

一种基于新型查表方法的统一计算设备架构并行计算全息算法

蒋晓瑜 丛彬 裴闯 闫兴鹏* 赵锴

装甲兵工程学院信息工程系,北京 100072

摘要 为解决点源法计算全息速度较慢的问题,提出了一种新的查表算法,命名为三角函数查表法(T-LUT算法)。该算法是基于点源法基本的数学公式,通过一系列数学近似与恒等变换,生成了一种纯相位查找表,该查找表具有三维特性,并具有生成速度快、精度高、占用内存少等特点,克服了点源法重复计算相位的缺点。同时采用统一计算设备架构(CUDA)并行计算在图形处理器(GPU)上加以实现,并进行了三次并行优化。在算法的验证与对比实验中,采用单显卡(GPU显卡)实现T-LUT算法,在不牺牲全息图再现像质量的前提下,成功地将点源法计算全息的速速大幅度提升。实验发现在不同的物空间采样点数量的情况下,速度相对于点源法GPU运算提升30倍至近千倍不等。

关键词 全息;三角函数查表算法;查表法;统一计算设备架构;并行计算

中图分类号 O438.1

文献标识码 A

doi: 10.3788/AOS201535.0209001

A New Look-up Table Method of Holographic Algorithms Based on Compute Unified Device Architecture Parallel Computing

Jiang Xiaoyu Cong Bin Pei Chuang Yan Xingpeng Zhao Kai

Department of Information Engineering, The Academy of Armored Forces Engineering, Beijing 100072, China

Abstract We propose a new look-up table algorithm, named trigonometric look-up tables (T-LUT) method to increase the speed of coherent ray trace (CRT) algorithm in generating holograms. The proposed algorithm is based on the original ray tracing method. By using a series of mathematical approximation changes and identical transformations, we generate a pure phase look-up table. This look-up table has three-dimensional features, fast generating speed, high precision, less memory capacity and so on. The new method overcomes CRT's shortcomings in computing the phase repeatedly. We implement T-LUT on graphics processing unit (GPU) using parallel computing by compute unified device architecture (CUDA), and optimize it three times. After algorithm optimizing, a series of experiments are carried out to verify the results by using a single GPU card. Our results indicate that the T-LUT algorithm can effectively shorten the computing time without sacrificing the quality of hologram reconstruction. Due to the different quantities of spatial samples, the speed has increased 30 to nearly a thousand times compared with original ray tracing algorithm implemented on GPU.

Key words holography; trigonometric look-up table method; look-up table method; compute unified device architecture; parallel computing

OCIS codes 090.1760; 090.1995; 090.2870

1 引言

近年来,计算全息克服了光学全息对实验条件要求高的缺点,可以灵活地生成各种所需的全息图,因而

收稿日期: 2014-08-04;收到修改稿日期: 2014-10-08

基金项目: 全国优秀博士学位论文作者专项资金资助项目(201432)、北京市自然科学基金(4152049)

