

并行化叠层成像算法研究

肖俊 李登宇 王雅丽 史祎诗

Ptychographical algorithm of the parallel scheme

Xiao Jun Li Deng-Yu Wang Ya-Li Shi Yi-Shi

引用信息 Citation: *Acta Physica Sinica*, 65, 154203 (2016) DOI: 10.7498/aps.65.154203

在线阅读 View online: <http://dx.doi.org/10.7498/aps.65.154203>

当期内容 View table of contents: <http://wulixb.iphy.ac.cn/CN/Y2016/V65/I15>

您可能感兴趣的其他文章

Articles you may be interested in

多波长同时照明的菲涅耳域非相干叠层衍射成像

Incoherent ptychography in Fresnel domain with simultaneous multi-wavelength illumination

物理学报.2016, 65(12): 124201 <http://dx.doi.org/10.7498/aps.65.124201>

厚样品三维叠层衍射成像的实验研究

Experimental study on three-dimensional ptychography for thick sample

物理学报.2016, 65(1): 014204 <http://dx.doi.org/10.7498/aps.65.014204>

基于背景最佳滤波尺度的红外图像复杂度评价准则

An evaluation criterion of infrared image complexity based on background optimal filter scale

物理学报.2015, 64(23): 234202 <http://dx.doi.org/10.7498/aps.64.234202>

基于叠层衍射的数字水印算法研究

Watermarking algorithm based on ptychographical imaging

物理学报.2015, 64(23): 234201 <http://dx.doi.org/10.7498/aps.64.234201>

可调谐相位板空域频域联合分析

Combined analysis of tunable phase mask within spatial and frequency domain

物理学报.2015, 64(22): 224207 <http://dx.doi.org/10.7498/aps.64.224207>

并行化叠层成像算法研究*

肖俊 李登宇 王雅丽† 史祎诗

(中国科学院大学, 北京 100049)

(2016年3月29日收到; 2016年5月22日收到修改稿)

叠层成像是定量相位恢复技术的重要研究方向, 它通过照明探针的交叠式扫描, 使用叠层迭代相位恢复算法对待测样品进行恢复, 但成像效率与成像质量之间的矛盾等问题已成为其瓶颈之一. 本文从叠层成像迭代恢复算法的基本原理入手, 提出了基于CPU和GPU的两种分块复振幅重建并行算法, 并通过模拟实验研究了不同待测样品尺寸、不同分块、不同孔径数目对并行加速效果的影响. 模拟实验结果表明: 两种并行算法可正确地恢复出样品的复振幅信息, 并且显著提升了重建速度, 使得重建耗时比传统叠层成像算法有了数量级的下降, 在一定程度上解决了成像效率与成像质量之间的矛盾, 有望实现准实时成像, 为叠层成像在相关领域更广泛的应用提供了一定的技术指导. 实验结果同时表明: 在最优分块时, 并行重建加速比与样品的大小有关, 样品越大, 加速效果越明显; 同一个样品在不同分块下重建会得到不同的加速比, 这与硬件设备密切相关, 而成像中孔径的数目不会对并行加速比产生明显的影响.

关键词: 叠层成像, 相位恢复, 并行计算**PACS:** 42.30.-d, 42.25.Fx, 42.30.Rx, 42.30.Kq**DOI:** 10.7498/aps.65.154203

1 引言

叠层成像 (ptychography) 的概念是 Hoppe 在 1968 年至 1973 年提出, 经 Fienup 和 Misell 的改进逐渐发展而来^[1,2]. 为了研究晶体结构, Hoppe 提出了基于叠层成像原理的扫描相干衍射成像方法, 并通过研究晶体和非晶体的扫描透射电子衍射显微成像, 验证了此方法的有效性. Rodenburg 和 Faulkner^[3-6] 结合相位恢复算法将此方法进行了多次改进, 并在电子显微镜以及生物等领域得到了应用. 随着研究和应用的深入, 叠层成像的成像效率与成像质量之间的矛盾等问题日益突出, 这是因为叠层成像迭代算法要耗费大量的时间完成迭代过程, 并且随着样品采样点的增多, 重建时间呈非线性增长; 此外, 采样点的增多对处理器和存储系统等也提出了更高的要求, 而减少迭代次数, 则会造成较大的成像误差.

针对这些问题, 本文在分析叠层成像迭代算

法基本原理的基础上, 提出了基于CPU和GPU的两种分块并行重建方案, 并在多核CPU和支持CUDA的GPU上对两种方案进行实现. 此外, 本文在第四节通过大量实验, 分析了可能影响并行性能的几种因素. 研究的结果对实现准实时叠层成像有一定的指导作用.

2 叠层成像迭代算法的基本原理

叠层成像是通过移动一个全透的小孔使入射平面波照射到样品的不同位置, 如图 1(a) 所示, 记录由此得到的一系列衍射强度图样, 并根据这些图样重构出样品的振幅与相位信息^[7-10]. 这一过程的关键在于每次照射样品的一个“层”都要和与其相邻的至少一个“层”发生交叠, 如图 1(b) 所示. 重构算法在分别重建每“层”的复振幅时要同时满足其他“层”的衍射分布约束, 因此最后样品的整体复振幅信息是所有“层”的共解.

* 国家重点基础研究发展计划 (批准号: 2014CB931900)、国家自然科学基金 (批准号: 61350014, 61307018, 61471338)、中国科学院青年创新促进会 (批准号: 2015361) 和中国科学院大学校长基金、中国科学院“科教结合”教育创新项目.

