

文章编号: 1005-5630(2025)01-0032-09

DOI: 10.3969/j.issn.1005-5630.202312190136

异构加速绝对平面检测

李雨杭¹, 韩 森^{1,2}, 李雪园², 徐春风¹, 龚晨曦^{1,2}

(1. 上海理工大学 光电信息与计算机学院, 上海 200093;

2. 苏州慧利仪器有限责任公司, 江苏 苏州 215000)

摘要: 光学干涉检测领域的不断发展要求检测仪器具备更高的横向分辨率。高分辨率意味着处理时间变长, 测试效率变低。为提高测试效率, 提出了一种利用 CPU/GPU 异构计算并行加速的 Zernike 多项式绝对平面检测方法。该方法使用 CPU 进行流程控制, 利用 GPU 多核优势将检测平面中的元素离散并行求解, 并在 Zernike 系数求解中使用混合精度, 在峰谷值和均方根值求解中使用线程束原语指令进一步优化性能。结果显示, 使用 RTX3070-Laptop, 在 512×512、1024×1024、2048×2048 和 4096×4096 分辨率的光学平面检测中, 该方法整体处理速度比 CPU 处理速度分别提高了近 47、56、58 和 70 倍。

关键词: 绝对平面检测; 异构并行计算; Zernike 多项式拟合

中图分类号: TN 206 **文献标志码:** A

Heterogeneous acceleration absolute flatness testing

LI Yuhang¹, HAN Sen^{1,2}, LI Xueyuan², XU Chunfeng¹, GONG Chenxi^{1,2}

(1. School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China;

2. Suzhou H & L Instruments LLC., Suzhou 215000, China)

Abstract: The growing field of optical interferometry requires instruments with higher lateral resolution. High-resolution implies long processing time, which significantly affects testing efficiency. In order to improve the testing efficiency, a Zernike polynomial absolute flatness testing utilizing parallel acceleration of CPU/GPU heterogeneous computing was proposed, which used CPU for process to control and GPU multi-core advantage to discretize the elements in the flatness for parallel solving. Performance was further optimized using mixed-precision in Zernike coefficient solving and peak-to-valley (PV) and root mean square (RMS) were solved with warp-level primitive instructions. Using 512×512, 1024×1024, 2048×2048, and 4096×4096 optically flat surfaces on the RTX3070-Laptop hardware, the speed increased by 47, 56, 58, and 70 times respectively in the overall process compared to the CPU version.

Keywords: absolute flatness testing; heterogeneous parallel computing; Zernike polynomial fitting

收稿日期: 2023-12-19

基金项目: 国家重点研发计划 (2022YFF0607704)

第一作者: 李雨杭 (1996—), 男, 硕士研究生, 研究方向为干涉测量与高性能计算。

E-mail: pcdack@hotmail.com

通信作者: 韩 森 (1961—), 男, 教授, 研究方向为干涉测量。E-mail: senhanemail@126.com

引 言

光学平面是光学系统和仪器工程中最常见的光学元件。在光学干涉测量中, 平面波前像差标准是其他各种测试的基础。标准参考面的形状会影响传统相对测量方法的精度。当需要较高的测量精度(如精度高于 $1/20$ 波长)时, 必须消除干涉仪的系统误差和参考面误差。为了解决这个问题, 提出了绝对平面测量。绝对平面测量是通过消除光学仪器的系统误差和参考平面误差对测量结果的影响, 来获得测试平面的绝对表面的方法。目前, 使用最广泛的方法是由 Fritz^[1] 于 1984 年提出的, 是一种基于传统三面互检的可编程三面互检法。首先, 利用 Zernike 多项式性质将平面的表面形状误差分解为正交基函数; 然后, 再利用最小二乘法对这些基函数的系数进行拟合; 最终, 得到被测绝对平面。许多研究人员在此方法的基础上进行了改进研究^[2], 并提出了许多新方法^[3-5]。但由于计算复杂或条件苛刻, 其中大多数方法并未在工业领域得到广泛应用^[6]。随着测量技术的发展, 人们对大型精密光学测量仪器的需求更加迫切, 如美国国家点火装置以及中国神光-III 装置^[7] 均需要数千块超大口径光学器件。这些超大口径光学器件对干涉仪的横向分辨率提出了更高的要求, 然而现有算法的测量效率尚不能满足这一要求。因此, 本研究采用了图形处理单元(graphics processing unit, GPU)加速技术, 旨在提升测量效率。在干涉测量仪器中获取到干涉图像后, 先提取相位, 然后对提取的相位进行解包裹, 最后使用 Zernike 多项式拟合的方法实现绝对平面测量。在此流程中, 针对干涉图的异构并行相位提取和相位解包裹已有诸多学者进行了探索和研究^[8-9]。但是, 对 Zernike 多项式拟合的绝对平面测量步骤进行探索的还比较少。在 Zernike 多项式的应用方面, 大多数研究人员关注的是 Zernike 多项式的 GPU 并行实现及其在图像处理中的应用^[10]。例如, Toharia 等^[11] 将 Zernike 多项式应用于射击边界检测, 并通过多 GPU 多中央处理单元(central processing unit, CPU) 异构计算实现了超过 1 000 帧/s 的检测

