

doi: 10.3788/gzxb20164503.0310001

采用图形处理器加速的部分相干光实时生成方法

倪小龙, 刘智, 姜会林, 陈纯毅, 刘艺, 齐冀, 宋卢军

(长春理工大学 空地激光通信技术国防重点学科实验室, 长春 130022)

摘要:介绍了采用液晶空间光调制器产生部分相干光的基本理论和方法;用分离变量的方式对算法进行了并行优化,对部分相干光生成线程进行了并行序列优化并测试了生成速度;设计了双孔干涉实验检测本方法所产生的部分相干光相干长度的准确性.实验结果表明:针对 256×256 像素的液晶空间光调制器,生成相干长度为 0.15 mm、0.9 mm 以及 1.5 mm 的相位屏,图形处理器所用时间分别为 0.9 ms、1.75 ms 以及 2.4 ms,相比于采用中央处理器所用的计算时间 16.5 ms、37.8 ms 以及 52.4 ms,效率提升明显;考虑中央处理器读取、发送数据及液晶响应的的时间,采用图形处理器并行加速的部分相干光实时生成方法的频率可达 312 Hz.实验生成相干长度为 0.15 mm 和 1.5 mm 的部分相干光束,光束相干度均方根误差分别为 0.022 011 和 0.020 883,峰谷值分别为 0.0743 25 和 0.072 998.

关键词:相干光学;实时部分相干光生成;并行加速;光场调制;相干检测

中图分类号:O436; O439

文献标识码:A

文章编号:1004-4213(2016)03-0310001-6

Partially Coherent Beam Real-Time Generation Method Accelerated by Graphic Processing Unit

NI Xiao-long, LIU Zhi, JIANG Hui-lin, CHEN Chun-yi, LIU Yi, QI Ji, SONG Lu-jun

(Fundamental Science on Space-Ground Laser Communication Technology Laboratory,
Changchun University of Science and Technology, Changchun 130022, China)

Abstract: The theory and method for generating partially coherent beam based on liquid-crystal spatial light modulator were given. The method was optimized by using partition convolution, the generation of partially coherent beam was parallel sequential optimized and the generation speed was tested. The coherence width accuracy of partially coherent beam based on double hole interference was detected. Experimental results show that, to a spatial light modulator with a resolution of 256×256 , the total time of generating the random phase maps with coherence widths of 0.15mm, 0.9mm and 1.5mm by graphic processing unit are 0.9 ms, 1.75 ms and 2.4 ms, respectively, which are 16.5 ms, 37.8 ms and 52.4 ms by using central processing unit. Considering the reading, sending and response time of the liquid-crystal spatial light modulator by central processing unit, the real-time partially coherent beam generation frequency is up to 312 Hz, which is accelerated by graphic processing unit. For generating a partially coherent laser beam with coherence widths are 0.15 mm and 1.5 mm respectively, the the root-mean-square error are 0.021 306 and 0.020 883, the peak-to-valley are 0.073 576 and 0.072 998.

Key words: Coherent optics; Real-time partially coherent beam generate; Parallel computing; Light field modulation; Coherent measurement

OCIS Codes: 100.1390; 030.1640; 230.6120; 160.3710

基金项目:国家自然科学基金(Nos. 61007046, 61275080)资助

第一作者:倪小龙(1988-),男,博士研究生,主要研究方向为大气光学、液晶光调制. Email: nixiaolong@gmail.com

导师(通讯作者):姜会林(1945-),男,教授,博士,主要研究方向为光学系统设计、光电检测技术和激光通信技术等. Email: HLJiang@cust.edu.cn