† 通信作者. E-mail: wangyali2003@163.com

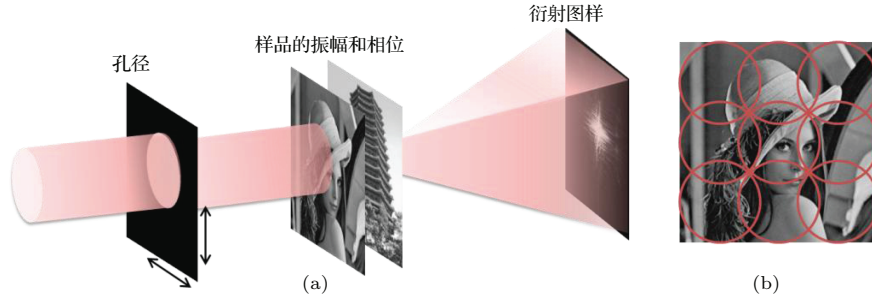


图1 叠层成像基本原理 (a) 叠层成像光路; (b) “层”交叠位置

Fig. 1. The principle of ptychography: (a) Schematic of the experimental setup; (b) circles indicate beam positions of overlap.

相比于其他技术, 叠层成像算法不但具有鲁棒性强、收敛速度快等特点, 而且还可以消除共轭解的问题.

在迭代算法中, 设 $P(\mathbf{r})$ 为照明函数, 物体的复振幅分布函数为 $O(\mathbf{r})$, 则透过物体的出射波函数为

$$\Psi_{n,i}(\mathbf{r}) = P(\mathbf{r})O_n(\mathbf{r} + \mathbf{R}_i), \quad (1)$$

其中 n 为当前迭代次数, i 为第 i 个小孔的位置, \mathbf{r} 是第一个孔径的物面坐标矢量, \mathbf{R}_i 表示第 i 个位置相对于第一个位置移动的矢量. 第一次迭代时, 物体的复振幅分布函数随机分布. 经过快速傅里叶变换得到出射波在远场的振幅和相位分布为

$$\Psi'_{n,i} = |A_{n,i}| \exp(i\Phi_{n,i}) = \text{FFT} \{ \Psi_{n,i} \}. \quad (2)$$

用实际记录下的第 i 个位置处的远场光强的平方根替代计算所得的振幅,

$$|\Psi'_{n,i,\text{new}}(\mathbf{r})| = \sqrt{I_i}, \quad (3)$$

$$\Psi'_{n,i,\text{new}}(\mathbf{r}) = \sqrt{I_i} \exp(i\Phi_{n,i}), \quad (4)$$

经过逆快速傅里叶变换可以得到一个新的输出波函数, 并据此更新物函数

$$\Psi_{n,i,\text{new}}(\mathbf{r}) = \text{FFT}^{-1} \{ \Psi'_{n,i,\text{new}}(\mathbf{r}) \}, \quad (5)$$

$$O_{n+1}(\mathbf{r} + \mathbf{R}_i) = O_n(\mathbf{r} + \mathbf{R}_i) + \beta U(\mathbf{r}) [\Psi_{n,i,\text{new}}(\mathbf{r}) - \Psi_{n,i}(\mathbf{r})], \quad (6)$$

其中 α, β 为调节因子, α 为了防止分母为 0, β 是反馈参数.

更新后的物函数作为新的猜测值进入下一次迭代.

叠层成像算法每次用一个已知的照明光束乘以一个更新后的物函数. 由于相邻位置的照明光束之间都是有交叠的, 因此经过多次迭代即可求得所有“层”的共解, 从而得到原始的物体复振幅分布.

3 叠层成像迭代算法的并行方案设计

3.1 并行方案分析与设计

高质量叠层干涉成像技术在移动全透小孔时, 会产生一系列的“层”, 这些“层”必须至少和其他一个“层”发生交叠, 而相互交叠的“层”之间对于重构对方“层”的复振幅有一定的约束, 而如果没有交叠, 则不会产生影响. 基于这一特点, 可以考虑两种并行方式:

方式一, 以层为单位的并行重建;

方式二, 先对层进行分块, 然后基于分块的并行重建.

第一种方式中, 根据图 2 中的结构按照小孔的位置对层进行编号, 第 l 行第 j 列的层编号记为 (l, j) . 由图 2 可知, 对编号为 $(1, 1)$ 的层, 只有编号为 $(1, 2)$ $(1, 3)$ $(2, 1)$ $(2, 2)$ $(2, 3)$ $(3, 1)$ $(3, 2)$ $(3, 3)$ 的层与之有联系, 虽然层 $(3, 3)$ 可能并没有直接与层 $(1, 1)$ 重叠, 但在重构层的算法中, 是以矩形块为单位的, 因此也将其纳入对 $(1, 1)$ 层的重构中. 这样对于任意编号的层, 都可以找到与其重构相关的层, 因此每个层的复振幅重构对其他层的重构没有影响, 进而相互之间可以并行处理. 但此方式中, 每个层都需要花费大量的时间来独自重建, 并且重构出每个层后, 还需要利用较长的时间来处理层与层之间的交叠部分, 因此会导致重建效率和重建质量的降低.

