

DOI: 10.3969/j.issn.1673-6141.2020.02.002

基于 GPU 的太阳图像斑点重建技术实现

宣经纬^{1,2,3}, 饶长辉^{1,2}, 钟立波^{1,2}, 田雨^{1,2}

(1 中国科学院自适应光学重点实验室, 四川 成都 610209;

2 中国科学院光电技术研究所, 四川 成都 610209;

3 中国科学院大学, 北京 100049)

摘要: 在地基太阳观测中, 光线在穿越大气层时会受到大气湍流的影响而导致图像扭曲、变形以致质量下降。为了消除或降低大气湍流的影响, 事后图像处理技术被用来获得高分辨力的太阳图像。基于斑点干涉法和斑点掩模的事后重建算法可以获得高分辨力的图像, 但由于计算复杂度高, 难以满足实时性的要求。在讨论了算法原理的基础上, 使用 CUDA 并行计算架构实现了太阳斑点重建算法并行化。实验结果表明, 在 GPU 环境下, 一张 TiO 通道 2304 pixel×1984 pixel 像素大小的图像, 可以在 70 s 内完成重建, 相比运行在 CPU 上的串行程序, 加速比可达 7 以上。

关键词: GPU; CUDA; 图像重建; 斑点干涉法; 斑点掩模法; 并行计算

中图分类号: TN911.73 **文献标识码:** A **文章编号:** 1673-6141(2019)02-0090-011

Implementation of Solar Speckle Image Reconstruction Based on GPU

XUAN Jingwei^{1,2,3}, RAO Changhui^{1,2}, ZHONG Libo^{1,2}, TIAN Yu^{1,2}

(1 Key Laboratory of Adaptive Optics, Chinese Academy of Science, Chengdu 610209, China;

2 Institute of Optics and Electronics, Chinese Academy of Science, Chengdu 610209, China;

3 University of Chinese Academy of Sciences, Beijing 100049, China)

Abstract: In ground-based solar observation, when the light passes through atmosphere, it will be affected by atmospheric turbulence which will cause translation, distortion and blurring of the received image. In order to eliminate or reduce the effects of atmospheric turbulence, post-image processing technology was used to obtain high-resolution solar images. High-resolution image can be obtained by post-facto reconstruction algorithm based on the speckle interferometry and speckle masking. However, due to its complex calculation, the algorithm can not meet the requirement of real-time detection. On the basis of the principle of the algorithm, solar speckle image reconstruction algorithm has been parallelized by using CUDA parallel computing architecture in this research. The experimental results show that a 2304 pixel×1984 pixel image of TiO channel

基金项目: Supported by National Natural Science Foundation of China (国家自然科学基金, 11727805, 11703029), National Defense Science and Technology Innovation Fund of Chinese Academy of Sciences(中国科学院国防科技创新基金, YJ16K006)

作者简介: 宣经纬 (1993 -), 男, 江苏盐城人, 研究生, 主要从事图像并行计算方面的研究。 **E-mail:** xuan.j.w@foxmail.com

导师简介: 饶长辉 (1971 -), 男, 四川成都人, 博士, 研究员, 博士生导师, 主要从事自适应光学方面的研究。

E-mail: chrao@ioe.ac.cn

收稿日期: 2019-03-08; **修改日期:** 2019-03-29

can be reconstructed within 70 s under GPU environment. Compared with the program running on CPU, the speed-up ratio can up to 7.

Key words: GPU; CUDA; image reconstruction; speckle interferometry; speckle masking; parallel computing

1 引言

太阳是离地球最近的恒星, 是人类赖以生存的重要恒星, 太阳中发生的许多活动会对我们的生活产生影响。为了减轻太阳活动对人类生活的影响, 需要对太阳进行观测, 以提前预测太阳活动, 以便做出及时的应对措施。而在地基太阳观测中, 光线在穿越大气层时会受到大气湍流的影响产生波前误差, 使接收到的图像出现不同程度的平移、扭曲和模糊, 难以达到望远镜的衍射极限^[1]。为了消除大气湍流对成像的影响, 研究人员们提出了众多的解决方法, 其大致可以分为两大类: 针对成像过程的自适应光学^[2](Adaptive optics, AO) 和针对成像结果的事后图像重建^[3]。

太阳图像事后重建中常使用的算法有许多, 包括: 斑点重建、多帧盲解卷积、相位差法等, 本文采用斑点重建法。斑点重建又分为振幅重建和相位重建, 振幅重建采用的是 Labeyrie 提出的斑点干涉法^[4], 相位重建则采用的是 Weigelt 提出的斑点掩模法^[5]。这两种算法都需要对大量的瞬时短曝光图像进行复杂计算, 数据处理十分耗时, 无法达到实时重建的要求。近些年来, 非常擅长处理密集型数据的 GPU 技术开始被尝试用到天文图像重建中, 并取得了不错的效果。