收稿日期:2015-07-01;录用日期:2015-10-23

<http://www.photon.ac.cn>

0 引言

1970年以来,部分相干光在自由空间激光通信^[1-3]、激光材料热处理、激光扫描、激光可控核聚变、成像^[4-5]光学以及非线性光学等领域有广泛的应用^[6-7]。目前,产生部分相干光的方法主要有两种,其中最为常见的为Ari T. Friberg提出的采用旋转毛玻璃进行部分相干光生成的方法,通过将完全相干的激光光束照射到匀速旋转的毛玻璃上生成了部分相干光^[8],Dario Ambrosini等随后对这种方法所生成部分相干光的准确度进行了实验验证^[9]。采用旋转毛玻璃的方法所生成的部分相干光光束均匀性较好,但其生成光束的相干长度由毛玻璃表面微结构的特征尺寸唯一决定,若需要对光束的相干长度进行更改,需重新制作毛玻璃,相干长度的更改非常不便,且成本较高,生成光束相干长度的准确性较低。液晶空间光调制器具有高分辨率、线性可编程、低成本以及低功耗等优点,非常适合对光束的相位进行调控。Emil Wolf等提出了采用液晶空间光调制器生成部分相干光的方法^[10-11],可以提高生成部分相干光相干长度的准确度,但若需改变光束的相干长度,仍需要重新生成相位屏^[12-13]。显然这种方法仍存在着一定的局限性,无法快速便捷地对光束相干长度进行高准确度实时调控。

本文提出了一种基于图像处理(Graphic Processing Unit, GPU)的部分相干光实时生成算法,并在整个生成过程中对GPU进行并行优化,优化了线程运行机制,引入了并行序列优化。与传统方法相比,可以实时根据需求更改所生成光束的相干长度,提高相干长度更改的速度及便捷性,且相干长度的调整可以在一定范围内实现高准确度的连续可调。

1 部分相干光生成算法

大多数的液晶空间光调制器都是通过电信号进行控制的,对每个像素加载不同强度的控制信号,便可以实现对入射光束相位的调制。而电压大小通常是由加载到液晶上的灰度图控制的。所以,利用液晶空间光调制器生成部分相干光的主要工作是计算相应的随机灰度图。首先,假设随机相位 $\phi(\rho, \omega)$ 服从零均值高斯分布,其二阶相关性可表示为^[14]

$$\langle \phi(\rho_1, \omega) \phi(\rho_2, \omega) \rangle = \phi_0^2 \exp \left[-\frac{|\rho_1 - \rho_2|^2}{2\delta_\phi^2} \right] \quad (1)$$

式中

$$\phi_0 = \sqrt{\langle |\phi(\rho, \omega)|^2 \rangle} \quad (2)$$

δ_ϕ 为相位相干长度。液晶空间光调制器出射平面 $z = z_0$ 的功率谱密度函数可以表示为

$$W(\rho_1, \rho_2, z_0) = S_0 \mu(\rho_1, z_0) \mu^*(\rho_2, z_0) \times \langle \exp i[\phi(\rho_2, \omega) - \phi(\rho_1, \omega)] \rangle \approx S_0 \mu(\rho_1, z_0) \mu^*(\rho_2, z_0) \times \exp \left[-\frac{|\rho_1 - \rho_2|^2}{2\delta_\phi^2} \right] \quad (3)$$

式中, $S_0 = S_0(\omega)$ 是初始波面的频率密度 $\mu(\rho, z_0)$,表示频率为 ω 时的部分相干电磁场, ρ_1 和 ρ_2 为在出射平面 $z = z_0$ 处的二维位置向量。此时,出射光束为部分相干光,并且,相干长度由随机相位函数的相位相干长度 δ_ϕ 唯一决定。因此,只需控制液晶空间光调制器(Liquid Crystal Spatial Light Modulator, LC SLM)所加载随机相位的相位相干长度,便可以在液晶空间光调制器出射平面上生成定相干长度的部分相干光束。可见采用液晶空间光调制对激光光束进行相干度控制的前提是生成不同相位相干长度的随机相位屏,通过控制相位相干长度的大小,来控制输出光束相干长度的值。

随机相位屏的生成过程中,首先,生成一个二维服从零均值高斯分布的实值函数 $R_\phi(\rho)$ 。由于阵列单元独立服从高斯分布,所以互不相关的,即

$$R_\phi(\rho_1) R_\phi(\rho_2) = \delta^{(2)}(\rho_1 - \rho_2) \quad (4)$$

式中, $\delta^{(2)}(\rho)$ 为二维狄拉克函数。然后,通过随机函数 $R_\phi(\rho)$ 的卷积产生高斯相关随机函数 $g_\phi(\rho)$

$$g_\phi(\rho) = f_\phi(\rho) * R_\phi(\rho) \quad (5)$$

其窗口函数 $f_\phi(\rho)$ 表示为

$$f_\phi(\rho) = \exp \left(-\frac{\rho^2}{\gamma_\phi^2} \right) \quad (6)$$

函数 $g_\phi(\rho)$ 的二阶相关性表示为

$$[g_\phi(\rho_1) g_\phi(\rho_2)] = \frac{\pi \gamma_\phi^2}{2} \exp \left[-\frac{(\rho_1 - \rho_2)^2}{2\gamma_\phi^2} \right] \quad (7)$$