作者简介: 蒋晓瑜(1967—),男,博士,教授,主要从事信息光学及机器视觉方面的研究。

E-mail: jiangxiaoyu2007@gmail.com

*通信联系人。E-mail: ganxp02@gmail.com

在理论与应用研究各方面得到广泛的重视。但是计算全息所涉及的计算量大,尤其是在计算具有复杂表面及深度信息的三维物体时,其生成速度难以满足实时显示的要求。点源法又称为相干光线追迹法(CRT),是计算三维物体全息图较为通用的方法,它将所有物点发出的光线在全息面上与参考光干涉叠加从而获得计算全息图^[1]。该方法操作灵活,可以获得较高质量的重建像,且重建像具有真三维特性。为了提高计算速度,一系列基于点源法的查找表优化算法相继被提出。最初,由Lucente^[2]提出的查找表算法(LUT)是通过预先计算,以减少点源法的计算负荷,但是这种查找表在三维计算全息上将占用超大的内存,且没有高效的寻址方法,导致计算速度提升有限。鉴于此, Kim等^[3]实现了一种新的查找表(N-LUT)的方法,以减少内存使用量,但减少之后的查找表仍然是吉字节级别的,近期又对N-LUT方法进行了改进,改进后命名为RLE-based N-LUT法^[4],该方法可以将查找表的内存再次降低一个数量级。2009年, Pan等^[5]提出的分裂查找表(S-LUT)法将三维查找表的内存降低了很多,同时简化了计算公式。但在查找表中存储的数据为双精度数据,该查找表在内存占用上仍具有提升空间。2013年, Jia^[6]等提出了一种名为压缩查找表(C-LUT)的方法,这种查找表在离线计算中不计算深度信息,即达到了查找表内存大小与深度信息无关,大大减小了三维查找表的内存。但是由于减小了离线操作的复杂度,导致在线计算全息图时需要加入深度信息,相比于以上其他几种查找表,并没有实现在算法复杂度上的精简,尽管依靠图形处理器(GPU)硬件加速,使得C-LUT计算全息的速度并不慢,但是单从算法本身的角度来说,优化效果还有很大提升空间。

此外,国内外许多学者从不同角度对计算全息的算法进行了研究,取得了一定的研究成果,推动了计算全息的发展^[7-12]。本文提出三角函数查找表(T-LUT)法,将原始点源法的公式进行一系列数学推导,包括三角函数恒等变换与相关的数学近似,在查找表离线制作中,同样选择将深度信息加入到离线计算中,降低了在线计算时算法的复杂度,同时将查找表的内存理论上降低到与S-LUT相当的程度。在计算查找表时,采用双精度并行计算,生成速度快、数据准。生成查找表时,采用单精度数据存储,在实际内存占用方面又降低了一半。结合统一计算设备架构(CUDA)编程,将并行思想引入T-LUT算法,对点源法进行了并行优化,实验中采用单显卡GPU计算。结果表明,三角函数查找表(T-LUT)法相对于点源法GPU运算,速度提升30倍至近千倍不等。

2 三角函数查找表法

点源法的基本数学表达式为

$$I(x_h, y_h) = \sum_{j=0}^{N-1} A_j \cos \left[\frac{2\pi}{\lambda} \sqrt{(x_j - x_h)^2 + (y_j - y_h)^2 + z_j^2} \right], \quad (1)$$

式中 $I(x_h, y_h)$ 表示全息面上每点的复振幅, N 表示物空间采样点数量, λ 是参考光波长, A_j 表示物点的振幅, (x_j, y_j, z_j) 是物空间内物点坐标值,具体空间坐标系如图1所示。

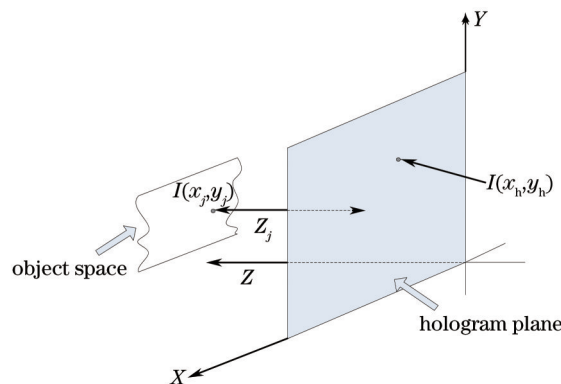


图1 空间位置关系

Fig.1 Spatial relationship between an object point and a hologram point

在实验中要想获得效果较好的再现图像,需合理调整物空间采样间隔和空间光调制器加载全息图的采样间隔之间的比例。因此 x_j 、 y_j 的取值间隔是具有物理意义的物空间的采样间隔 p 。 x_h 、 y_h 的取值间隔为空间光调制器加载全息图的间隔 q 。为了便于公式的后续推导,这里将 x_j 、 y_j 以及 x_h 、 y_h 进行归一化处理,