第二种方式则可以避免第一种方式中存在的问题. 为了提高重建速度, 同时减少边界的处理, 并提升恢复效果, 在第二种方式中采用对样本采样点进行分块的策略, 如图 3 所示. 因为边界上存在没有两个“层”同时覆盖到的区域, 因此有效的重建区域如图中大矩形所示. 由图 3 可知, 该方式将有效重建区域按矩形块划分, 相邻的两个矩形块之间

有很小的重叠区域. 重建时, 首先找出覆盖每个矩形块的所有“层”, 然后以这些“层”的相互约束迭代重构出每个分块的复振幅信息, 最后在所有分块重建完毕后, 对相邻分块之间的重叠区域进行均值操作, 以平滑分块间的边缘. 因此, 方式二不仅可以减少边界处理的时间, 还可以实现更充分的并行加速. 此外, 在方式二的实现过程中, 还可以根据样品采样点的多少和硬件的性能调节分块的个数, 以得到最优的加速效果. 因此, 本文基于第二种方式设计具体的并行方案.



图2 叠层成像算法中各层示意图

Fig. 2. Hierarchical diagram in ptychographical algorithm.

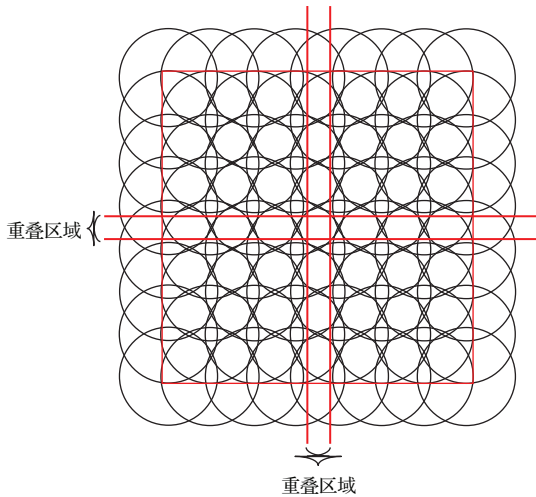


图3 分块重建示意图

Fig. 3. Schematic diagram of block reconstruction.

3.2 基于CPU的并行方案设计

基于CPU的并行主要分为多CPU和多核CPU两种方式. 多CPU并行方式可以较大地提升程序性能, 但其并行环境搭建复杂, 且不利于使

用与移植. 而在多核CPU的并行方式中, 并行性能提升幅度与CPU的核数密切相关, 随着CPU性能的不断提升, 计算核心数目不断增加, 基于多核CPU的并行也能显著提升性能, 同时便于移植, 且并行模块之间数据交换方便而快速. 因此本文采用多核CPU的并行方式. OpenMP是基于多核CPU的并行编程模型, 它面向共享内存, 提供了对并行算法的高层抽象描述. 因此本文使用OpenMP^[11]完成任务并行的实现, 同时采用FFTW库完成快速傅里叶变换与反变换^[12-14]操作. 并行方案流程图如图4所示, 其中 n 是当前迭代的次数, num 是CPU的核心数目, $obj(k)$ ($k = 1, 2, 3, \dots$)表示重建的第 k 个分块的复振幅信息, 每次迭代过程都会更新 $obj(k)$ 的值.

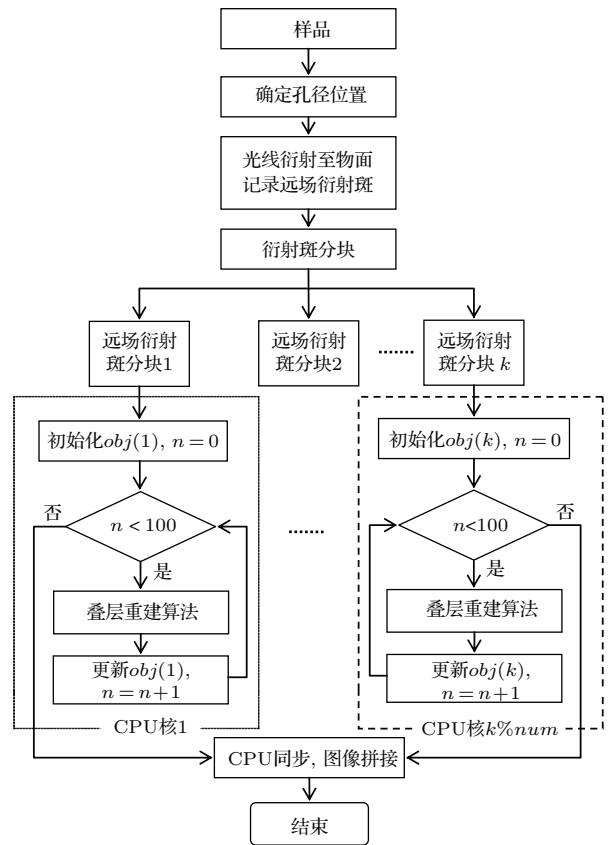


图4 CPU并行重建流程图

Fig. 4. Flow chart of parallel reconstruction based on CPU.

根据图4可知, 并行重建的具体步骤如下.

1) 衍射图样获取: 根据参数确定每个孔径的位置, 将每个孔径衍射至物面, 记录远场衍射斑.

2) 衍射图样分块: 根据实际需要对衍射图样进行分块, 分块大小可以调节, 但需要确保相邻的两个块的有效重建区域有重叠.

3) 迭代重建: 用 OpenMP 来实现每个分块的并行, 其中每个分块分别调用叠层重建算法进行计算, 即把每个块的重建任务交给一个 CPU 计算核心, 如果计算核心的数目小于分块数目, 则每个 CPU 核心完成一个块的重建后, 继续重建未重建的块, 直到所有的图像块全部重建完成.