率。然而, 针对光学波前像差拟合异构并行技术, 测量参数求解的并行化和分辨率优化的研究较少。

本文旨在解决传统绝对平面检测 Zernike 多项式计算算法在处理高分辨率干涉图时效率低下的问题。为了提高计算效率, 提出了一种 CPU-GPU 异构并行加速算法。该算法通过在 CPU 和 GPU 之间分配计算任务, 优化了笛卡尔坐标系下的 Zernike 多项式计算、多项式拟合系数求解、理想平面拟合以及测量指标 [如峰谷(peak-to-valley, PV) 值和均方根(root mean square, RMS) 值] 计算等关键步骤。此外, 本文还采用协作组线程束原语指令和混合精度方法, 进一步提升了算法的执行效率。

1 并行算法设计

1.1 异构并行计算

传统的科学计算性能提升依赖于 CPU 的时钟频率。然而, 芯片上高密度晶体管的功耗和散热所造成的物理瓶颈极大地延缓了 CPU 频率的提升^[12]。与 CPU 不同, GPU 自诞生之日起就被用于计算与图像相关的数据。

从图 1 可知, GPU 包含比 CPU 更多的核心数量, 因此, GPU 比 CPU 更适合处理大规模数据的并行计算, 但它的核心频率、缓存和控制单元较弱, 对复杂问题处理能力较弱。为了充分发挥两者优势, 有人提出了异构并行计算方式。异构并行计算就是将算法分解为复杂流程控制部分和简单并行处理部分, 由 CPU 处理复杂逻辑运算部分, GPU 处理简单高吞吐运算部分, 从而达到高性能的诉求。在科学计算领域, 越来越多的学者采用异构并行计算来解决相关领域问题: 赵锴坤等^[13] 将 CPU/GPU 异构并行计算用于物理领域, 计算加速重力场反演, 将计算性能提升了 380 倍; 李浩^[14] 将异构并行计算用于天文数据处理, 在进行相干消色散脉冲星数据处理时, 该计算相较于 CPU 处理速度提升了 2 倍。