即令 $x_j = pX_j$, $y_j = pY_j$; $x_h = qX_h$, $y_h = qY_h$, 其中 X_j, Y_j 以及 X_h, Y_h 为不具有物理意义的数学坐标。则点源法原始数学表达式可调整为

$$I(x_h, y_h) = \sum_{j=0}^{N-1} A_j \cos \left[\frac{2\pi}{\lambda} \sqrt{\Delta X^2 + \Delta Y^2 + z_j^2} \right], \quad (2)$$

式中 $\Delta X^2 = (pX_j - qX_h)^2$, $\Delta Y^2 = (pY_j - qY_h)^2$ 。在光学再现系统中, 存在重建距离 $z_j \gg \Delta X$, $z_j \gg \Delta Y$, 由二项式近似可得 $\sqrt{\Delta X^2 + \Delta Y^2 + z_j^2} = z_j + \frac{1}{2z_j}(\Delta X^2 + \Delta Y^2)$, 因此(2)式可以化简为

$$I(x_h, y_h) = \sum_{j=0}^{N-1} A_j \cos \left\{ \frac{2\pi}{\lambda} \left[z_j + \frac{1}{2z_j}(\Delta X^2 + \Delta Y^2) \right] \right\}. \quad (3)$$

由(3)式可见, 每计算全息图中的一个像素点 $I(x_h, y_h)$, 都需要计算所有物点对该全息点的贡献: $A_j \cos \left\{ \frac{2\pi}{\lambda} \left[z_j + \frac{1}{2z_j}(\Delta X^2 + \Delta Y^2) \right] \right\}$, 而该贡献中的相位部分 $\cos \left\{ \frac{2\pi}{\lambda} \left[z_j + \frac{1}{2z_j}(\Delta X^2 + \Delta Y^2) \right] \right\}$, 对于全息面上的每个像素点计算来讲, 是不断进行重复计算的。这里提出的 T-LUT 查找表是一种纯相位查找表, 将该重复计算的相位部分预先计算存入表格。即

$$T_{\text{able}} = \cos \left\{ \frac{2\pi}{\lambda} \left[z_j + \frac{1}{2z_j}(\Delta X^2 + \Delta Y^2) \right] \right\}. \quad (4)$$

实验中空间光调制器固有的采样间隔为 $8 \mu\text{m}$, 即 $q = 8 \mu\text{m}$ 。同时为了扩展查找表的应用范围, 便于分析实验结果, 将物空间的采样间隔 p 设定为 5~10 倍全息面采样间隔, 即 $p = aq$ (其中 a 为整数, 取值范围为 $5 \leq a \leq 10$)。令 $\Delta x = aX_j - X_h$, $\Delta y = aY_j - Y_h$ 。在光学再现系统中, 可令参考光波长 λ 是固定的, 而重建距离 z_j 可人为选取。为了简化计算, 这里将 z_j 取为 λ 的整数倍, 即 $\frac{2\pi}{\lambda} z_j$ 是 2π 的整数倍。由于 λ 是纳米量级, z_j 的空间采样为毫米量级, 因此在计算全息过程中能够将 z_j 精确地取为 λ 的整数倍。根据三角函数性质, (4) 式可简化为

$$T_{\text{able}} = \cos \left[c_1 (\Delta x^2 + \Delta y^2) \right]. \quad (5)$$

式中 $c_1 = \frac{\pi q^2}{\lambda z_j}$ (对同一深度的物点在计算全息时, z_j 为常数)。从(5)式中可以得出: 在生成的全息图大小不变的情况下, 随着空间采样点的增加, 即使是二维表格, 所需内存也将达兆字节级别, 如果再考虑深度信息这一变量, 则表格的内存又会增加一个数量级, 所以继续化简(5)式为

$$T_{\text{able}} = \cos(c_1 \Delta x^2) \cos(c_1 \Delta y^2) - \sin(c_1 \Delta x^2) \sin(c_1 \Delta y^2). \quad (6)$$