4) 图像拼接: 对 CPU 计算线程进行同步操作, 即所有重建线程运行完成后, 通过对相邻两块图像的重叠部分进行均值操作来实现图像的拼接.

3.3 基于 GPU 的并行方案设计

图形处理单元 (GPU) 包含大量的计算核心, 拥有强大的计算能力, 其通用计算技术已经在科学计算、并行计算方面得到了广泛应用. 由 NVIDIA 推出的通用并行计算架构 CUDA, 使 GPU 能够解决复杂的计算问题^[12], 且其开发技术成熟而简单. 因此本文采用基于 NVIDIA GPU 的 CUDA 标准, 将大量的傅里叶变换用 cuFFT 提供的批处理功能在 GPU 上完成^[15,16], 实现并行重建, 其并行方案的流程图如图 5 所示, 其中 n 是当前迭代的次数, m 为孔径的个数, $obj(k)$ ($k = 1, 2, 3, \dots$) 表示第 k 个分块重建的复振幅信息, 每次迭代过程都会更新 $obj(k)$ 的值.

根据流程图可知, 重建的具体步骤包括:

1) 衍射图样获取 根据参数确定每个孔径的位置, 将每个孔径衍射至物面, 记录远场衍射斑;

2) 衍射图样分块 根据实际需要衍射图样进行分块, 分块大小可以调节, 原则上任意大小的分块都可以, 但需要确保相邻的两个块的有效重建区域有重叠;

3) 拷贝衍射图样数据至显存 为了提高性能, 充分利用 GPU 计算资源, 采用任务级别并发, 任务内部并行的策略, 即每个块的不同孔径处的重建交替执行, 而孔径内部样品点的复振幅重建并行执行, 这个过程需要使用 cuFFT 的批处理功能, 以实现对不同的块的并发重建, 因此需要将每个分块的不同位置孔径处的衍射图样拷贝到连续的显存中, 如图 6 所示, 第一个孔径处, 第一块衍射图样的起始地址是 x 长度是 $length$, 则第二块相同位置孔径处的衍射图样的起始地址为 $x+length$, 其他孔径处每块的衍射图样存储也是类似的, 然后将所有的孔径对应的所有块的衍射图样数据拷贝到 GPU 显存中;

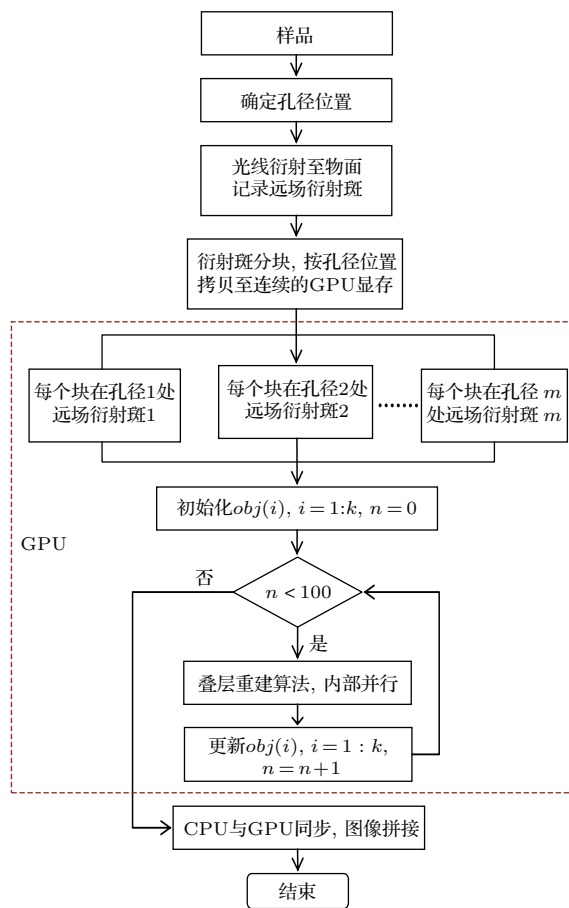


图 5 GPU 分块重建流程图

Fig. 5. Flow chart of block reconstruction based on GPU.

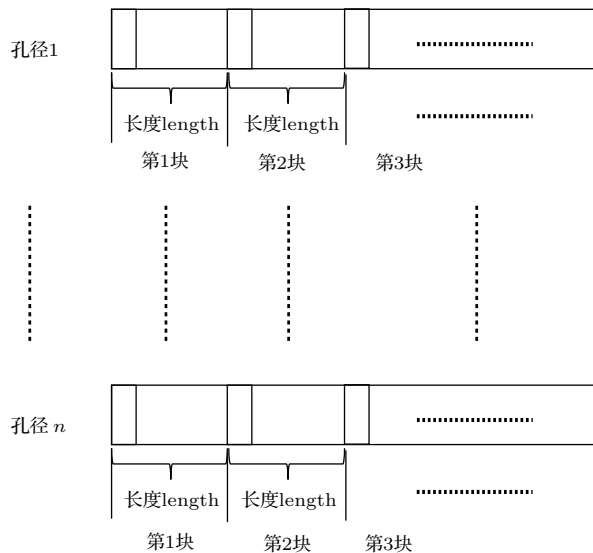


图 6 远场衍射斑占用的内存区域

Fig. 6. The memory occupied by the far field diffraction spot.