可见此时相位函数 $g_\phi(\rho)$ 的相位相干长度为 δ_ϕ ,通过控制函数 $g_\phi(\rho)$ 的相位相干长度 δ_ϕ ,并将其加载到液晶空间光调制器上,便可以实现对光束相干度的控制。需要说明的是,这里提到的相干长度 δ_ϕ 为相对量,为无量纲量,其与绝对的相干长度 l_c 的关系为 $l_c = m\delta_\phi$, m 为相干长度转换系数,由液晶的像元尺寸和光束所经过光学系统的缩放倍率共同决定。但是,这个计算的过程耗时较长。为了实时地计算相位屏,需要使用GPU进行加速及并行优化。

2 并行优化

2.1 算法并行优化

为了便于操作,将随机相位函数 $g_\phi(\rho)$ 转换成直角坐标的形式,式(5)就变成了两个二维矩阵卷积,即

$$g_\phi(x, y) = (f_\phi * R_\phi)(x, y) = \sum_n \sum_m f_\phi(x - n, y - m) R_\phi(n, m) \quad (8)$$

同一维卷积一样,二维矩阵卷积的实质是将卷积核矩阵翻转(旋转 180°),等同于一维信号的翻转,然后将卷积核矩阵依次从上到下、从左到右滑动,计算在模板与原始图像交集的元素的乘积和,作为卷积以后的数值。由于窗口函数的尺寸问题,需要对随机函数用零进行补全。图1为生成部分相干光矩阵卷积示意图,图

中最外一圈区域便为补全区域. 为了更好展示这个计算过程, 图中采用与函数相对应的灰度图来表示各个相应矩阵.

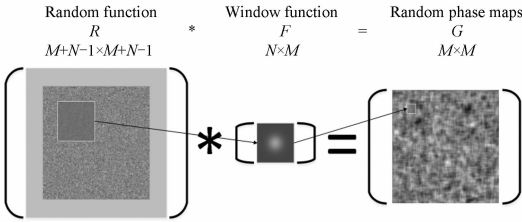


图 1 部分相干光生成算法示意图

Fig. 1 Diagram of partially coherent beam generating algorithm

为了提高 GPU 的计算速度, 需要做进一步的并行优化. 将窗口函数分解为一个行矩阵和一个列矩阵, 使得整体的计算量由 $n \times n$ 减小到 $n + n$, 一次卷积运算被分解为两次独立的卷积运算. 首先, 进行行卷积运算; 然后利用行卷积所得计算数据进行列卷积计算. 这样也同时减少了很多条件的限制. 并且, 在进行行卷积的时候, 不需要考虑垂直方向数据补全; 同样地, 在进行列运算的时候, 不需要考虑水平方向的数据补全. 这在很大程度上减小了数据补全的数量, 进一步降低了计算量.

在对算法进行并行优化后, 就可以在通用计算架构 (Compute Unified Device Architecture, CUDA) 环境下采用 GPU 进行编程计算. GPU 采用的是英伟达公司的 GTX680 型 GPU, 具有 1536 个流处理器, 频率为 1 008 Hz, 具有 2 048 MB 存储器, 内存带宽为 192 GB/s, 每秒浮点计算能力为 3.09TFLOPs. 矩阵卷积和部分相干光生成的程序利用英伟达公司提供的 CUDA 库采用 C++ 进行编写. 为了进一步提升 GPU 计算的速度, 还需要针对 GPU 硬件本身进行优化. 在 CUDA 架构下, 每一个 Block 对应一个物理处理器, 最大限度地利用 Block 资源是并行优化的核心技术. 由于本文采用的液晶分辨率为 256×256 , 所以一共需要计算 65536 个值. 在计算的过程中, 每一个线程用来计算一个像素的值, 这样需要 65536 个线程. 每个 Block 的 Thread 数应为 Warp 块大小的倍数^[15]. Warp 是一个最小单指令多线程 (Single Instruction Multiple Thread, SIMT) 单元, 它以 32 个并行线程为一组来创建、管理、调度和执行线程. 由此设定 Thread 数为 32, 可以计算得到所需 Block 的数目为 2 048.

2.2 并行序列优化

部分相干光生成单线程流程如图 2.