由于在(4)式简化到(5)式的变化中, 将多余的相位 $\frac{2\pi}{\lambda} z_j$ 消除了, 使得(6)式的复杂度相比于 N-LUT 查找表的最终表达式更加简便, 这也是这里查找表生成速度更快的原因之一。通过数学分析方法, (6)式可以进一步简化。为了便于算法的数学推导, 将查找表的物点数以及生成的全息图大小均选取为 $N \times N$ (假定 N 足够大, 满足实验所需), 即 X_j, Y_j, X_h, Y_h 的取值范围均为 $[0, N]$ 。因此, $|\Delta x|, |\Delta y|$ 取值范围均在区间 $[0, aN]$ 中, 且可取到该区间内所有整数。同理, (6)式中的 Δx^2 与 Δy^2 在寻址区间上也是完全相同的, 即 $\sin(c_1 \Delta x^2)$ 与 $\sin(c_1 \Delta y^2)$ 以及 $\cos(c_1 \Delta x^2)$ 与 $\cos(c_1 \Delta y^2)$ 只需计算其中之一即可。所以最终需要计算的二维查找表为

$$\begin{cases} T_{\text{able1}} = \cos(c_1 \Delta t^2) \\ T_{\text{able2}} = \sin(c_1 \Delta t^2) \end{cases}, \quad 0 \leq t \leq N. \quad (7)$$

在实验中, 选取 $a = 10$, 即真实物空间采样间隔为全息面采样间隔的 10 倍。选取 $N = 2048$, 即一个物面 (z_j 相同) 的物点数在 $2048 \text{ pixel} \times 2048 \text{ pixel}$ (4×10^6) 之内, 生成的全息图像素值大小在 $2048 \text{ pixel} \times 2048 \text{ pixel}$ 之

内的所有二维实验数据都可以通过(7)式的表格进行寻址、计算(包括一些复杂表面物点分布以及生成各种长宽比的矩形全息图),且寻址方式简单快速,(7)式中两个分表,可以同时进行 $|\Delta x|$ 、 $|\Delta y|$ 的寻址。每次计算生成全息图时,不需要再次生成表格。

但是当物空间采样点为三维点集时,则 $c_1 = \frac{\pi q^2}{\lambda z_j}$ 中的深度信息 z_j 不能够将其再看做常量,所以需要进一步对三维表格进行推导。可令 $c = \frac{\pi q^2}{\lambda}$ 为常数,则(7)式可转化为

$$\begin{cases} T_{\text{able1}} = \cos\left(\frac{c}{z_j} \Delta t^2\right) \\ T_{\text{able2}} = \sin\left(\frac{c}{z_j} \Delta t^2\right) \end{cases}, \quad 0 \leq t \leq N. \quad (8)$$

至此,成功地将三维表格的三维变量拆分为两组二维变量,有效地解决了三维查找表内存过大的问题,(8)式即为最终确定下来的三维三角函数查找表(T-LUT)。设物空间的平面维度 $W = 1024 \text{ pixel}$, $H = 1024 \text{ pixel}$ (平面采样点),深度维度为50层($D = 50$),全息图大小限制为 $X_h = 1024 \text{ pixel}$, $Y_h = 1024 \text{ pixel}$,设采样点间隔与空间光调制器(SLM)加载像素的尺寸比为 $a=10:1$ 。经计算,LUT算法查找表内存大小约为23.6GB;N-LUT算法的查找表内存大小约为200 M;S-LUT算法的查找表内存大小约为8.6M;采用的T-LUT算法的查找表内存仅在4 M左右;由于C-LUT算法的查找表中不包含深度信息,因此C-LUT算法查找表内存占用最小,约为0.17 M。所提出的T-LUT算法在查找表内存大小上在已公布的众多查找表中仅次于C-LUT查找表,但是相比于C-LUT查找表,T-LUT查找表中存储了深度信息,从而在在线计算三维物体全息图的速度上具有优势。