4) 迭代重建 用叠层成像迭代算法重建每个孔径处所有块的复振幅信息, 直到所有块重建完毕, 然后将重建后的数据从 GPU 显存拷贝回 CPU

内存, 开始每一块的拼接, 最终完成整个样品的重建工作.

4 实验结果与分析

本文对提出的基于CPU和GPU的叠层成像重建方案做了模拟实验并进行了结果分析. 在基于CPU的并行方案实验中采用的硬件配置为: CPU为Intel(R) Core(TM) i5-2400, 其主频为3.10 GHz, 拥有4个核; 内存为4.00 GB. 在基于GPU的并行方案实验中采用的GPU为NVIDIA Tesla C2070, 其CUDA核心数448; CUDA核心频率为1.15 GHz; 显存6 GB. 实验中采用的操作系统为32位的Windows 7, 而开发环境选择的是vs2010.

4.1 样品采样点数与加速比的关系

本文模拟了在光束尺寸相同、交叠率相同并且分块个数相同时, 不同样品采样点数对重建时间的影响. 实验中对样本的多种不同采样点数的重建过程进行了多次实验, 限于篇幅, 本文仅给出采样点数为128 pixels × 128 pixels, 256 pixels × 256 pixels, 512 pixels × 512 pixels以及1024 pixels × 1024 pixels时的实验结果, 其运行耗时如表1所列. 表1中列出的时间是调整分块参数得到的最优运行时间, 其最优分块的共同特点是分块后每块的采样点的个数都是64 pixels × 64 pixels左右, 即128 pixels × 128 pixels的样品分成2 × 2块, 256 pixels × 256 pixels的样品分成4 × 4块, 以此类推.

由表1可知: 两种并行方案均获得了良好的加速效果. 其中, 基于OpenMP的并行方案在4核处理器上的加速比可以达到6倍, 而基于CUDA的并行加速比可以高达18倍左右, 重建花费的时间有了数量级的减少. 此外, 分析表1中数据可发现: 随着采样点成倍增大, 基于OpenMP的加速比也在增大, 且主要分布在4—6之间. 这是因为算法中要

做大量的傅里叶变换和反变换, 而快速傅里叶变换时间复杂度为 $n \log n$, 对图像做傅里叶变换的过程是水平方向每一行分别做傅里叶变换, 之后对每一列分别做傅里叶变换, 因此图像的傅里叶变换时间复杂度是 $n^2 \log n$. 假设对样品的采样点大小为 $aN \text{ pixels} \times aN \text{ pixels}$, 将它分成 $a \times a$ 个大小为 $N \text{ pixels} \times N \text{ pixels}$ 的块, 由于在实验中采用的处理器是4核的, 因此每个核要完成 $a \times a/4$ 个块的重建任务, 而核与核之间并行执行, 因此加速比应为

$$\frac{a^2 N^2 \log(aN)}{(N^2 \log N) a^2 / 4} = 4(\log a / \log N + 1). \quad (7)$$

根据(7)式, 将采样点为128 pixels × 128 pixels的样品数据分成大小为64 pixels × 64 pixels的块后, $a = 2, N = 64$, 其理想的加速比为

$$4(\log 2 / \log 64 + 1) = 4.67. \quad (8)$$

同理可得其他三种采样点数成像时的理想加速比, 如图7所示. 由图7可知: 随着样品采样点数的增加, 理想加速比与实际加速比同时增大, 而实

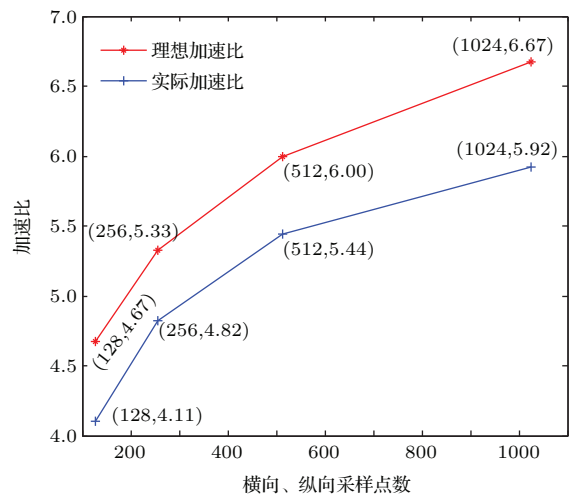


图7 OpenMP方案中不同样品采样点数对应的理想与实际加速比

Fig. 7. The speedup ratio of ideal and actual situations under different sampling points in the OpenMP scheme.

表1 样品在不同采样点数重建时间

Table 1. Reconstruction time of the sample under different sampling points.

样品大小 pixels × pixels	不并行 时间/s	OpenMP 并行		CUDA 并行	
		时间/s	加速比	时间/s	加速比
128 × 128	7.114	1.732	4.107	1.235	5.760
256 × 256	31.428	6.52	4.820	2.927	10.737
512 × 512	140.759	25.865	5.442	9.207	15.288
1024 × 1024	624.641	105.471	5.922	33.164	18.835

实际加速比之所以略小于理想加速比是因为程序中存在不能并行的部分, 此外, 线程同步也会耗费一定的时间. 因此, 本文设计的并行方案的理论结果与实际结果一致, 并且可以适应不同的样品大小, 样品越大, 采样点越多, 加速效果越明显.