先讨论有关 GPU 并行算法的设计和应用, 图 2 展示了 GPU 内核运行方式。在 CPU 端启动核函数并根据问题动态分配资源后, GPU 会根

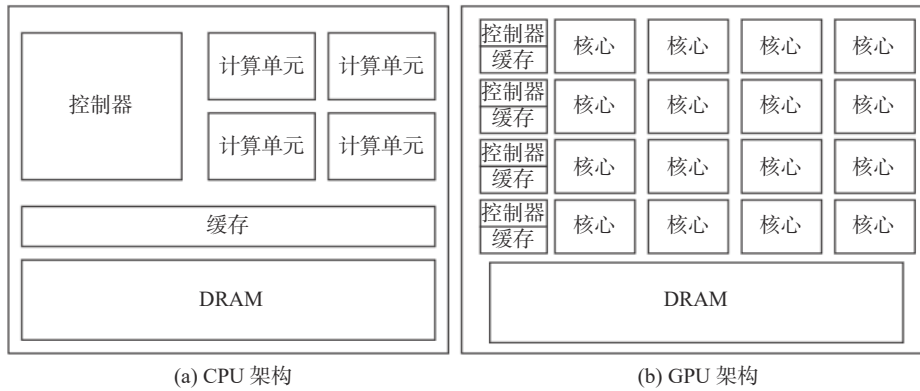


图 1 相同硅面积的 CPU 与 GPU 架构比较

Fig. 1 Comparison of CPU and GPU architectures with the same silicon area

据核函数中的代码进行运行。在 GPU 端运行核函数时，每个核函数会启动线程网格，每个线程网格由若干线程块组成，而线程块由多个线程构

成，这些线程负责处理数据。同一个线程块中若干个线程组成一个线程束，线程束中的线程以不同数据资源执行相同的指令。

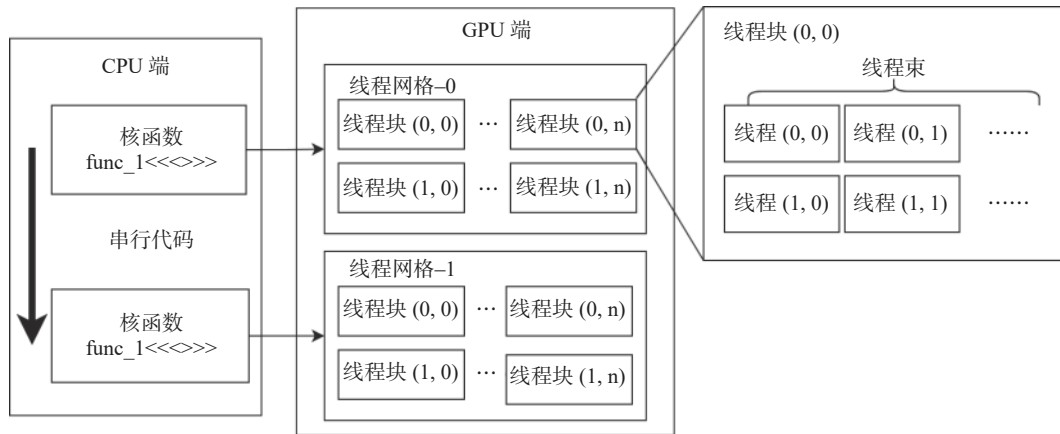


图 2 GPU 内核运行方式

Fig. 2 GPU kernel runtime method

利用 CPU/GPU 异构技术来实现基于 Zernike 多项式绝对平面测量算法。在实际异构加速计算过程中，GPU 部分主要负责对数据进行运算，CPU 部分负责逻辑流程处理以及 CPU/GPU 之间的数据通信，整体的流程如图 3 所示。在本研究中，CPU 主要负责初始化需要拟合的 Zernike 项数，加载面型数据，判断当前项数是否满足测量需要，根据待测平面分辨率分配 GPU 资源（包括 GPU 内存和线程数量以及线程块数量），启动 GPU 计算核心以及从 GPU 获取计算结果。GPU 主要负责计算 Zernike 多项式在笛卡尔坐标系下的映射，拟合 Zernike 多项式参数，拟合理想面型以及计算 PV 值和 RMS 值。其中，整个计算最复杂的部分就是 GPU 算法的设计，接着

将对这一部分的实现方法进行介绍。

1.2 Zernike 多项式映射计算

Zernike 多项式是一组定义在单位圆域内的完全正交基^[15]。其数学描述包括 2 个主要部分

$$R_n^m(\rho) = \frac{\sum_{x=0}^{(n-|x|)/2} (-1)^s (n-s)!}{s! [0.5(n+|m|)-s]! [0.5(n-|m|)-s]!} \rho^{n-2s} \quad (1)$$

$$\Theta(m\theta) \begin{cases} \cos(m\theta), m \geq 0 \\ \sin(-m\theta), m < 0 \end{cases} \quad (2)$$

式(1)为独立的径向多项式 $R_n^m(\rho)$ ，式(2)为角多项式 $\Theta(m\theta)$ 。 $R_n^m(\rho)$ 的各项只与径向有关， n 表示它们的多项式阶数， s 表示多项式次数， ρ 表

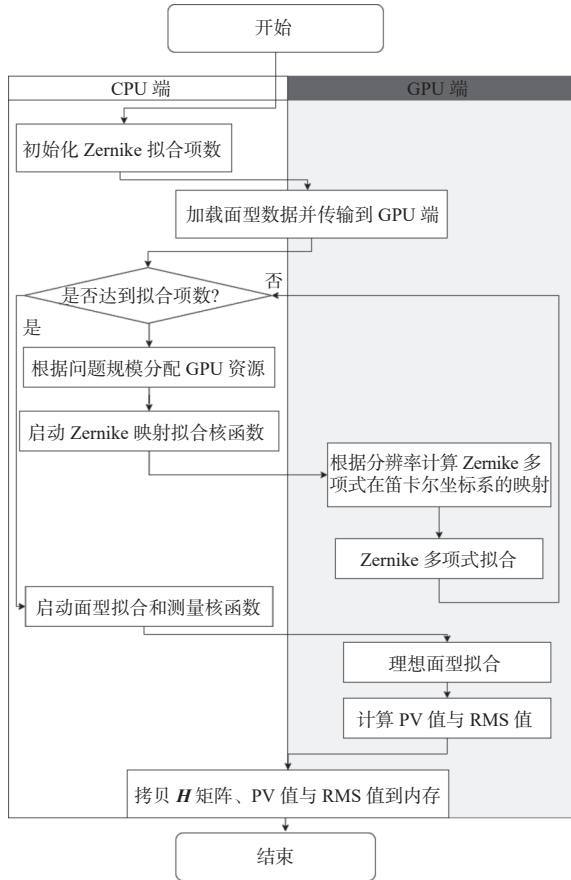


图 3 异构加速计算总流程

Fig. 3 Heterogeneous accelerated computing flow

示弧度, m 表示角频率。 $\Theta(m\theta)$ 的各项与振幅角度有关, θ 表示振幅角度。其极坐标表达式为

$$Z_n^m(\rho, \theta) = R_n^m(\rho) \cdot \Theta(m\theta) \quad (3)$$

Zernike 多项式与光学设计中的各种经典像差相对应, 常用于描述光学系统的波前像差。

图 4 为 Zernike 多项式映射的异构并行的计算流程在 5×5 分辨率下的示例图。在干涉测量中, 通常是对解相位包裹后的图像进行测量, 解相位包裹后的数据使用直角坐标表示。因此, 需要建立径向多项式 $R_n^m(\rho)$ 和角多项式 $\Theta(m\theta)$ 单位极坐标与直角坐标的映射关系, 即图 4 中 cart2pol 计算部分。这种关系的转换可依据待测面型的宽度和高度信息生成。假设待测拟合波面 $W(\rho, \theta)$ 的像素尺寸为 ROWs×COLs, 在区间 $[-1, 1]$ 内生成网格采样点, 然后根据式(4)和式(5)的转换, 便可得出这组映射关系。

$$Radial_map(x, y) = \sqrt{X(x, y)^2 + Y(x, y)^2} \quad (4)$$

$$Angular_map(x, y) = \arctan \left[\frac{Y(x, y)}{X(x, y)} \right] \quad (5)$$

可以发现, 该计算中像素之间数据不存在依赖关系。因此, 可以借助 GPU 多核特性来分配像素尺寸的线程数量并进行计算, 快速得到该分辨率直角坐标系下的极坐标信息, 然后按照式(1)和式(2)便可分别计算 $R_n^m(\rho)$ 和 $\Theta(m\theta)$ 。最后, 通过对应相乘(图 4 中 mult 计算部分)可得