3 算法在CUDA上的并行实现及优化

在T-LUT查表法推导过程中,得到了点源法计算全息图的常用表达式(3)式。在CUDA架构中,GPU显卡对双精度计算速度上支持不足,即使是最新的Tesla专业计算显卡对双精度计算的支持也远远比不上GTX系列中端显卡对单精度计算的支持^[13]。因此,这里采用的查找表法,将需要双精度计算的正弦,余弦等函数预先计算出来,回避了计算精度和计算速度之间的矛盾,且在并行计算部分将计算复杂度减少。

根据T-LUT的最终结果(8)式,可以将(3)式转化为

$$I(x_h, y_h) = \sum_{j=0}^{N-1} A_j \left(T_{\text{able1}}|_{\Delta x} \times T_{\text{able1}}|_{\Delta y} - T_{\text{able2}}|_{\Delta x} \times T_{\text{able2}}|_{\Delta y} \right). \quad (9)$$

(9)式的编程逻辑可以理解为:由一个已知数组(物空间采样值)通过一系列计算生成一个新的数组(全息图),满足已知数组中的所有元素都和新生数组中每个元素的生成有计算关系,且新生数组中每个元素之间不存在计算关系。分析该逻辑,(9)式存在两重并行优化。第一重并行在于如何并行计算出全息图中的一个点,即已知数组中的所有元素进行并行计算,然后得到新生数组中的一个元素。第二重并行在于如何并行的计算出全息图中的所有点,即将新生数组的所有元素并行求出。图2是对物点为 $256 \text{ pixel} \times 256 \text{ pixel}$,生成全息图大小为 $1024 \text{ pixel} \times 1024 \text{ pixel}$ 的双重并行程序(程序1)的逻辑流程图。

程序1满足了双重并行计算全息数组的实验要求,实现了远超GPU的速度优化。但由于在实验中全局内存中生成的缓存数据过大,严重影响全息图生成速度,且限制了空间物点的采样量。为此,对该程序进行了二次优化,本次优化将第一重并行优化为循环求和,将上述程序1存在全局内存的缓存数据大大减小后存储在数据存取速度快的寄存器上,并充分的利用了共享内存及其特性。在较多物点的全息图生成速度上又取得了进一步的提升,存储空间优化单重并行程序(程序2)流程图如图3所示。

程序2在GPU端计算全息数组的核心算法实现上采用单重并行,即并行的计算出全息数组中的所有元素。虽然只实现了一重并行要求,但对物点采样的数量没有了限制,且在多物点采样的情况下生成全息图的速度再次得到提高。