4.2 分块数对加速效果的影响

本文模拟了在光束尺寸相同、交叠率相同时, 不同的采样点数在不同分块下的时间消耗. 实验中选取了多种不同大小的样品采样点数, 都获得了相同的结论, 限于篇幅, 本文仅给出以尺寸为 128 pixels × 128 pixels, 256 pixels × 256 pixels, 512 pixels × 512 pixels 以及 1024 pixels × 1024 pixels 的采样点数的实验结果, 分块大小选为 2 × 2, 4 × 4 等. 两种并行方案的运行耗时分别如表 2 和表 3 所列. 其中“分块太小”表示按指定的分块数目分块后, 每个分块太小, 包含的“层”太少, 导致重建的误差太大, 而不能重建出样品的原始信息.

在基于 OpenMP 的并行方案中, 随着分块的增加, 加速比也随着增加, 当分块大小达到 64 × 64 时, 性能基本达到最大并趋于稳定, 这是因为对于采样点数 $N \times N$ 的样品数据, 若将其分为 $a \times a$ 块, 则每块的大小为 N/a , 此时 4 核 CPU 的理想加速比为

$$\frac{N^2 \log N}{\left(\frac{N}{a}\right)^2 \left(\log \frac{N}{a}\right) \frac{a^2}{4}} = \frac{4}{1 - \log a / \log N}. \quad (9)$$

根据 (9) 式, 以 1024 pixels × 1024 pixels 的采样点数为例, 不同分块对应的理想加速比与实际加速比如图 8 所示, 可以看出当分块小于 12 × 12 时, 实际加速比与理想加速比相差较大, 但整体趋势是一致的. 这是由 CPU 的缓存机理导致的: 在程序运行中, 操作系统会根据局部性原理, 将经常访问的数据保存在缓存中, 可以加速数据的读写, 从而提高程序性能, 但缓存有一定的大小限制, 在叠层成像迭代算法中, 每次迭代都会用到一些相同的数据, 操作系统同样会加载这些数据到缓存中. 此时根据分块大小的不同, 会出现两种情况:

- 1) 当分块较少时, 每个分块的采样点相对较多, 占用的存储较大, 使操作系统不能将所有的计算数据同时加载进缓存, 因此存在频繁的缓存与内存数据交换, 进而影响程序性能;
- 2) 当分块较多时, 计算中每个分块的大小相对变小, 因此可以加载进更多的计算数据进入缓存, 直到分块的大小使得所有的计算数据都能加载入缓存时, 性能达到最优.

针对本文的实验环境, 经测试, 当分块的大小在 64 pixels × 64 pixels 左右时, 加速效果基本接近理想状态.

在基于 CUDA 的并行方案中, 运行时间基本不会受分块的影响, 这是因为 GPU 有大量的计算核心, 傅里叶变换对程序的影响相对减小, 而其他运算部分由于要大量的访问全局显存, 耗时比重增大, 同时由于相邻分块有重叠, 带来了额外的计算开销, 因此分块重建时间没有明显的变化.

表 2 OpenMP 不同分块运行时间 (s)
Table 2. Run time of different blocks in the OpenMP scheme (s).

样品采样大小 pixels × pixels	分块大小					
	不分块	2 × 2	4 × 4	8 × 8	12 × 12	16 × 16
128 × 128	7.114	1.84	1.731	分块太小	分块太小	分块太小
256 × 256	31.428	11.31	6.67	6.52	分块太小	分块太小
512 × 512	140.759	64.36	44.74	25.865	26.208	分块太小
1024 × 1024	624.641	284.01	248.25	174.819	105.471	102.648

表 3 CUDA 不同分块运行时间 (s)
Table 3. Run time of different blocks in the CUDA scheme (s).

样品采样大小 pixels × pixels	分块大小					
	不分块	2 × 2	4 × 4	8 × 8	12 × 12	16 × 16
128 × 128	1.313	1.235	1.324	分块太小	分块太小	分块太小
256 × 256	3.012	2.927	2.931	3.212	分块太小	分块太小
512 × 512	10.20	10.48	11.21	10.634	10.441	分块太小
1024 × 1024	35.21	34.93	33.82	33.348	33.164	33.266

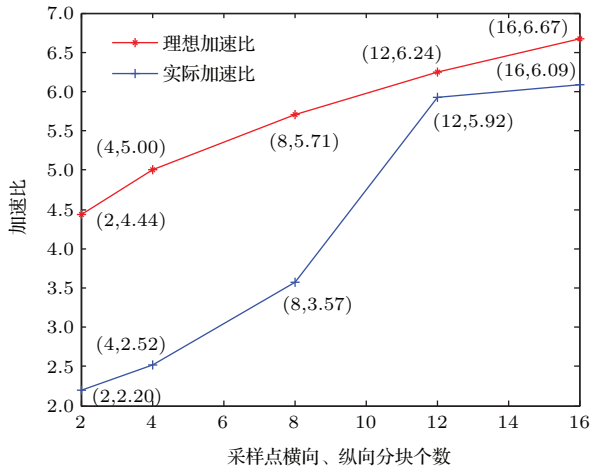


图8 OpenMP方案中1024×1024图片在不同分块下的加速比
Fig. 8. The speedup ratio of image with the size of 1024×1024 under different blocks in the OpenMP scheme.

4.3 光束尺寸和交叠率对加速效果的影响

光束尺寸和交叠率的不同主要影响的是孔径的个数. 光束尺寸越小, 要覆盖整个样品就需要更多的光束; 如果光束之间的交叠率越高, 那么同样需要更多的光束. 因此光束尺寸和交叠率的不同主要反映在重建时孔径的个数上. 本文通过实验模拟

了样品采样点固定为256 pixels × 256 pixels, 孔径个数分别为3 × 3, 4 × 4, 5 × 5和6 × 6时的并行加速效果, 如表4所列.