上海天文台 Ding 等^[6]对 1.56 m 望远镜上采集的 329 pixel×248 pixel 大小双星斑点图像, 利用一块 NVIDIA Quadro 2000D GPU, 实现了斑点干涉算法重建双星图像的加速, 获得了约 7 倍的加速。云南天文台李雪宝^[7]利用 CUDA 实现了最耗时的重谱计算算法的并行化, 重建单帧 TiO 通道 256 pixel×256 pixel 子块图像, 运行时间减少至 1.1 s, 获得了约 6 倍加速。云南天文台向永源^[8]对斑点掩模法进行了深入的分析和优化, 最大程度地精简了运算量, 实现了子块图像四维重谱并行计算, 另外也在集群中实现了斑点掩模法的快速重建。Wöger 等^[9]为正在建设中的 DKIST 望远镜设计了一个配备 NVIDIA Tesla C2050 GPU 的实时数据处理平台, 利用改写自 KISIP 的算法, 实现了 225 个 128 pixel×128 pixel 像素大小子块的相位在 4.2 s 内重建; 并期望在未来, 利用下一代 GPU 集群, 实现 3 s 内重建完一组 (80 帧) 大小 4k pixel×4k pixel 的图像。

本文在介绍了算法原理的基础上, 详细论述了斑点重建算法在 GPU 上的并行化方案, 使用 CUDA 并行计算架构实现太阳斑点重建算法并行化。

2 斑点重建基本原理

斑点图最初是指激光经过随机介质, 受到空间随机相位扰动后形成的波前干涉图案。这种图案会呈现出散开的粒状光斑, 故称为“斑点图”。在太阳观测过程中, 受大气湍流影响, 望远镜探测面的波前相位也会受到扰动, 形成斑点图案。在天文学观测中, 斑点图通常采用短曝光 (曝光时间小于 10 ms) 的方式来获取。研究发现, 这种由短曝光方式采集到的斑点图案, 其最小结构尺寸和望远镜不受大气湍流以及像差影响时的艾里斑半径相同, 即斑点图中含有更多的目标细节信息可用于高分辨力重建。斑点图像重建分为两大部分, 目标的傅里叶振幅 (模) 重建和目标的傅里叶相位重建。原始图像经傅里叶变换到频率域, 再通过不同的算法复原振幅和相位, 经傅里叶逆变换得到复原图像。

2.1 振幅重建

1970 年, Labeyrie^[4] 发现短曝光图像可以将大气湍流的变化“冻结”住, 此时的短曝光图像中包含了用于高分辨力重建的高频信息。同时, 他还发现短曝光光学传递函数 (Optical transfer function, OTF) 模的平方的系综平均可以在较高频率上获得可观的幅值。

短曝光情况下图像的傅里叶变换为

$$I_i(f) = O(f)S_i(f) \quad 0 \leq i \leq N, \quad (1)$$

式中 $I_i(f)$ 为第 i 张短曝光图像的傅里叶变换, 在实际工程中, 一般选取 100 张短曝光图像; $O(f)$ 为目标傅里叶变换, $S_i(f)$ 为第 i 张短曝光图像拍摄时的光学传递函数。

多帧短曝光模的平方的系综平均为

$$\langle |I_i(f)|^2 \rangle = \langle |O(f)|^2 \rangle \langle |S_i(f)|^2 \rangle, \quad (2)$$

式中, $\langle \rangle$ 表示系综平均; $\langle |S_i(f)|^2 \rangle$ 称为斑点传递函数 (Speckle transfer function, STF)。

$$T(f) = |S_i(f)|^2. \quad (3)$$

在一组短曝光图像拍摄过程中, 由于时间非常的短 (100 帧大约 1 s), 可以假设目标是恒定不变的, 故可认为 $|O(f)|^2 = \langle |O(f)|^2 \rangle$ 。

由公式 (2) 可得到目标的功率谱

$$|O(f)|^2 = \frac{\langle |I_i(f)|^2 \rangle}{\langle |S_i(f)|^2 \rangle}. \quad (4)$$

此时, 如果 STF 已知, 目标功率谱就可以由式 (4) 计算得出。在太阳观测中, 由于缺少夜天文观测中的参考星或参考点源, 因此 STF 需要由建模得出。研究发现, 当大气湍流引入的波前畸变满足 Kolmogorov 谱^[10] 时, 可以通过计算大气相干长度 r_0 得到 STF。1983 年由 Von der Lühe 提出的谱比法^[11] 被广泛用于太阳图像斑点重建中计算大气相干长度 r_0 。

谱比的定义为

$$\varepsilon(f) = \frac{\langle |I_i(f)|^2 \rangle}{\langle |I_i(f)|^2 \rangle} = \frac{|O(f)|^2 \langle |S_i(f)|^2 \rangle}{|O(f)|^2 \langle |S_i(f)|^2 \rangle} = \frac{\langle |S_i(f)|^2 \rangle}{\langle |S_i(f)|^2 \rangle}. \quad (5)$$