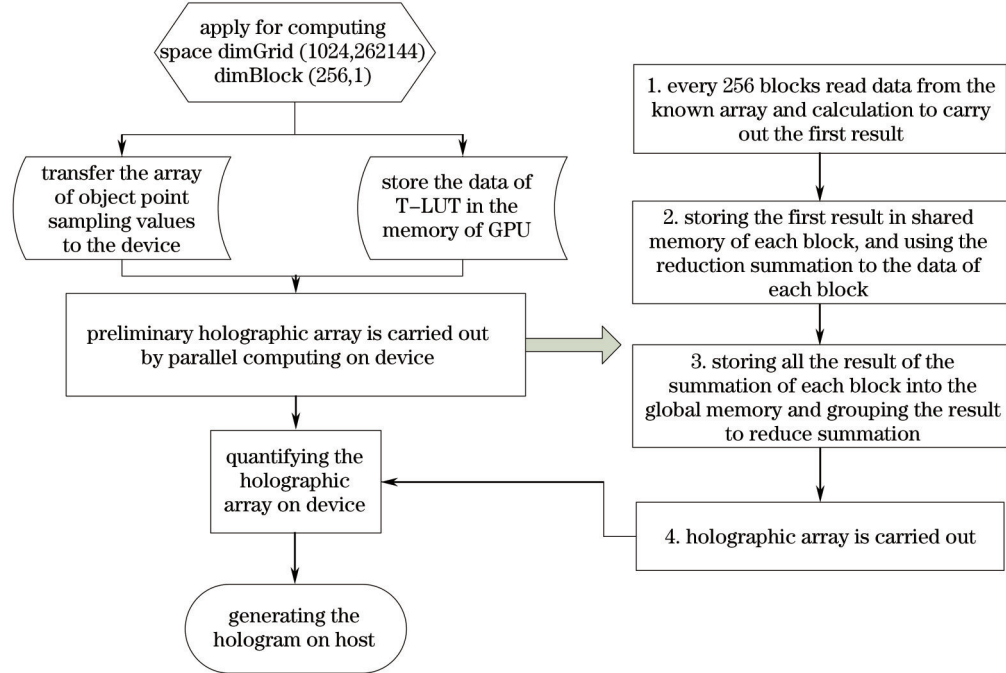


图 2 双重并行程序(程序 1)流程图

Fig.2 Flow chart of the first T-LUT by CUDA

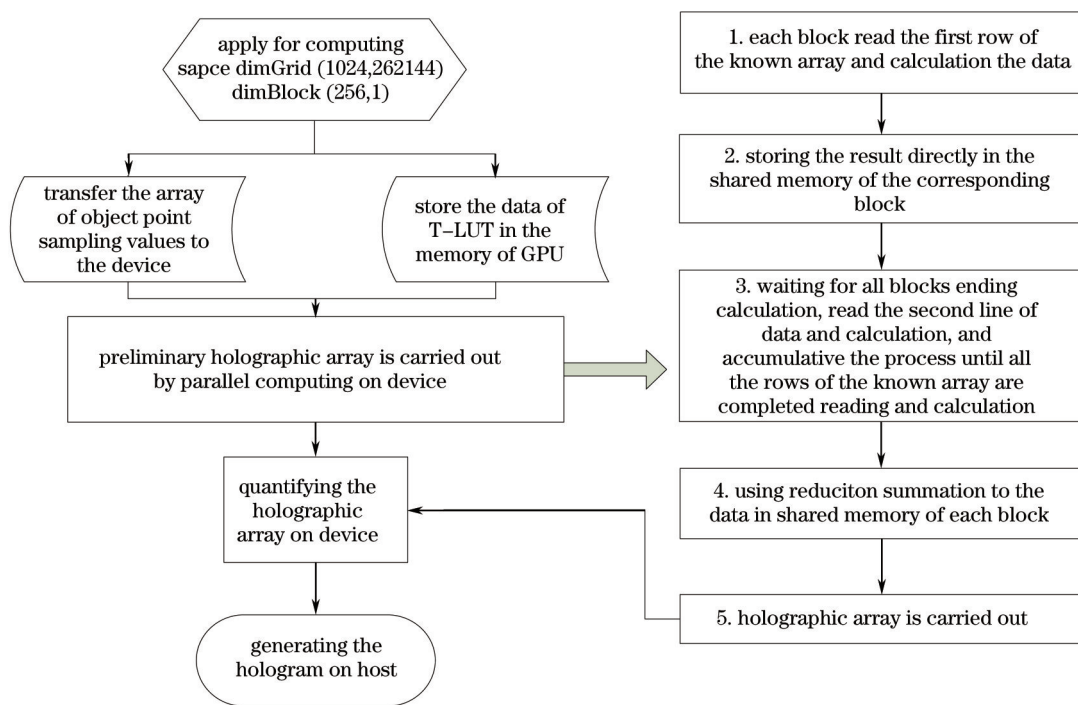


图 3 存储空间优化单重并行程序(程序 2)流程图

Fig.3 Flow chart of the second T-LUT by CUDA

程序 2 将所有缓存数据都存到寄存器和共享内存中,当循环的次数太多时,则仍有优化的空间。为此,对实验程序进行了第三次优化(程序 3),这次优化在缓存数据大小和多重并行之间进行权衡,根据数据量大,提供能够让 GPU 达到峰值计算能力的线程配比,又取得了一次速度优化。本次优化取得了速度上的提升,没有对物点采样产生限制,线程配比可根据物点采样的变化进行权衡,即权衡循环次数和全局内存访问时间,不存在统一的线程配比及流程图。