线程开始工作后, 首先由 CPU 生成高斯分布的零均值随机相位函数矩阵和窗口函数矩阵; 然后, 窗口函数和查阅表 (Look Up Table, LUT) 被送入高速的常量存储器中; 经过预处理的随机相位函数被从中央处

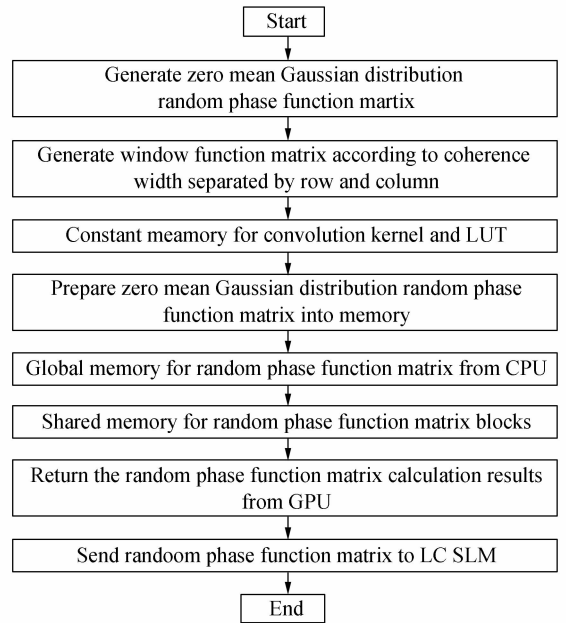


图 2 部分相干光生成计算单线程流程图

Fig. 2 Flowchart of single thread of partial coherent light generation calculation

理器 (Central Processing Unit, CPU) 送入到 GPU 的全局存储器中后被载入共用存储器中进行计算. 最后, 计算结果从 GPU 中返回到 CPU 中, 传送并被加载到液晶空间光调制器上. 可见每一个生成线程主要有 GPU 计算、CPU 读取以及液晶空间光调制器进行响应这三个过程. 若这三个过程采用串行的方式进行运行, 则每个线程的时间为三个过程的时间总和. 为了进一步提升部分相干光生成的速度, 需要对线程进行并行优化, 分别精确得到这三个过程所需要的时间.

分别设置了两组实验进行对比, 一组采用 GPU 进行计算并应用了并行优化, 另一组采用 CPU 进行计算. 计算两组生成 100 张灰度图所用的平均时间, CPU 读取以及液晶空间光调制器进行响应的平均时间.

实验所生成灰度图的对应相干长度分别为 0.15 mm、0.9 mm 以及 1.5 mm. 灰度图分辨率为 256×256 , 计算机 CPU 为 Intel 公司的 i7 3930 k (6 核心, 12 线程), 系统内存为 16 GB, GPU 为英伟达公司的 GTX680. 图 3(a)~(c) 分别为 GPU 计算所需时间、CPU 计算所需时间、CPU 读取时间及液晶空间光调制器响应时间. 可以看出, 采用 GPU 计算相干长度为 0.15 mm、0.9 mm 以及 1.5 mm 的灰度图所需时间分别为 0.9 ms、1.75 ms 以及 2.4 ms; 而采用 CPU 进行计算所需时间分别为 16.5 ms、37.8 ms 以及 52.4 ms, 远远大于采用 GPU 进行计算的时间. 若想提高 CPU 的计算速度, 通常需要提高 CPU 的频率. 但受限于半导体科技, 提高 CPU 的速率已经变得越来越难. 与之相比, 对运算进行并行优化后, 采用数以千计的流处理器进行运算能够有效提高计算速度. 从图中还可以看

到 CPU 读取数据并将数据传送到液晶所需的平均时间为 3.2 ms,液晶的平均响应时间为 2.5 ms.

