU 向嵌入式平台推进,还有一段摸索的过程。

参考文献:

- [1] 贾平,张葆,孙辉. 航空成像移模糊恢复技术[J]. 光学精密工程,2006,14(4):697-703.
JIA P, ZHANG B, SUN H. Restoration of motion-blurred aerial image [J]. *Opt. Precision Eng.*, 2006,14(4):697-703. (in Chinese)
- [2] 孙辉,张葆,刘晶红. 基于维纳滤波的运动模糊消除算法及其在航空成像系统中的应用[J]. 光学精密工程,2005,13(6):735-740.
SUN H, ZHANG B, LIU J H. Restoration of motion-blurred image based on Wiener filter and its application in aerial imaging system [J]. *Opt. Precision Eng.*, 2005,13(6):735-740. (in Chinese)
- [3] 李仕,孙辉,张葆. 一种运动模糊图像的实时恢复算法[J]. 光学精密工程,2007,15(5):767-772.
LI SH, SUN H, ZHANG B. A method for real-time restoration of motion-blurred images [J]. *Opt. Precision Eng.*, 2007,15(5):767-772. (in Chinese)
- [4] 李仕,张葆,孙辉. 航空光电成像模糊的实时恢复[J]. 光学精密工程,2007,15(8):1287-1292.
LI SH, ZHANG B, SUN H. Real-time restoration using real discrete Fourier transform for aerial E-O imaging system [J]. *Opt. Precision Eng.*, 2007,15(8):1287-1292. (in Chinese)
- [5] MORELAND K, ANGEL E. The FFT on a GPU [C]. In *SIGGRAPH/Eurographics Workshop on Graphics Hardware 2003 Proceedings*, 2003:112-119.
- [6] 韩昌元. 信息光学基础理论及其应用[M]. 长春:长春出版社,1989.
HAN CH Y. *Fundamental Theory and Application of Information Optics* [M]. Changchun: Changchun Press, 1989. (in Chinese)
- [7] PHARR M, FERNANDO R. *GPU Gems2* [M]. Addison-Wesley Professional, 2005.
- [8] 邓劲. 图形处理器上的快速傅立叶变换[J]. 现代电子技术,2007(10):151-154.
DENG J. FFT on graphics processing unit [J]. *Modern Electronic Technique*, 2007(10):151-154. (in Chinese)

作者简介:



李仕(1984—),男,浙江苍南人,博士研究生,主要从事航空成像补偿算法方面的研究。E-mail: brightlishi@gmail.com



孙辉(1963—),男,吉林柳河人,研究员,主要研究方向为数字图像处理与分析、计算机仿真技术等。E-mail: sunh@ciomp.ac.cn



张葆(1966—),男,吉林盘石人,研究员,工学博士,博士生导师,主要从事航空光电成像技术的研究。E-mail: cleresky@vip.sina.com