由式 (5), 可以计算出一系列不同 r_0 情况下的 $\varepsilon_1(f)$ 值, 再由采集到的短曝光图像计算谱比值 $\varepsilon_2(f)$, 匹配与 $\varepsilon_2(f)$ 最接近的 $\varepsilon_1(f)$ 值, 就可以认为此时的 $\varepsilon_1(f)$ 与 $\varepsilon_2(f)$ 具有相同的相干长度, 即认为生成 $\varepsilon_2(f)$ 的 r_0 就是 $\varepsilon_1(f)$ 所对应的相干长度 r_0 。得到了 r_0 也就可以计算出 STF, 代入式 (3) 得到目标的功率谱 $|O(f)|^2$, 对功率谱开平方就可以得到目标傅里叶振幅 $|O(f)|$ 。

2.2 相位重建

Weigelt 和 Lohman 提出的斑点掩模法 (Speckle Masking)^[5] 常被用于重建相位, 斑点掩模法引入了两个概念, 即三相关 (Triple correlation) 和双频谱 (Bispectra)。

三个强度分布 $I_n(x)$ 、 $I_m(x)$ 和 $I_k(x)$ 的三相关定义为

$$I_{nmk}^{(3)}(x, x') = \int I_n(x'') I_m(x'' + x) I_k(x'' + x') dx'', \quad (6)$$

三相关的傅里叶变换, 即双频谱的定义为

$$\tilde{I}_{nmk}^{(3)}(u, v) = \int I_{nmk}^{(3)}(x, x') \exp[-2\pi i(ux + vx')] dx dx', \quad (7)$$

由式 (6) 和式 (7) 可以得到

$$\tilde{I}_{nmk}^{(3)}(u, v) = \tilde{I}_n(u) \tilde{I}_m(v) \tilde{I}_k(-u - v), \quad (8)$$

其中 $\tilde{I}_n(u)$ 、 $\tilde{I}_m(v)$ 、 $\tilde{I}_k(-u - v)$ 分别是 $I_n(x)$ 、 $I_m(x)$ 、 $I_k(x)$ 的傅里叶变换。特别的, 当 $n = m = k$ 时, 令 $\tilde{I}_n^{(3)}(u, v) = \tilde{I}_{nmk}^{(3)}(u, v)$, 此时有

$$\tilde{I}_n^{(3)}(u, v) = \tilde{I}_n(u) \tilde{I}_n(v) \tilde{I}_n(-u - v), \quad (9)$$

结合式 (1) 和卷积理论, 可得

$$\tilde{I}_n^{(3)}(u, v) = \tilde{O}^{(3)}(u, v) \tilde{S}_n^{(3)}(u, v), \quad (10)$$

其中 $\tilde{I}_n^{(3)}$ 、 $\tilde{O}^{(3)}$ 、 $\tilde{S}_n^{(3)}$ 分别是 I_n 、 O 、 S_n 的双频谱。则式 (10) 的系综平均形式为

$$\langle \tilde{I}_n^{(3)}(u, v) \rangle = \tilde{O}^{(3)}(uv) \langle \tilde{S}_n^{(3)}(u, v) \rangle. \quad (11)$$

式 (11) 中的 $\langle \tilde{S}_n^{(3)}(u, v) \rangle$ 被称为广义斑点传递函数。研究表明, $\langle \tilde{S}_n^{(3)}(u, v) \rangle$ 是实值函数且在望远镜截断频率以下大于 0。所以可以得到

$$\text{phase}\{\tilde{I}_n^{(3)}(u, v)\} = \text{phase}\{\tilde{O}^{(3)}(u, v)\}. \quad (12)$$

据此可以认为, 目标的相位信息可以由双频谱的相位递推出来。

2.3 重建流程

太阳图像斑点重建的流程如图 1 所示, 首先从硬盘中读入重建数据到内存中, 接着对当前子块执行重建程序直至所有子块重建完毕, 最后将重建完的子块拼接为大图。其中子块重建包含以下几个步骤: 子块分割、多帧平均、相关对齐、振幅重建、相位重建。本节将详细介绍子块分割、多帧平均、相关对齐、子块拼接这几个步骤的原理, 振幅重建和相位重建在前面已有涉及, 故不再赘述。