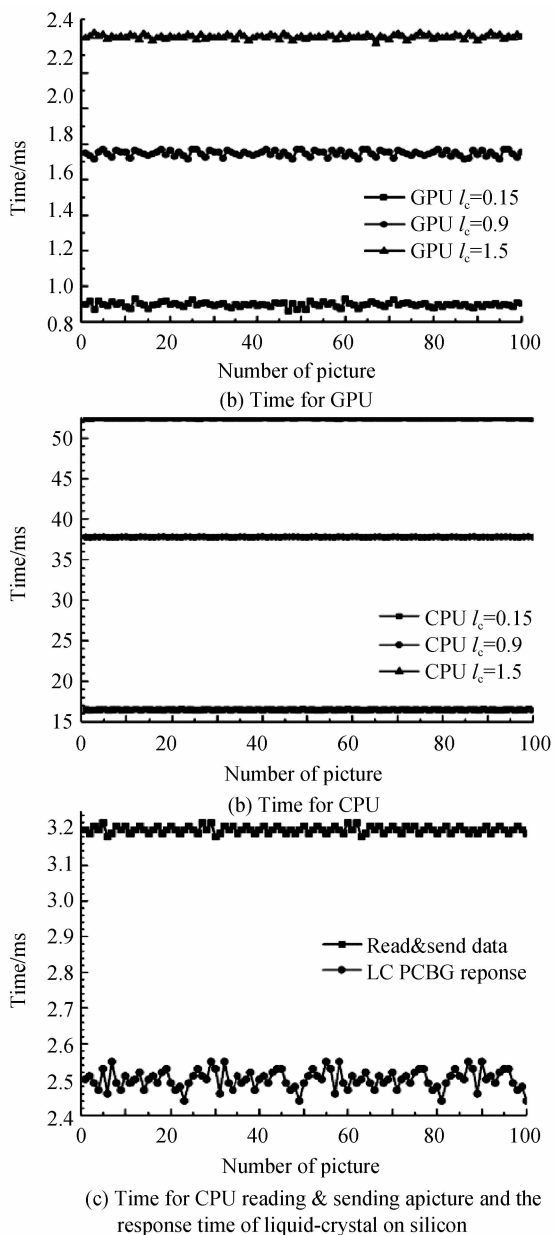


图3 计算时间对比

Fig. 3 Contrast of the calculation time

线程采用串行工作方式生成 0.15 mm 的相干长度所需时间为 7.5 ms,对应的频率为 133 Hz,还需要进一步进行并行序列优化.并行序列优化示意如图 4.在并行工作模式下,在计算完成前,下一个生成进程已经开始了.所以并行序列优化能很大程度上提高系统的生成速度.两个相邻进程的间隔由生成过程中耗时间最长的过程所决定.对于部分相干光的生成过程来说,最耗时间的过程为随机相位图的生成,即 GPU 计算的时间.并且,随机相位图的计算时间受所要生成相位图的相干长度影响很大,相干长度越长所需要的时间越长.光束的相干长度可以通过光学系统的缩放来进行调节,所以过长的相干长度实际上并没太大的作

用,只需要在一定的范围内可调即可.

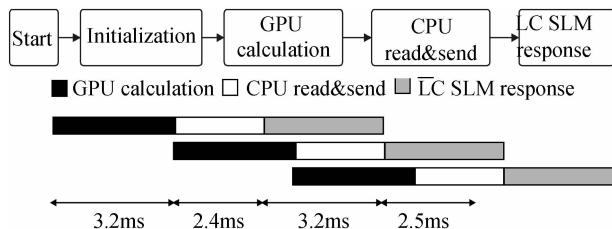


图4 液晶部分相干光生成装置并行序列

Fig. 4 Parallel sequential of the liquid-crystal partially coherent beam generator

如果光束的相干长度调整区间为 0 到 1.5 mm,则 GPU 计算每幅灰度图的时间小于 2.4 ms.此时,在部分相干光的生成进程中,CPU 读取数据并将数据传送到液晶的过程消耗时间最长,为 3.2 ms.故此时的延时时间应设为 3.2 ms.在这种情况下,当第一个进程还在读取数据并传递数据到液晶的时候,第二个进程已经开始计算随机相位图;当第三个进程开始计算随机相位图的时候,液晶正在响应第一个进程所加载的随机相位图,第二个进程正在读取数据并传递数据到液晶.按照这样的并行工作模式进行工作,整个系统生成部分相干光的频率可达 312 Hz,并且相邻两个进程的工作没有冲突,每个进程都有序地运行.在并行工作模式下,系统工作的速度受限于进程中最耗时间的过程,在次系统中为 CPU 读取数据并传递数据到液晶的过程.所以尽管采用进一步的优化(快速傅里叶变换等)可以减小 GPU 计算的时间,但对于整个系统的速度提升并没有效果,故没有采用.

3 实验与结果分析

为了测试采用 GPU 并行加速的实时部分相干光生成方法所生成部分相干光相干长度的准确性,进行了实验验证.实验装置及光路图如图 5.