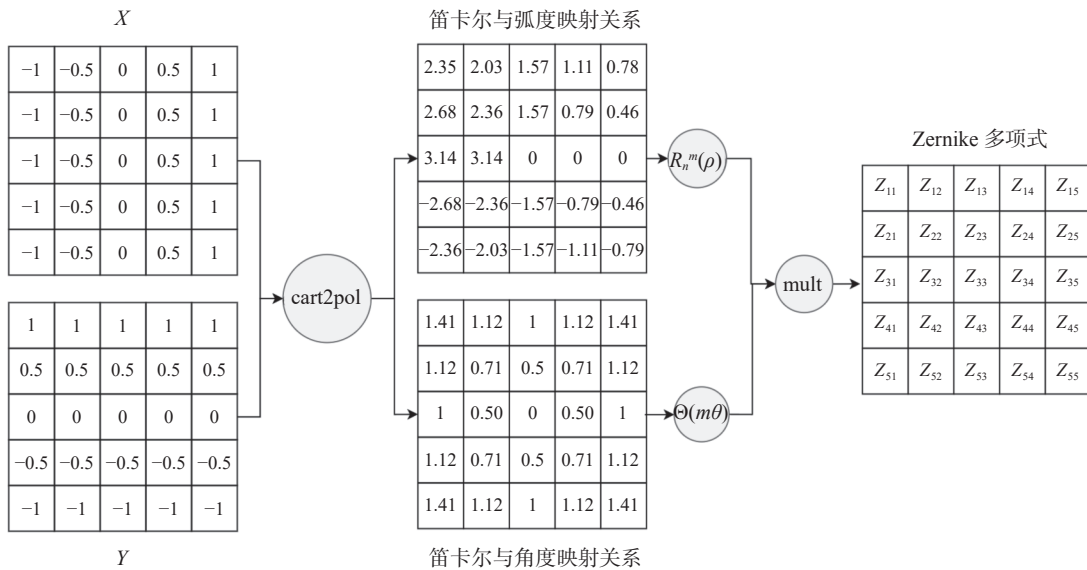


图 4 计算 5×5 分辨率的 Zernike 多项式示例

Fig. 4 Calculating 5×5 Zernike polynomials

到当前分辨率下笛卡尔坐标系的 Zernike 多项式。

1.3 Zernike 多项式系数求解

由于 Zernike 多项式与光学像差相对应，因此任何波前像差都可以用 Zernike 多项式作为基函数表示为

$$W(x,y) = \sum_{i=1}^n a_i Z_i(x,y) \quad (6)$$

式中： $Z_i(x,y)$ 是第 i 项 Zernike 多项式； a_i 是该多项式对应的 Zernike 系数； $W(x,y)$ 是真实离散面型数据。为了方便计算机运算，将式(6)转为矩阵表示形式

$$W = ZA \quad (7)$$

式中： $A = (a_1, a_2, \dots, a_n)^T$ ； $Z = (Z_i)_{R \times C}$ ， $R \times C$ 为分辨率。多项式拟合是为了求解其系数，即公式中的系数向量 A 。如式(7)所示，需要根据已知真实波面数据 $W(x,y)$ 计算出相应的系数 A 。借助 GPU 可实现 LU 分解算法，通过三角求解器可求解该问题。

如表 1 所示，当前异构传统直接算法性能相较于单纯的 CPU 计算性能提升了 4~8 倍。考虑到数值算法的发展以及 GPU 低精度矩阵乘硬件的性能优势，本文将使用混合精度迭代法作为求解器，进一步提升线性求解器性能^[16]。

表 1 不同线性求解器求解耗时
Tab. 1 Time used for different linear solvers

方法	耗时/ms			
	512×512	1 024×1 024	2 048×2 048	4 096×4 096
常规方法	12.4	48.5	367.3	1 203.2
异构传统直接法	2.6	8.9	41.1	221.7

如图 5 所示，混合精度迭代法通过混合计算半精度(FP16)和单浮点精度(FP32)在传统的直接异构并行方法基础上将性能提升了 1~5 倍。同时可以发现，随着待测面型的分辨率增大，线性求解器获得的性能加速比就越大，在 4 096 × 4 096 分辨率条件下其性能提升了约 5 倍。