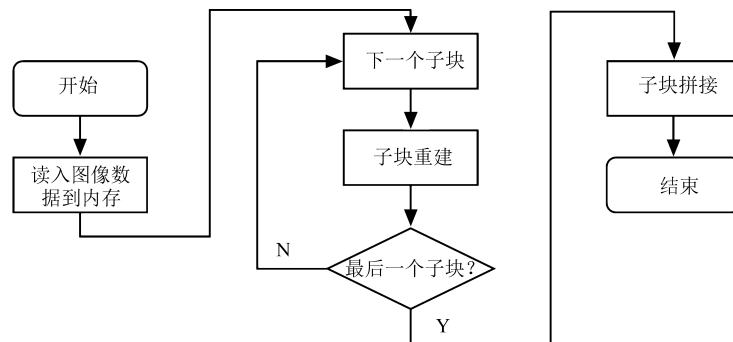


图 1 太阳图像斑点重建流程图

Fig.1 Flowchart of solar image speckle reconstruction

1) 子块分割: 大气湍流的分布是三维的, 这使得不同方向目标的光线在穿越大气时受到的扰动不同, 这就是大气湍流的非等晕特性^[12]。目标在一定的角度内受到的扰动可以认为是近似相同的, 该角度称为等晕角, 对应的图像区域称为等晕区。假设可以把原始视场分割为众多的子块, 使得在这些子块内等晕性近似成立, 这就是分割子块的目的。斑点重建算法是以子块为单元进行重建的, 这种特性使得其非常适合重建出大视场的高分辨力图像。同时, 以子块为重建单元的特性使得子块间重建过程相对独立, 无论是在集群中, 还是多核 CPU 中, 亦或是核心此处采用的 GPU 技术, 并行任务的划分就会变得十分容易。考虑到后续计算会使用快速傅里叶变换 (FFT) 算法, 子块大小的像素数一般选定为 2 的幂次方, 如 64, 128, 256 等。

2) 多帧平均: 分割后的子块内各帧间会有一些的平移, 需要对偏移进行校正。校正的基准有两种选取方法: 第一种方法是在子块内若干短曝光图像中任意选取一张对比度相对较高的作为参考基准。第二种方法是计算出子块内所有帧的平均值, 校正的时候就以这张平均图 (值) 为基准计算出它与其他各帧的偏移量, 逐一校正。计算式如下

$$m_i = \frac{1}{N}(f_1 + f_2 + \dots + f_N), \quad (13)$$

式中, m_i 为第 i 个子块的多帧平均值; i 为子块号; N 为子块内的总帧数; $(f_1 + f_2 + \dots + f_N)$ 为当前子块内的连续短曝光图像。

3) 相关对齐: 受湍流和风力的影响, 采集到的短曝光图像帧与帧之间会产生抖动偏移, 倘若不进行偏移纠正会影响到后续的重建结果。计算偏移量的方法有很多, 这里采用归一化互相关 (Normalization cross correlation, NCC) 算法。归一化互相关算法具有对光照强度不敏感、抗干扰的特点, 常被用在图像匹配领域中。假设有两幅图像 $I_r(x, y)$ 与 $I_n(x, y)$, 它们之间的归一化互相关矩阵的计算方法为

$$C_{r,n}(\delta_x, \delta_y) = \frac{\iint I_r(x + \delta_x, y + \delta_y)I_n(x + \delta_x, y + \delta_y)dxdy}{\sqrt{\iint I_r^2(x + \delta_x, y + \delta_y)dxdy} \sqrt{\iint I_n^2(x + \delta_x, y + \delta_y)dxdy}}. \quad (14)$$

归一化互相关矩阵的最大值位置相对图像中心点的偏移就是两幅图像间的相对偏移。为了简化计算, 根据空域的卷积等于频域的相乘这一性质, 可以将计算过程简化为

$$C_{r,n} = \mathfrak{F}^{-1} \left\{ \frac{\mathfrak{F}\{I_r(x, y)\}\mathfrak{F}\{I_n(x, y)\}}{\sqrt{\iint \mathfrak{F}\{I_r(x, y)\}\mathfrak{F}^*\{I_r(x, y)\}dk_xdk_y} \sqrt{\iint \mathfrak{F}\{I_n(x, y)\}\mathfrak{F}^*\{I_n(x, y)\}dk_xdk_y}} \right\}. \quad (15)$$

4) 子块拼接: 得到子块图形的傅里叶模和傅里叶相位后, 经傅里叶逆变换就可以得到重建后的子块图, 但这不是最终需要的结果, 还需要将重建完成的子块对齐和拼接成最终的单帧高分辨力图像。子块拼接分为两个步骤来实施: 首先是将每个子块和之前计算过的多帧平均图对齐, 接着将重建后的子块拼接为最终的大图。

3 斑点重建算法 GPU 并行实现

利用钟立波^[13]等设计出的 C 语言串行程序, 由 100 帧 128 pixel×128 pixel 像素大小的短曝光子块图像重建出 1 帧 128 pixel×128 pixel 像素大小的高分辨力子块图像, 在 CPU 上各计算步骤的耗时如表 1 所示。可以看出整个算法最耗时的步骤是子块分割、相关对齐、振幅重建中的谱比计算、相位重建中的双频谱计算, 大约占到了单个子块重建时间的 78.9%。下面主要讨论这四个步骤的并行化方法与优化策略。

表 1 单个子块 CPU 串行重建时间

Table 1 Reconstruction time of single sub-image cubes by CPU

Item	Time-consuming by CPU(ms)	Occupancy time ratio(%)
Sub-image segmentation	71	14.5
Multi-frame mean	13	2.7
Correlation alignment	68	13.9
Spectral ratio calculation	83	16.9
Bispectrum calculation	205	41.8
Phase recursion	40	8.2
The others	50	10.2
Total time-consuming of single sub-image	490	

3.1 斑点重建算法并行化实现

1) 子块分割并行化

斑点重建的输入是约 100 张的短曝光图像, 子块分割需要在这 100 张图像的不同位置分割出相同大小的小图像, 这 100 张小图像的集合称之为子块。在内存的维度上思考子块分割的过程, 其实上是一个存储区寻址和拷贝的过程, 不涉及复杂的运算。因此, 考虑使用和子块中像素数一样多的线程来实现并行化, 即一个像素对应一个线程。