表4 不同个数孔径下重建时间(s)
Table 4. Reconstruction time with different number of apertures (s).

并行方式	3 × 3	4 × 4	5 × 5	6 × 6
不并行	31.428	49.998	78.14	105.507
OpenMP	6.67	9.968	15.336	20.049
CUDA	2.927	4.488	6.625	8.969

由于孔径的个数直接影响迭代中内层循环的次数, 而每次迭代的时间与内层循环的次数成正比. 因此不论并行或不并行, 时间都应该与孔径的个数成正比, 表4中的数据与此相符, 因此并行算法加速效果不受光束尺寸和交叠率的影响.

4.4 并行重建方案对成像质量的影响

本文模拟了采样点数为256 pixels × 256 pixels, 分块数为2 × 2的情况下, 并行重建方案对成像质量的影响. 实验选用振幅相关系数 C_a 和相位相关系数 C_ϕ 以及振幅均方差 MSE_a 和相位均方差 MSE_ϕ 来评价成像的质量. 图9给出了并行方案中,

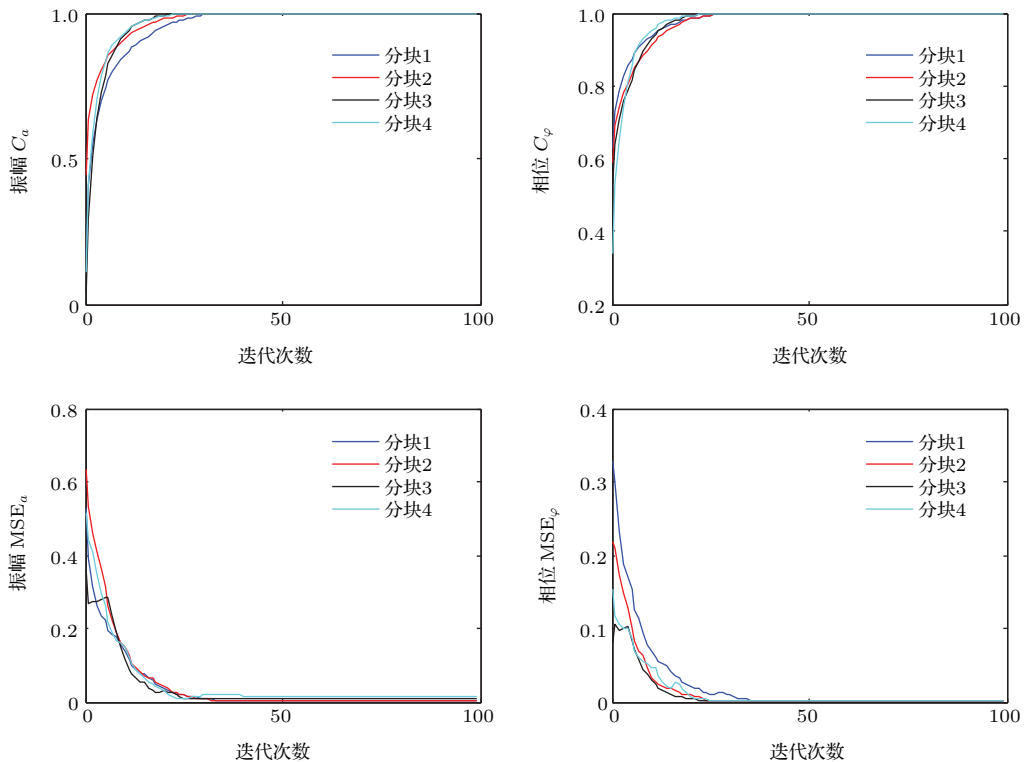


图9 分块重建时振幅与相位的均方差和相关系数

Fig. 9. Mean square deviation and correlation coefficient of amplitude and phase in block reconstruction.

每个分块在迭代重建过程中每步的相关系数和均方差的变化情况. 由图9可知, 每个分块都能很快收敛, 且最终成像结果与原始样品的相关性接近1, 而均方差接近0, 说明成像质量很好.

最终合成后的均方差和相关系数如表5所列. 由表5可知: 并行重建对成像质量几乎没有影响, 这也说明了本文并行重建方案的合理性和正确性.

表5 分块重建的振幅与相位的相关系数和均方差

Table 5. Correlation coefficient and mean square deviation of amplitude and phase in block reconstruction.

C_a	C_ϕ	MSE_a	MSE_ϕ
0.9959	0.989911	0.00338	0.00077

5 结 论

本文结合叠层成像迭代恢复算法, 从并行加速的角度出发, 提出了分块重建方案, 并且系统地分析了不同样品采样点数、不同分块、光束尺寸和交叠率对加速效果的影响. 模拟实验结果表明, 分块重建方案可以正确地恢复出样品的复振幅信息, 重建出的振幅和相位与原始振幅相位相关性接近于1, 而均方误差接近于0, 证明了分块重建方案的正确性. 从实验数据可以看出, 分块重建并行加速方案可以明显提升加速性能, 采样点数越多, 得到的加速比也越高, 而孔径数目的不同, 基本不会影响加速比, 算法性能稳定. 此外, 研究和分析也表明分块的大小对基于多核CPU的并行方案性能提升影响较大, 即最优分块和硬件设备的计算性能以及存储系统密切相关, 要取得最佳的加速效果, 需要实验测出硬件对应的最优的分块大小. 总之, 本文提出的叠层成像迭代算法的分块并行方案在多核CPU和GPU上都能得到明显的时间性能提升, 尤其在GPU上可以得到18倍的加速比, 使得迭代算法的重建耗时有了数量级的减少. 可以预见, 如果将分块并行策略应用于多CPU或者多GPU, 重建性能会有更大幅度的提升, 而当CPU或GPU的

数目足够多时, 有望实现准确实时的完成任意大小样品的成像任务.