计算出所有系数后，根据需要用 Zernike 多项式与系数计算拟合波面。在实际计算过程中，借助 GPU 核心数量多，理论吞吐量大的优势为

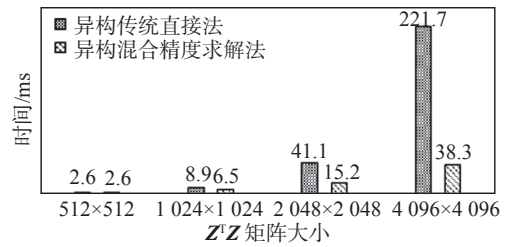


图 5 异构线性求解器性能比较

Fig. 5 Performance comparison of heterogeneous linear solvers

每个独立像素分配线程来完成计算。在假设求解系数为 36 项的基础上测得的性能数据如图 6 所示，可得该方法在超高分辨率(4 096 × 4 096)情况下相较于传统方法的性能提升了 71.8 倍。

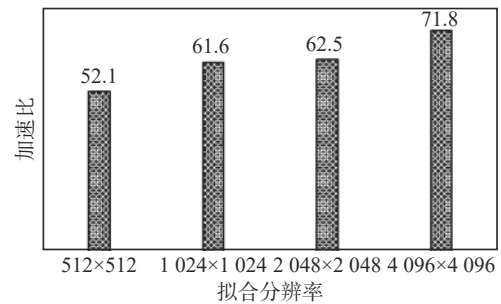


图 6 异构计算方式拟合理想面型加速比

Fig. 6 Heterogeneous computational approach for fitting face acceleration ratios

1.4 PV 值与 RMS 值异构并行计算

在绝对平面测量中，通常使用峰谷值(V_{pv})和均方根值(V_{rms})作为评估指标

$$V_{pv} = \max[H(x,y)] - \min[H(x,y)] \quad (8)$$

$$V_{rms} = \sqrt{\frac{\sum_{x,y} H(x,y)^2}{RC}} \quad (9)$$

式中： $H(x,y)$ 为拟合波面与真实波面 $W(x,y)$ 的差； R 为分辨率宽度； C 为分辨率高度。

根据式(8)可知， V_{pv} 是通过矩阵中的最小和最大值做差求得。该计算数据之间的依赖性较强，无法使用 Zernike 多项式映射的方式加速，可通过组合规约并行模型来进一步提高求解速度。在 Chen 等^[8] 和 Jradi 等^[17] 提出的 CPU/GPU 异构规约模型中，存在较多分支语句，GPU 逻辑处理分支的效率较差，复杂度较高。其具体解决方案的步骤如图 7(a) 所示，计算步骤需要

$N/2$ 步, 其中 N 为线程块中线程数量。当前线程块线程束(线程束由多个线程组成)的数量为 2。为进一步提升效率, 对其进行了性能优化, 主要借助协作组线程束规约原语指令, 该指令通常用于 GPU 程序性能优化, 能够将每个线程束中的数据进行单指令规约^[18]。图 7(b)所示为使用指令优

化后的峰/谷值求解算法, 优化前需要 N 个线程迭代 $N/2$ 次, 每次对问题规模进行折半计算, 直到求解出最后的元素。优化后, 只需要执行 2 步即可完成计算: 首先, 通过规约原语指令将每个线程束的最小/最大值筛出; 随后, 只需比较线程束中筛出元素即可。算法的复杂度和耗时均降低。