2) 多帧平均并行化

计算多帧图像平均图的目的是为下一步的相关对齐做准备, 其基本计算过程是对每一帧中的相同像素位置的像素值做累加再求平均。在 GPU 上实现这部分算法时, 采用的是设置和单帧图像像素数一样多的线程实现加速的。其具体做法为: 以 100 帧 128 pixel×128 pixel 大小的子块图像为例, 设置线程数为 128 pixel×128 pixel, 每一个线程负责且只负责最终平均图上一个像素位置的值。每个线程对 100 帧图像中相同像素位置的像素值进行累加, 累加值再除以总帧数就得到了平均图。

3) 相关对齐并行化

相关对齐是为了校正帧与帧之间的位移, 使用互相关算法可以矫正。互相关运算在频域中计算起来更为方便, 因而需要对每一帧图像都进行二维傅里叶变换, 再对每帧的二维傅里叶变换结果的共轭和参考帧的二维傅里叶变换结果进行点乘, 点乘的结果再进行傅里叶逆变换就得到互相关矩阵, 互相关矩阵最大值的位置相对中心点的偏移量即该帧相对参考帧的位移。这部分的主要运算是傅里叶变换和像素级乘法以及坐标寻址, 也非常适合做并行化。

具体的做法如图 2 所示。以 100 帧 128 pixel×128 pixel 像素大小的子块图像为例, 首先利用 CUDA 提供的 CUFFT 库, 对分割后的子块中 100 帧图像 $f(i)$ 做傅里叶正变换, 得到 $F(i)$; 同时对上一步中的平均图利用 CUFFT 做傅里叶正变换, 得到 $M(0)$, 变换结果再取共轭, 得到 $M^*(0)$ 。接下来设置线程数为 128×128×100, 每个线程负责 $F(i)$ 中一个像素点和 $M^*(0)$ 的点乘, 得到结果 $R(i)$, 再利用 CUFFT 对 $R(i)$ 做傅里叶逆变换得到 $r(i)$ 。

在 $r(i)$ 中寻找最大值像素点时, 运用了 GPU 运算中常用的归约算法。设置和帧数一样多的 100 个线程块, 线程块内最大化线程数为 1024 (本实验使用的 K80 GPU 允许线程块内最大线程数是 1024), 每个线程块负责 $r(i)$ 中的一帧, 每个线程块内的 1024 个线程调用核函数寻找最大像素值的位置。

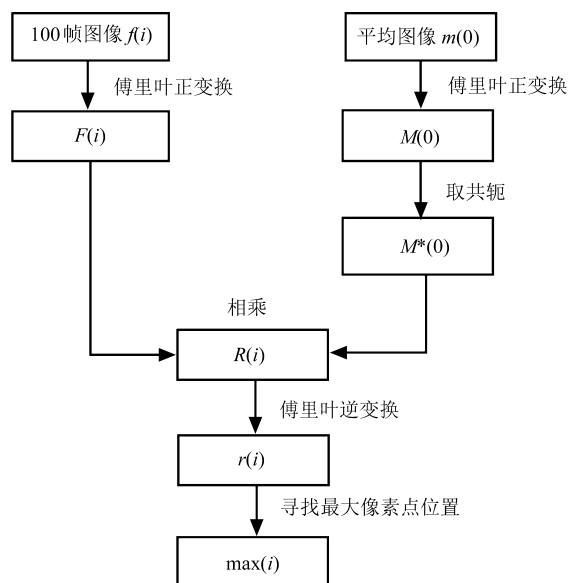


图 2 相关对齐示意图

Fig.2 Schematic sketch of correlation alignment

4) 谱比计算并行化

计算谱比前先要求得平均功率谱 $\langle |I_i(f)|^2 \rangle$ 和功率谱的平均 $\langle |I_i(f)|^2 \rangle$ 。以 100 帧 $128 \text{ pixel} \times 128 \text{ pixel}$ 像素大小的子块图像为例, 首先使用 CUDA 提供的 CUFFT 库对子块中 100 帧图像进行批量傅里叶正变换; 接着, 设置线程数为 $128 \times 128 \times 100$, 在 Kernel 中计算 100 帧图像傅里叶变换的平均和功率谱的平均; 然后, 设置线程数为 128×128 , 在 Kernel 中由上一步的傅里叶变换的平均计算平均功率谱 (平均傅里叶变换的功率谱); 最后, 平均功率谱与功率谱的平均相除即得谱比值, 继而可求得斑点传递函数, 有了斑点传递函数进而求得傅里叶振幅。