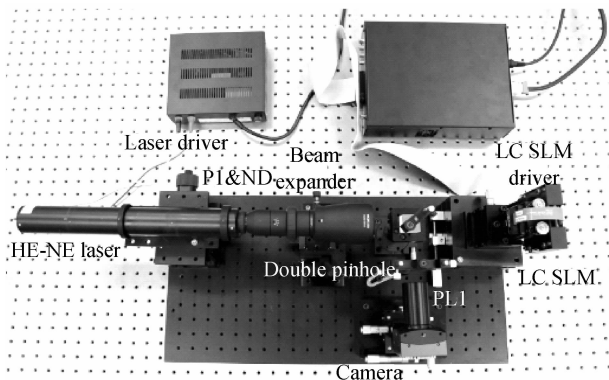


图5 相干长度检测实验示意图

Fig. 5 Experimental setup for testing the coherence width

实验中使用美国 BNS 公司生产的液晶空间光调制器 HSP256-0635,液晶分辨率为 256×256 .控制用计算机 CPU 为 Intel 公司的 i7 3930k(6 核心,12 线程),

系统内存为 16GB, GPU 为英伟达公司的 GTX680. 如图 5, 氩氦激光器产生标准高斯分布的激光光束, 经由伽利略式扩束装置(2~5 倍连续可调)进行扩束以调整光斑直径, 由一个衰减片(Neutral density, ND), 调节光束的强度使其在观测设备的动态范围内; 然后通过一个偏振滤光片 P1 对光束的偏振态进行调整, 使其满足液晶空间光调制器的需求. 液晶空间光调制器为相位调制器件, 入射光以 5° 的入射角入射到液晶上进行光束相干度的调整, 加载液晶上的随机相位屏相干长度分别为 0.9 mm 和 1.5 mm. 出射的部分相关光照射到双孔 P1 和 P2 上后发生干涉, 由透镜 PL1 会聚到互补金属氧化物半导体(Complementary Metal Oxide Semiconductors, CMOS)相机光敏面上进行观察. 这里所采用的双孔直径为 $150 \mu\text{m}$, 孔间距分别为 0.9 mm 和 1.5 mm. 透镜的口径为 25 mm, 焦距为 100 mm. 相机为德国 Mikrottron 公司的 CMOS 相机, 相机的分辨率为 1280×1024 , 像元尺寸为 $14 \mu\text{m}$, 满分辨率采样频率为 506 Hz, 在 633 nm 波段上相机的量子效率大于 40%, 信噪比优于 59 dB.

相干长度定义为两个相干度 $j_{12} = 1/e$ 的两点间距, 这里相干度 j_{12} 的定义为^[16]

$$j_{12} = \frac{I_{\max} - I_{\min}}{I_{\max} + I_{\min}} \quad (10)$$

式中, I_{\max} 和 I_{\min} 分别为相机观测到的干涉条纹中最亮点亮度值和最暗点亮度值, 在这里直接带入最亮点和最暗点的灰度值即可. 理论上, 经过液晶调制过的激光光束的相干长度分别为 0.15 mm 和 1.5 mm, 所以入

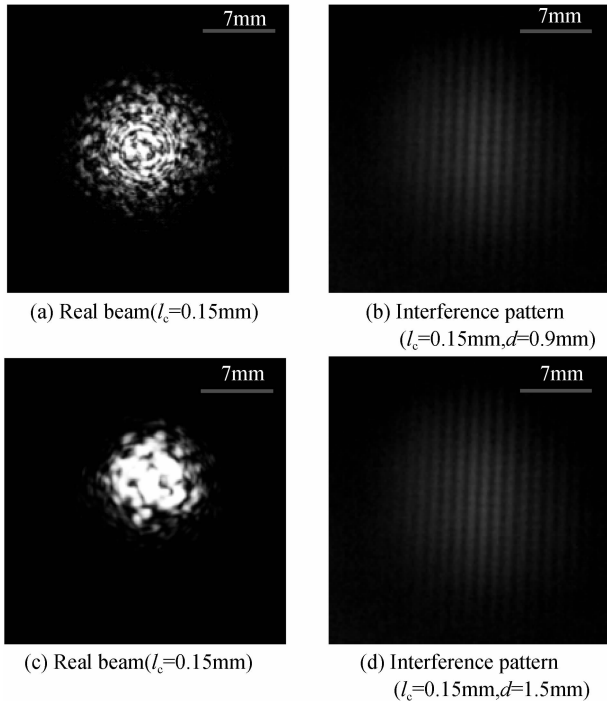


图 6 双孔干涉图

Fig. 6 The double pinhole interference pattern

射对应间距的双孔发生干涉时, 相机处测得的相干度为 $1/e$. 实验结果如图 6.