4 实验结果

这里实验采用的硬件平台为:处理器为 Intel Core i7-4930k,内存 8 GB,显卡为单块显卡,型号是 Nvidia GTX Titan Black,计算能力 3.5,内存 4 GB。软件平台为 Win7 操作系统,CPU 仿真实验软件为 Matlab 2013b,GPU 的应用程序编程接口(API)为 CUDA 5.0,其编译环境为 Visual studio 2010,Device 端调试软件为 Nvidia Nsight HUD Launcher 3.0。实验中搭建了光学全息再现平台,其规格如表 1 所示。

表 1 光学全息再现平台规格表

Table 1 Optical holography reconstruction platform parameters

Parameter	Value
Model of SLM	HEO 1080p SLM
Type of SLM	Pure phase modulator
Range of phase modulation	0~ 2π
Pixel size of SLM	8 μm
Wavelength of laser used in reconstruction system	633 nm
Sampling interval of object space	64~90 μm

实验中分别对不同物点采样,进行点源法(CRT)的 GPU 计算仿真,以及上述三种 T-LUT 算法的 CUDA 并行计算,统计了各种方法生成全息图的时间(全息图大小为 1024 pixel \times 1024 pixel)。其结果如表 2 所示。

表 2 全息图生成时间表(单位: s)

Table 2 Hologram generation time (unit: s)

Points of object space / 10^3	Time of Matlab simulation by GPU	Computation time of the first T-LUT by GPU	Computation time of the second T-LUT by GPU	Computation time of the third T-LUT by GPU
1	31	0.91	0.83	0.83
4	142	1.83	2.84	1.83
10	289	4.32	6.78	4.32
16	481	7.21	7.46	4.49
65	Out of memory	38.72	22.53	12.27
90	Out of memory	43.73	29.47	18.56
262	Out of memory	58.92	45.32	31.42

物点采样超过 2×10^4 之后,在 GPU 端的 Matlab 仿真实验将提示内存溢出,但根据 GPU 端前段数据的测试估计,当物点采样数达到 2.5×10^5 以上的时候,这里采用的 T-LUT 优化算法将达到相比于 GPU 原始点源法近千倍的加速比。T-LUT 最终优化算法在实验条件下,计算全息速度优于 S-LUT 以及 C-LUT 等其他算法,例如物空间采样点数为 90000 pixel 时,C-LUT 算法在计算全息(彩色)速度约为 200 s^[5],T-LUT 算法在同等实验条件下约为 60 s(18.56 \times 3 s)。

图 4 给出了由 GPU 仿真生成全息图的光学再现像和采用所提算法生成全息图的光学再现像的 CCD 图像对比。图(5)给出了采用所提算法生成的三维图像全息图光学再现像的 CCD 图像。

结构相似性指标测量(SSIM)是目前公认的较好衡量两幅灰度图像相似度的指标。通过对上述 CCD 采集的全息图光学再现像同原始图想比对,获得图 4(a)~(d)中 SSIM 分别为 0.6、0.58、0.46、0.47。上述实验结果证实了 T-LUT 算法在速度上具备优势,并且没有降低再现像质量。图 5 给出了采用所提算法生成的三维图像全息图光学再现像。其中图 5(a)是全息图在 $z_j = 1899$ mm 处的光学再现像,此时星星聚焦;图 5(b)是全息图在 $z_j = 1266$ mm 处的光学再现像,此时月亮聚焦;图 5(c)是全息图在 $z_j = 633$ mm 处的光学再现像,此时卡通人物聚焦。可见,所提算法可以在计算速度、内存等方面具备优势的情况下,有效地完成三维物体全息图的快速生成。