5) 双频谱获计算并行化

双频谱的计算量非常的大, 几乎占到了整个算法耗时的一半, 因此双频谱计算的优化是重中之重。这里仍以 100 帧 $128 \text{ pixel} \times 128 \text{ pixel}$ 像素大小的子块图像为例来说明, 首先使用 CUDA 提供的 CUFFT 库对子块中 100 帧图像进行批量傅里叶正变换; 接着, 设置线程数为 $100 \times 128 \times 128$, 对上一步批量变换后的图像频谱进行平移, 每个线程负责处理单个像素点的频谱; 然后调用 bs_init 函数, 该函数运行在 CPU 上, 其作用是根据重建参数对双频谱进行精简, 并返回最终要被使用到的双频谱数量 bs_cnt 以及双频谱索引矩阵, 这样就不需要计算全部位置的双频谱而只需要计算我们需要用到的位置的双频谱, 无疑降低了计算量; 在得到重建要用到的双频谱数量 bs_cnt 后, 设置线程数为 $100 \times \text{bs_cnt}$, 其目的是每一帧都分配 bs_cnt 个线程, 这 bs_cnt 个线程负责计算当前帧 bs_cnt 个位置的双频谱。各线程在调用 Kernel 函数时, 需要首先计算当前像素频谱的共轭, 共轭的结果再与平移前的像素频谱以及平移后的像素频谱相乘, 即得到当前像素的双频谱; 最后设置线程数为 bs_cnt, 各线程调用 Kernel 函数计算平均双频谱。计算好的平均双频谱拷贝回 CPU 内存, 由 CPU 完成接下来比较复杂的相位递推过程。

3.2 斑点重建算法优化策略

使用 GPU 加速斑点重建算法, 要想提高程序的执行速度, 除了不断扩大程序的并行规模和减少 Kernel 函数的执行耗时, 充分地利用 GPU 和 CUDA 既有的特性来优化程序也非常的重要, 本节将介绍斑点重建算法并行化过程中的优化策略。

1) 使用 CufftPlanMany 代替 CufftPlan2D

斑点重建算法中涉及到大量的傅里叶变换, 如果能有效减少傅里叶变换的耗时, 将在很大程度上减少程序的总耗时。CUFFT 中提供的标准函数每次只能对单张图像做傅里叶变换, 而标准函数 CufftPlanMany 每次可以同时有多张图像进行傅里叶变换。为体现它们性能的差异, 将它们与 CPU 实现中公认最快的 FFTW 函数库做对比。比较过程中, 均选取大小为 128 pixel×128 pixel 像素的图像, 在 100、200、500、1000 帧四个场景下, FFTW、CufftPlan2D、CufftPlanMany 做傅里叶变换的耗时, 以及 CufftPlanMany 相比 FFTW 的加速比, 结果如表 2 所示。可以看出随着图像帧数的增加, CufftPlanMany 的性能始终保持在 FFTW 的 3 倍左右。因此, 此处采用 CufftPlanMany 一次性对 100 张短曝光图像做批量傅里叶变换。

表 2 FFT 计算时间对比

Table 2 Computation time comparison of FFT

Number of frames	FFTW(ms)	CufftPlan2D(ms)	CufftPlanMany(ms)	Speed-up of CufftPlanMany
100	6	14	2	3
200	12	30	4	3
500	30	72	8	3.75
1000	60	145	16	3.75

表 3 float 类型加法、减法、乘法和除法的延迟和吞吐量

Table 3 Latency and throughput of addition, subtraction, multiplication and division of the float type

Operation	Latency (clocks)	Throughput (ops/clocks)
addition	24	7.9
subtraction	24	7.9
multiplication	24	7.9
division	137	1.5

2) 减少 GPU 全局内存的访问次数

GPU 的内存单元包括全局内存、纹理内存、常量内存、共享内存、局部内存、寄存器等。各种内存单元的访存延迟不同, 延迟长短的顺序为: 全局内存 > 局部内存 > 纹理内存 > 常量内存 > 共享内存 > 寄存器。因而需要尽可能减少 GPU 全局内存的访问, 具体做法包括两个方面: 第一, 在 Kernal 函数中将需要重复使用的数据存放在共享内存中; 第二, 在 Kernal 函数中将需要重复使用的常量数据放在常量内存中。

3) 适当的合并 Kernal 函数

将那些 grid 和 block 大小相同、彼此间没有关联、输入参数类似的 Kernal 函数合并, 一方面可以减少 Kernal 函数启动和返回的耗时, 另一方面也能减少对于全局内存的访问次数。此外需要注意合并 Kernal 函数带来单个线程寄存器使用数量过多的问题, 这将导致单个 SM 硬件资源能够运行的线程数量减少, 并行规模低, GPU 的性能也就无法发挥出来。因此, 在考虑合并 Kernal 函数时需要考虑多个方面的因素。

4) 算术运算优化

对 GPU 并行化来说, 除了要关注 GPU 对算术运算功能的支持, 还需要关注算术运算的性能。Henry Wong^[14] 等对 Tesla 架构的 GT280 GPU 进行了测试, 表 3 摘取了其中对 float 类型加法、减法、乘法和除

法的延迟和吞吐量的统计。该型 GPU 的 float 类型除法计算延迟约是加法、减法、乘法的 5~6 倍;相反,除法计算单位时钟周期的吞吐量远小于加法、减法和乘法。