图 6(b) 为相干长度为 0.15 mm 的激光光束入射到间距为 0.15 mm 双孔后产生的干涉条纹, 实验测得该两点相干度 $j_{12} = 0.3618$; 图 6(d) 为相干长度为 1.5 mm 的激光光束入射到间距为 1.5 mm 双孔后产生的干涉条纹, 实验测得该两点相干度 $j_{12} = 0.3704$. 为了更好地分析并验证所生成部分相干光相干长度的准确性, 每个相干长度连续记录 500 帧图像后计算相干度并进行统计分析. 图 7 分别为相干长度为 0.15 mm 和 1.5 mm 两束激光入射对应双孔后通过所产生干涉条纹计算得到的相干度.

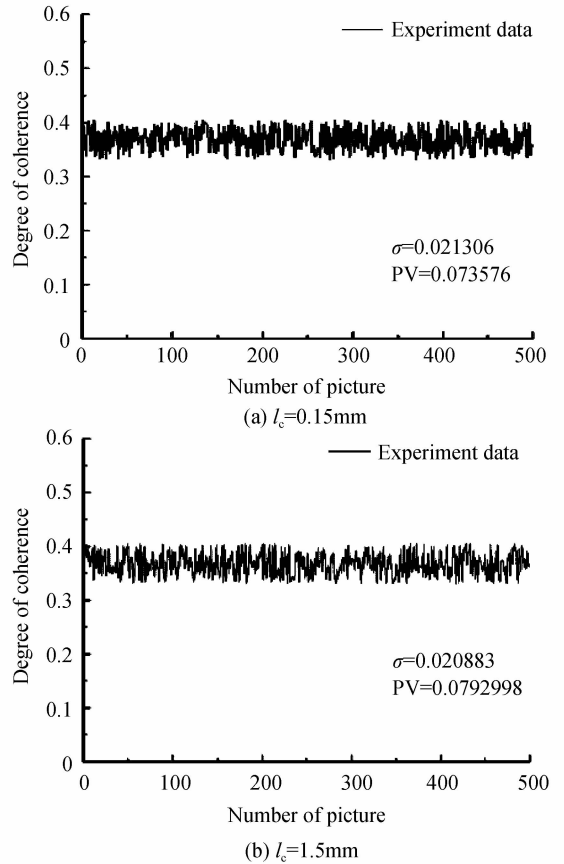


图 7 实验测得相干长度

Fig. 7 The coherence width of the experiment

对数据进行分析整理, 对于所生成的相干长度为 0.15 mm 的激光光束, 光束上间隔为 0.15 mm 两点的相干度均方根误差 $\sigma = 0.022011$, 峰谷值 $PV = 0.074325$; 对于所生成的相干长度为 1.5 mm 的激光光束, 光束上间隔为 1.5 mm 两点的相干度均方根误差 $\sigma = 0.020883$, 峰谷值 $PV = 0.072998$. 均与理论较相符, 可见基于 GPU 的并行加速实时部分相干光生成方法可以快速准确地生成一定相干长度的部分相干光束.

4 结论

本文提出了一种采用 GPU 并行加速的实时部分相干光生成方法,并针对 GPU 进行了算法的并行优化及并行序列优化. 经过优化,生成相干长度小于 1.5 mm 的部分相干光束,频率可达 312 Hz,满足实时生成部分相干光的速度要求. 并且,采用双孔干涉实验对所生成部分相干光的相干长度准确性进行了检测. 实验结果表明,生成的相干长度为 0.15 mm 和 1.5 mm 部分相干光束,其相干度均方根误差分别为 0.022 011 和 0.020 883,最大差别峰值分别为 0.074 325 和 0.072 998. 可见,本方法可以实时快速准确地生成相干部分相干光束.