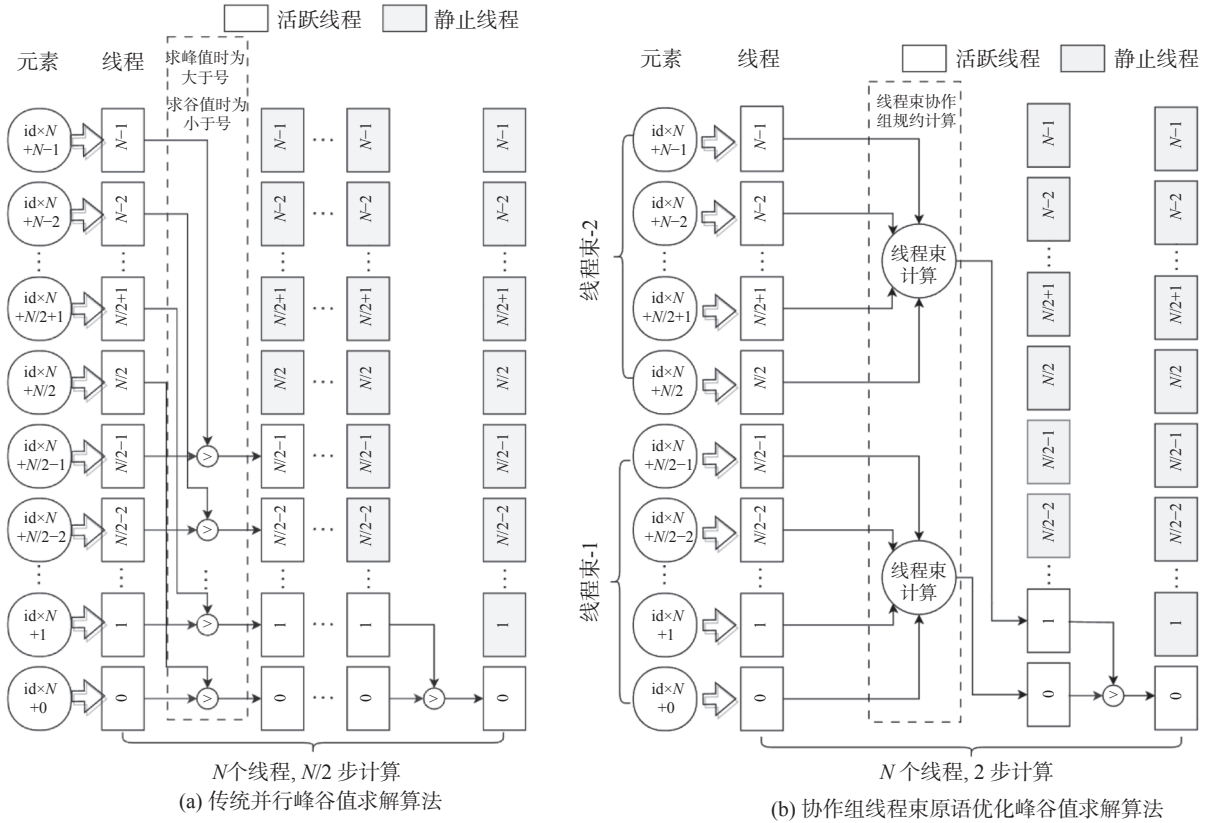


图 7 并行峰/谷值求解算法

Fig. 7 Parallel peak/valley solving algorithm

图 8 所示为当前实验环境下求解峰值或者谷值的性能对比图。本文算法相较于未优化的传统异构并行计算峰/谷值的算法快 8~20 倍, 并且更适合处理大分辨率的情况。求峰谷值需要计算峰值和谷值, 并对它们进行做差处理, 因此需要

在上述算法进行 2 次的基础上增加做差核函数。

根据式(9)可知, 求解 RMS 值的计算中, 需要计算波前拟合误差矩阵所有元素和的平方, 该算法与峰谷值求解逻辑相似。此外, 需要对求得和进行代数计算, 可利用额外的核函数来进行处理。

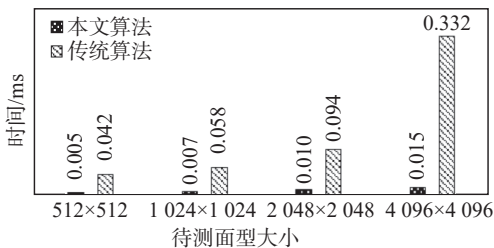


图 8 查找峰/谷值的不同算法性能

Fig. 8 Performance of different algorithms for finding peak/valley value

2 实验与分析

2.1 实验平台与仪器

基于上述并行异构算法搭建实验平台, CPU 为 AMD Ryzen 5900H, 加速 GPU 为 RTX3070-Laptop, 两者的具体参数见表 2, 整机的本地内

存为 16 GB。GPU 部分的代码根据 NVCC-11.6 测试环境操作系统内核为 Linux-5.17。来编译，CPU 部分代码用 GCC-7.3.0 来编译，

表 2 CPU 和 GPU 的参数
Tab. 2 CPU and GPU parameters

硬件	频率/GHz	最大频率/GHz	1级缓存/MB	2级缓存/MB	核心数
AMD Ryzen 5900H	3.3	4.3	4	16	8
RTX3070-Laptop	1.2	1.6	0.125	16	5 120

所使用的数据为图 9(a)所示的平面面型数据，该数据由干涉条纹解相位包裹获得。干涉条纹数据通常由图 9(b)所示干涉仪捕获。本文测试数据由苏州慧利仪器有限责任公司提供，共 4 组数据，每组数据具有不同的分辨率。

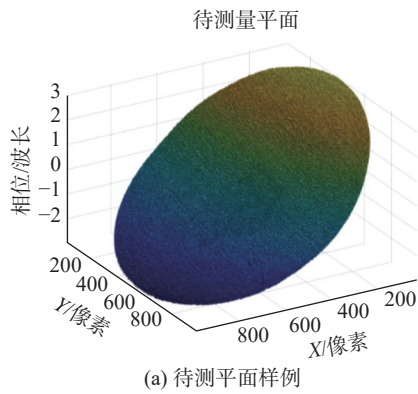


图 9 待测平面样例与干涉仪
Fig. 9 Example of the surface to be measured and the interferometer

2.2 功能验证

在性能验证之前，先对算法进行功能验证。将图 9(a)所示样例作为算法的输入，可以得到如图 10 所示的面型。该面型与常规计算方法算得的面型相同。

2.3 性能验证

在当前实验环境下，分别对 Zernike 多项式

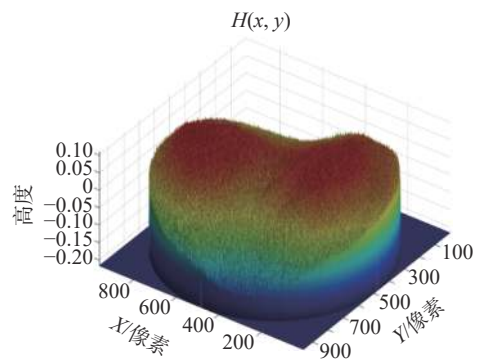


图 10 1 024 × 1 024 分辨率面型图
Fig. 10 Picture of 1 024 × 1 024 resolution flatness

计算、Zernike 系数求解以及 PV 值和 RMS 值求解等步骤进行性能测试。

图 11 所示为不同步骤在 4 组不同分辨率待测面型情况下异构计算加速比。其中图 11(a)为笛卡尔坐标系下的多项式计算，加速比为 50~70 倍；图 11(b)为多项式拟合，加速比为 6~25 倍；图 11(c)为测量面型、PV 值和 RMS 值计算部分，加速比为 50~70 倍。与常规计算相比，异构计算所获得的性能增益随着像素的增加而增加。