对于除法耗时比较严重的问题,此处提出的一种解决方案是利用乘法代替除法,即除以一个数等于乘以它的倒数。例如在多帧平均并行化中的 Kernal 函数中最后需要除以总帧数,可以改为乘以总帧数的倒数。

在 GPU 并行化开发过程中,算术运算更重要的是实现基本的运算功能,只有在访存优化完成,计算成为瓶颈的情况下才会考虑对算术运算进行优化。

4 重建结果和测时分析

4.1 实验环境

实验采用一台配有众核处理器的 AMAX 服务器。服务器配有 Intel Xeon E5-2670 v3 12 核 CPU,主频为 2.3 GHz,内存为 32GB;GPU 为英伟达公司的 Tesla K80,其具有 4992 个 CUDA 内核,双 GPU 设计,24GB 显存,单精度浮点型性能达 8.73 Teraflops;操作系统为 Windows 7;开发环境为 Visual Studio 2015, CUDA 9.0。

实验数据选取了 100 帧采自 NVST(新真空望远镜)的 TiO 通道短曝光图像,图像尺寸为 2304 pixel×1984 pixel 像素,按照等晕区划分为 30×35 共计 1050 个 50% 重叠的子块序列。

4.2 重建效果

利用加速后的斑点重建算法,对 100 帧 TiO 通道的短曝光斑点图序列进行重建,图 3(a) 为原始图像,图 3(b) 为串行算法重建图像,图 3(c) 为并行算法重建图像。从图中可以看出,使用重建算法时候,图像的对比度更高,轮廓和细节也更加清晰,分辨力得到了极大的提高。将 GPU 上并行化的重建结果与 CPU 上的串行重建结果做差后发现,并行化的重建结果和串行重建结果像素值存在差异,但这种差异是整体上普遍变高或变低。分析认为,带来这种差异的原因是 GPU 浮点计算误差造成的。

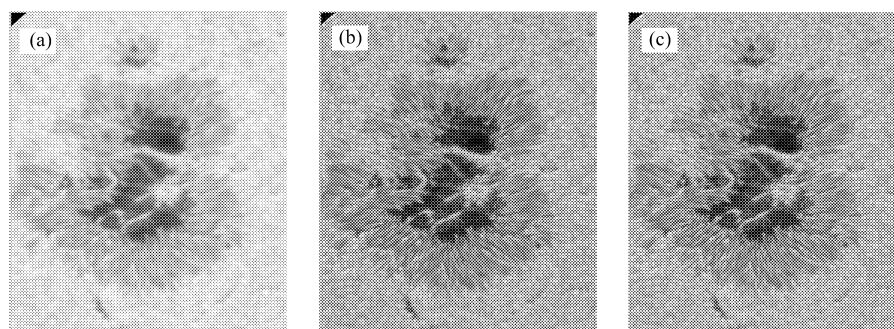


图 3 (a) 原始图像; (b) 串程序重建图像; (c) 并程序重建图像

Fig.3 (a) Original image; (b) reconstruction result by serial program; (c) reconstruction result by parallel program

4.3 并行与串行测时比较

斑点掩模算法各子块间处理独立,且多个子块的重建加速在单个 GPU 内很难实现,所以得到的结果指的是单个子块在单个 GPU 内的并行化实现,而完整大图可在所有子块重建完成后通过拼接得到。

表 4 列出了从 100 帧 128 pixel×128 pixel 像素大小的短曝光子块图像重建出 1 帧 128 pixel×128 pixel 像素大小的高分辨力子块图像各处理步骤的耗时和加速比。单个子块的总运行时间由 490 ms 减少到 53 ms,获得了 8.1 倍的加速。子块分割运行时间减少到了 0.47 ms,获得了约 74 倍的加速,多帧平均运行时间减

少到了 0.27 ms, 获得了约 48 倍的加速, 相关对齐运行时间减少到了 4 ms, 加速了约 17 倍, 谱比计算和双频谱计算也都获得了约 20 倍的加速。可见, 各处理步骤都得到了显著的加速。同时也注意到, 单个子块总耗时的加速比只有 8 左右, 远低于上述几个重建步骤的加速比。其原因是本文只对整个斑点重建过程的一部分做了加速, 算法中还有一些串行计算的存在, 故而单个子块总耗时的加速比要低于这几个被加速步骤的加速比。

表 4 单个子块 GPU、CPU 加速效果对比

Table 4 Comparison of acceleration effect for single sub-image cubes between CPU and GPU

Item	CPU(ms)	GPU(ms)	Speed-up ratio
Sub-image segmentation	35	0.47	74.5
Multi-frame mean	13	0.27	48.1
Correlation alignment	68	4	17
Spectral ratio calculation	83	4	20.75
Bispectrum calculation	205	9	22.8
Other calculations	86	35.26	/
Total time-consuming of single sub-image	490	53	8.1