参考文献

- [1] BORAH D K, VOELZ D G. Spatially partially coherent beam parameter optimization for free space optical communication [J]. *Optics Express*, 2010, **20**(18): 20746-20758.
- [2] KE Xi-zheng, HAN Mei-miao, WANG Ming-jun, *et al.* Spreading and wander of partially coherent beam through atmospheric turbulence in aslant path [J]. *Acta Photonica Sinica*, 2015, **44**(3): 0306001.
柯熙政, 韩美苗, 王明军, 等. 部分相干光在大气湍流中斜程传输路径上的展宽与漂移 [J]. *光子学报*, 2015, **44**(3): 0306001.
- [3] DENG P, KAVEHRAD M, LIU Z, *et al.* Capacity of MIMO free space optical communications using multiple partially coherent beams propagation through non-kolmogorov strong turbulence [J]. *Optics Express*, 2013, **21**(13): 15213-15229.
- [4] JENKINS M H, LONG J M, THOMAS K G, *et al.* Multifilter phase imaging with partially coherent light [J]. *Applied Optics*, 2014, **53**(16): D29-D39.
- [5] LI Xi-yu, GAO Xin, TANG Jia, *et al.* Amelioration of intensity correlation array towards high-orbit satellite imaging [J]. *Acta Photonica Sinica*, 2015, **44**(6): 0611008.
李希宇, 高昕, 唐嘉, 等. 面向高轨目标成像的强度相干阵列优化 [J]. *光子学报*, 2015, **44**(6): 0611008.
- [6] GAO Ming, LIU Yan-qing, WANG Fei, *et al.* Spread and wander characteristics of polarized and partially coherent laser beam propagated in turbulent atmosphere [J]. *Acta Photonica Sinica*, 2014, **43**(10): 1001002.
高明, 刘彦清, 王菲, 等. 偏振部分相干激光波束在湍流大气中传输的扩展和漂移 [J]. *光子学报*, 2014, **43**(10): 1001002.
- [7] CHEN C Y, YANG H M, ZHOU Z, *et al.* Effects of source spatial partial coherence on temporal fade statistics of irradiance flux in free-space optical links through atmospheric turbulence [J]. *Optics Express*, 2013, **24**(21): 29731-29743.
- [8] FRIBERG A T, GAO C Y, EPPICH B, *et al.* Generation of partially coherent fields with twist [C]. SPIE, 1997, **3110**: 317-328.
- [9] AMBROSINI D, SCHIRRIPA G, PAOLETTI D, *et al.* High-precision digital automated measurement of degree of coherence in the Thompson and Wolf experiment [J]. *Pure and Applied Optics*, 1998, **7**: 933-939.
- [10] SHIRAI T, WOLF E. Coherence and polarization of electromagnetic beams modulated by random phase screens and their changes on propagation in free space [J]. *Journal of the Optical Society of America A*, 2004, **21**(10): 1907-1916.
- [11] WANG F, LIU X, YUAN Y, *et al.* Experimental generation of partially coherent beams with different complex degrees of coherence [J]. *Optics Letters*, 2013, **38**(11): 1814-1816.
- [12] NELSON C, AVRAMOV-ZAMUROVICA S, KOROTKOVAB O, *et al.* Measurements of partially spatially coherent laser beam intensity fluctuations propagating through a hot-air turbulence emulator and comparison with both terrestrial and maritime environments [C]. SPIE, 2014, **8610**: 86100T.
- [13] AVRAMOV-ZAMUROVIC S, MALEK-MADANI R, NELSON C, *et al.* Probability Density Function (PDF) of intensity of a stochastic light beam propagating in the turbulent atmosphere [C]. SPIE, 2012, **8238**: 82380J.
- [14] SHIRAI T, KOROTKOVA O, WOLF E, *et al.* A method of generating electromagnetic Gaussian Schell-model beams [J]. *Journal of Optics A: Pure and Applied Optics*, 2005, **7**: 232-237.
- [15] NI Xiao-long, LIU Zhi, KONG Yue, *et al.* Parallel implementation of liquid crystal atmosphere turbulence simulator wavefront generation based on GPU [J]. *Chinese Journal Lasers*, 2014, **26**(3): 031011.
倪小龙, 刘智, 孔悦, 等. 基于 GPU 的液晶大气湍流模拟器波面生成的并行实现 [J]. *强激光与粒子束*, 2014, **26**(3): 031011.
- [16] BORN M, WOLF E. Principles of optics [M]. 7th Ed. Cambridge University Press, 1999, Chapter 10.