参考文献

- [1] Hoppe W 1969 *Acta Cryst. A* **25** 495
- [2] Hoppe W 1969 *Acta Cryst. A* **25** 508
- [3] Rodenburg J M, Faulkner H M L 2004 *Appl. Phys. Lett.* **85** 4795
- [4] Faulkner H M L, Rodenburg J M 2004 *Phys. Rev. Lett.* **93** 023903
- [5] Maiden A M, Rodenburg J M 2009 *Ultramicroscopy* **109** 1256
- [6] Humphry M J, Kraus B, Hurst A C, Maiden A M, Rodenburg J M 2012 *Nature Commun.* **3** 730
- [7] Wang Y L, Shi Y S, Li T, Gao Q K, Xiao J, Zhang S G 2013 *Acta Phys. Sin.* **62** 064206 (in Chinese) [王雅丽, 史玮诗, 李拓, 高乾坤, 肖俊, 张三国 2013 物理学报 **62** 064206]
- [8] Shi Y S, Wang Y L, Zhang S G 2013 *Chin. Phys. Lett.* **30** 054203
- [9] Shi Y S, Li T, Wang Y L, Gao Q K, Zhang S G, Li H S 2013 *Opt. Lett.* **38** 1425
- [10] Shi Y S, Wang Y L, Li T, Gao Q K, Wan H, Zhang S G, Wu Z B 2013 *Chin. Phys. Lett.* **30** 074203
- [11] Cai J J, Li M S, Zheng F 2007 *Computer Technology and Development* **10** 87 (in Chinese) [蔡佳佳, 李名世, 郑锋 2007 计算机技术与发展 **10** 87]
- [12] Hu X Y, Cao X L, Guo H, Chen J 2012 *Chin. J. Comput. Phys.* **29** 522 (in Chinese) [胡晓燕, 曹小林, 郭红, 陈军 2012 计算物理 **29** 522]
- [13] Frigo M, Johnson S G 1998 *Acoustics Speech and Signal Processing. Proceedings of the 1998 IEEE International Conference Seattle WA, May 12-15, 1998* p1381
- [14] Dong L, Ge W C, Chen K L 2010 *Information Technology* **4** 11 (in Chinese) [董萃, 葛万成, 陈康力 2010 信息技术 **4** 11]
- [15] Yang X, Li X Y, Li J G, Ma J, Zhang L, Yang J, Du Q Y 2014 *Spectroscopy and Spectral Analysis* **34** 498 (in Chinese) [杨雪, 李学友, 李家国, 马骏, 张力, 杨健, 杜全叶 2014 光谱学与光谱分析 **34** 498]
- [16] Li S, Wang J, Sun H 2008 *Optics and Precision Engineering* **16** 2414 (in Chinese) [李仕, 王晶, 孙辉 2008 光学精密工程 **16** 2414]

Ptychographical algorithm of the parallel scheme*

Xiao Jun Li Deng-Yu Wang Ya-Li[†] Shi Yi-Shi

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

(Received 29 March 2016; revised manuscript received 22 May 2016)

Abstract

Ptychography is an important technique in the quantitative phase imaging research domain, which employs the illuminating probes to scan the specimen in an overlapped requirement, and the reconstruction is conducted by using the ptychographic iterative engine. But the contradiction between the imaging efficiency and quality has become a bottleneck for its wide applications. In this paper, we start with the fundamental principle of the iterative algorithms for ptychographical imaging, and propose two parallel schemes based on CPU and GPU, besides the influences of the specimen size, the number of blocks and illuminating beams on the speedup of the two schemes are investigated via simulation experiment. The result shows that the complex amplitude of the specimen can be correctly reconstructed, meanwhile, the speed is significantly improved, which reduces the time consumed by one order of magnitude. This improvement solves the above contradiction, so that we can expect to achieve quasi-real-time imaging. The experimental data also indicate that 1) in optimal partition, parallel speedup is related to the size of the specimen, bigger size is corresponding to more obvious acceleration; 2) the same specimen under different partitions will speed up to different extents, which is closely related to the experimental hardware, however the number of illuminating beams has no significant effect on the speedup.

Keywords: ptychography, phase retrieval, parallel computing

PACS: 42.30.-d, 42.25.Fx, 42.30.Rx, 42.30.Kq

DOI: 10.7498/aps.65.154203

* Project supported by the National Basic Research Program of China (Grant No. 2014CB931900), the National Natural Science Foundation of China (Grant Nos. 61350014, 61307018, 61471338), the Youth Innovation Promotion Association of Chinese Academy of Sciences (Grant No. 2015361), the President Foundation of University of Chinese Academy of Sciences, the Fusion Foundation of Research and Education of Chinese Academy of Sciences, China.

[†] Corresponding author. E-mail: wangyali2003@163.com