图 12 所示为在 4 组不同分辨率待测面型情况下不同步骤的异构计算标准差。其中图 12(a)为笛卡尔坐标系下的多项式计算，标准差为 0.83~5.28 ms；图 12(b)为多项式拟合，标准差为 0.90~1.29 ms；图 12(c)为测量面型、PV 值和 RMS 值计算部分，标准差为 0.08~0.16 ms。3 个步骤的标准差均随着分辨率增大而减小，原因在于随着数据量的增加，GPU 的调度器可以更有效地分配任务，使每个处理单元的工作负载更加均衡，内存带宽的利用也更充分，从而有助于减小完成上述步骤的耗时差异，降低标准差。

表 3 所示为不同分辨率下采用常规计算方法和异构计算方法的平均耗时情况。通过分析可

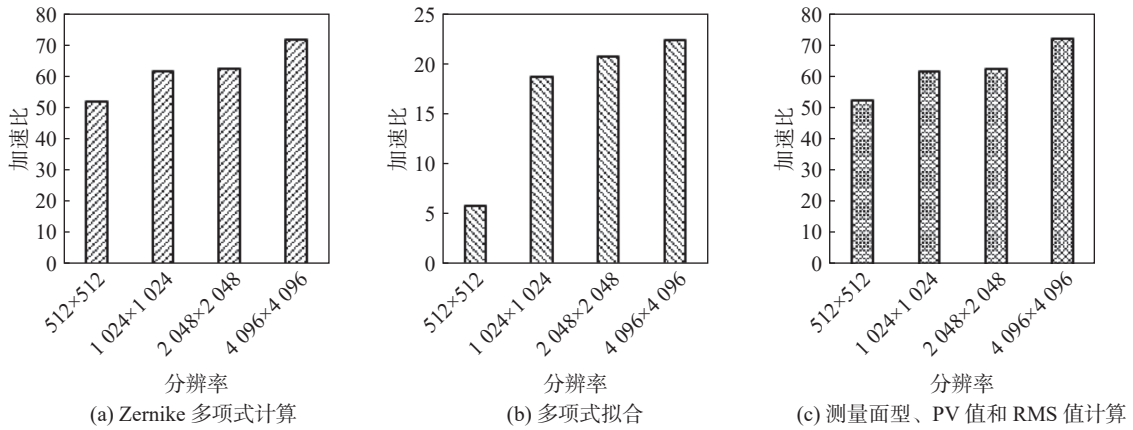


图 11 不同步骤在不同分辨率面型情况下的加速比

Fig. 11 Acceleration ratios for different steps with different resolution plane to be measured

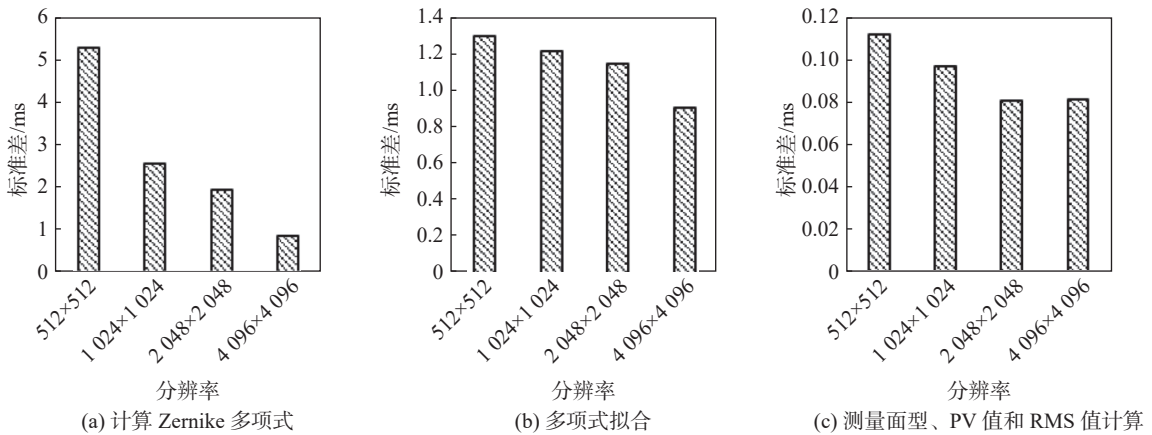


图 12 不同步骤在不同分辨率面型情况下的标准差

Fig. 12 Standard deviation for different steps with different resolution plane to be measured