表 5 给出了由 100 帧 2304 pixel×1984 pixel 像素大小的短曝光子块图像重建出 1 帧 2304 pixel×1984 pixel 像素大小的高分辨力图像的重建时间。可以看出, 使用 GPU 并行化后的算法相比于在 CPU 上运行的串行算法, 耗时由 528 s 下降至 67 s, 加速比可达 7.9 左右。由此可以说明基于 GPU 并行计算的太阳高分辨力斑点重建算法可以有效的缩短程序执行时间, 提高了程序运行效率。

表 5 全部子块 CPU 与单 GPU 重建时间对比

Table 5 Comparison of reconstruction time for all sub-image cubes between CPU and single GPU

CPU(s)	GPU(s)	Speed-up ratio
508	67	7.9

5 结 论

介绍了太阳图像斑点重建算法并行加速国内外研究现状, 针对斑点重建算法耗时长的问题, 在 GPU 上成功实现了斑点重建算法的并行加速。在此基础上, 在线程分配、数据读取、傅里叶变换优化等方面提出了相应的优化策略, 进一步减少了算法的耗时。实验结果表明, 在太阳图像斑点重建中运用 GPU 技术可以极大地提高运算的速度。在未来, 将探索多 GPU 协同计算、CPU-GPU 间协同计算及扩大算法的并行规模, 以期达到实时重建的效果。

参考文献:

- [1] Li Fei. *Phase Diversity Wavefront Sensing and Its Application in Image Restoration*[D]; Changsha: Doctorial Dissertation of Graduate School of National University of Defense Technology, 2011 (in Chinese).
李 斐. 相位差波前探测技术及其在图像恢复中的应用研究 [D]; 长沙: 国防科学技术大学博士论文, 2011.

- [2] Jiang Wenhan. Adaptive optics technology [J]. *Chinese Journal of Nature*, 2006, **28**(1): 7-13(in Chinese).
姜文汉. 自适应光学技术 [J]. 自然杂志, 2006, **28**(1): 7-13.
- [3] Xiang Yongyuan, Liu Zhong, Jin Zhenyu, et al. High resolution solar image reconstruction method [J]. *Progress In Astronomy*, 2016, **34**(1): 94-110 (in Chinese).
向永源, 刘 忠, 金振宇, 等. 高分辨力太阳图像重建方法 [J]. 天文学进展, 2016, **34**(1):94-110.
- [4] Labeyrie A. Attainment of diffraction limited resolution in large telescopes by Fourier analysing speckle patterns in star images [J]. *Astron Astrophys*, 1970, **6**(1): 85-7.
- [5] Weigelt G P. Modified astronomical speckle interferometry "speckle masking" [J]. *Optics Communications*, 1977, **21**(1): 55-59.
- [6] Ding Y Y. Image reconstruction in speckle interferometry [C]. Beijing:SPIE/COS Photonics Asia. *International Society for Optics and Photonics*, 2014: 92733C.
- [7] Li Xuebao. *Research on Technologies of Massive Data Parallel Processing for Solar Telescope* [D]. Yunnan: Doctorial Dissertation of Graduate University of Chinese Academy of Sciences (Yunnan Observatories), 2015 (in Chinese).
李雪宝. 太阳望远镜海量数据并行处理技术研究 [D]. 云南: 中国科学院研究生院 (云南天文台) 博士论文, 2015.
- [8] Xiang Yongyuan. *Research on High Speed High Resolution Solar Image Reconstruction Algorithm* [D]. Yunnan: Doctorial Dissertation of Graduate University of Chinese Academy of Sciences(Yunnan Observatories), 2016 (in Chinese).
向永源. 太阳高分辨高速重建算法的研究 [D]. 云南: 中国科学院研究生院 (云南天文台) 博士论文, 2016.
- [9] Wöger F, Ferayorni A. Accelerated speckle imaging with the ATST visible broadband imager [J]. *Proceedings of SPIE-The International Society for Optical Engineering*, 2012, **8451**(4): 84511C.
- [10] Kolmogorov A N. The Local Structure of Turbulence in Incompressible Viscous Fluid for Very Large Reynolds Numbers [J]. *Proceedings Mathematical & Physical Sciences*, 1991, **434**(1890): 9-13.
- [11] Von der Lühe O. Estimating Fried's parameter from a time series of an arbitrary resolved object imaged through atmospheric turbulence [J]. *Journal of the Optical Society of America A* 1984, **1**(5): 510-519.
- [12] Fried D L. Anisoplanatism in adaptive optics [J]. *Journal of the Optical Society of America*, 1982, **72**(1): 52-61.
- [13] Zhong Libo. *High-resolution reconstruction techniques of the solar images* [D]. Chengdu: Doctorial Dissertation of Graduate University of Chinese Academy of Sciences (Institute of Optics and Electronics), 2015 (in Chinese).
钟立波. 太阳高分辨力图像重建技术研究 [D]. 成都: 中国科学院研究生院 (光电技术研究所) 博士论文, 2015.
- [14] Henry Wong, et al. Demystifying GPU Microarchitecture through Microbenchmarking [C]. New York: *Performance Analysis of Systems & Software(ISPASS), 2010 IEEE International Symposium on. IEEE*, 2010: 235-246.