知, 如果采用传统计算方法, 在针对 $4\,096 \times 4\,096$ 高分辨率测量时的平均耗时约 42 s, 而采用异构方法平均仅需要 602 ms; 在针对 $1\,024 \times 1\,024$ 分辨率时, 异构算法的测量速度能够达 20 次/s 左右, 相较于常规算法其整体加速了 40~70 倍。在实际的干涉测量研究和应用中, 需要对面型进行重复测量, 使用异构算法将极大提高测试效率。

表 3 Zernike 绝对平面测量平均耗时
Tab. 3 Average time used for Zernike's absolute flatness testing

方法	测量平均耗时/ms			
	512×512	1 024×1 024	2 048×2 048	4 096×4 096
常规计算	710	2 752	10 412	42 567
异构计算	15	49	178	602

3 结 论

本文实现了 CPU/GPU 异构并行加速绝对平面检测, 包括 Zernike 多项式映射计算求解、Zernike 多项式拟合系数求解、绝对平面测量参数并行化算法设计与实现以及 CPU 流程控制部分。该方法可以提高高分辨率绝对平面测量的效率。测试了多组不同分辨率的光学平面, 随着分辨率的提高, GPU 并行加速带来的性能提升也越高。在当前实验环境中, 针对不同分辨率, 其性能提升了 40~70 倍, 且分辨率越大性能提升越明显, 其中在处理 $4\,096 \times 4\,096$ 分辨率的光学平面时, 平均只需要 0.6 s 左右, 比常规 CPU 计算性能提高了 70 倍左右, 极大提升了干涉测量效率。未来将进一步探索异构并行在其他光学面型测量领域的应用。

参考文献:

- [1] FRITZ B S. Absolute calibration of an optical flat[J]. *Optical Engineering*, 1984, 23(4): 234379.
- [2] VANNONI M, MOLESINI G. Absolute planarity with three-flat test: an iterative approach with Zernike polynomials[J]. *Optics Express*, 2008, 16(1): 340 – 354.
- [3] FREISCHLAD K R. Absolute interferometric testing based on reconstruction of rotational shear[J]. *Applied Optics*, 2001, 40(10): 1637 – 1648.
- [4] KÜCHEL M F. A new approach to solve the three flat problem[J]. *Journal for Light and Electronoptic*, 2001, 112(9): 387 – 391.
- [5] GRIESMANN U. Three-flat test solutions based on simple mirror symmetry[J]. *Applied Optics*, 2006, 45(23): 5856 – 5865.
- [6] 黄元申, 吕昊宇, 曾媛, 等. 绝对平面检测方法的研究进展 [J]. *光学仪器*, 2018, 40(1): 72 – 77.
- [7] AIKENS D M, RICH L, BAJUK D, et al. Developing enabling optics finishing technologies for the National Ignition Facility[C]//Proceedings of the Optical Fabrication and Testing. Kailua-Kona: OFT, 1998.
- [8] CHEN Y C, WANG T Y, KEMAO Q. Parallel advanced iterative algorithm for phase extraction with unknown phase-shifts[J]. *Optics and Lasers in Engineering*, 2021, 138: 106408.
- [9] HERNANDEZ-LOPEZ F J, RIVERA M, SALAZAR-GARIBAY A, et al. Comparison of multihardware parallel implementations for a phase unwrapping algorithm[J]. *Optical Engineering*, 2018, 57(4): 043113.
- [10] UJALDON M. GPU acceleration of Zernike moments for large-scale images[C]//Proceedings of 2009 IEEE International Symposium on Parallel & Distributed Processing. Rome: IEEE, 2009: 23 – 29.
- [11] TOHARIA P, ROBLES O D, SUÁREZ R, et al. Shot boundary detection using Zernike moments in multi-GPU multi-CPU architectures[J]. *Journal of Parallel and Distributed Computing*, 2012, 72(9): 1127 – 1133.
- [12] KHAN F H, PASHA M A, MASUD S. Advancements in microprocessor architecture for ubiquitous AI—an overview on history, evolution, and upcoming challenges in AI implementation[J]. *Micromachines*, 2021, 12(6): 665.
- [13] 赵锴坤, 朱炬波, 谷德峰, 等. MPI+CUDA 联合加速重力场反演的并行算法 [J]. *大地测量与地球动力学*, 2024, 44(4): 423 – 428.
- [14] 李浩. 基于 GPU 的相干消色散脉冲星数据处理技术研究 [D]. 北京: 中国科学院大学, 2021.
- [15] SHANNON R R, WYANT J C. *Applied optics and optical engineering*[M]. San Diego, California: Academic Press, 1992: 29 – 39.
- [16] ABDELFAH A, ANZT H, BOMAN E G, et al. A survey of numerical linear algebra methods utilizing mixed-precision arithmetic[J]. *The International Journal of High Performance Computing Applications*, 2021, 35(4): 344 – 369.
- [17] JRADI W A R, DO NASCIMENTO H A D, MARTINS W S. A fast and generic GPU-based parallel reduction implementation[C]//Proceedings of 2018 Symposium on High Performance Computing Systems. Sao Paulo: IEEE, 2018: 16–22.
- [18] HIJMA P, HELDENS S, SCLOCCO A, et al. Optimization techniques for GPU programming[J]. *ACM Computing Surveys*, 2023, 55(11): 239.

(编辑: 李晓莉)