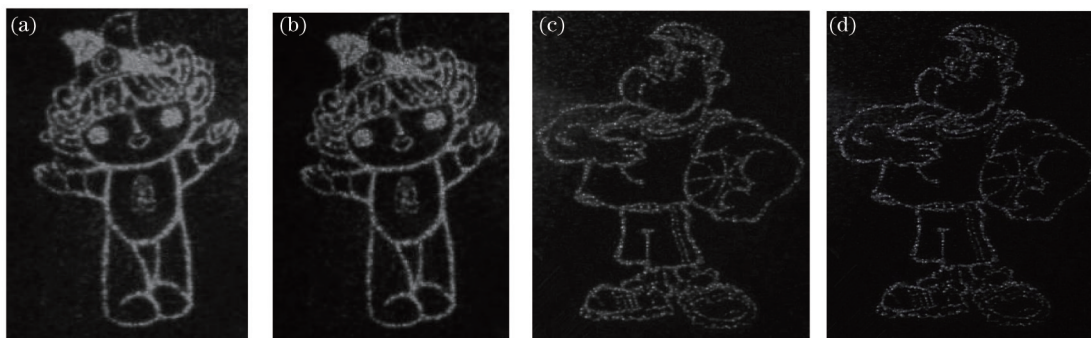


图 4 二维全息图再现像对比。(a)GPU CRT法再现像(福娃);(b)GPU T-LUT算法再现像(福娃);(c)GPU CRT法再现像(运动员);(d)GPU T-LUT算法再现像(运动员)

Fig.4 Optical reconstruction of 2D image. (a) Optical reconstruction by CRT method on GPU (fuwa); (b) optical reconstruction by T-LUT method on GPU (fuwa); (c) optical reconstruction by CRT method on CPU (athlete); (d) optical reconstruction by T-LUT method on GPU (athlete)

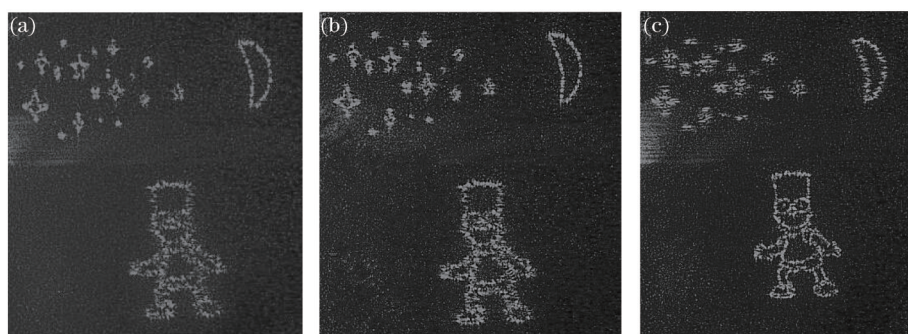


图 5 三维物体全息图光学再现像不同深度 CCD 采集图。(a) $z_f=1899$ mm; (b) $z_f=1266$ mm; (c) $z_f=633$ mm

Fig.5 Optical reconstruction of three-dimensional hologram in different depths acquired by CCD. (a) $z_f=1899$ mm; (b) $z_f=1266$ mm; (c) $z_f=633$ mm

5 结 论

本文提出了一种新的查找表(T-LUT)的计算全息算法,并成功地在 GPU 上并行实现,取得了良好的再现像效果和加速比,有效地解决了点源法计算全息计算量大、速度慢的缺点。同时横向比较其他基于查找表的计算全息算法,T-LUT算法在再现像质量和速度上均有优势。实验中的 GPU 平台是单卡 GPU 计算,在今后的实验中,拟将 T-LUT 算法在多卡 GPU 平台上实现,达到实时动态的三维全息图生成并再现^[4]。

参 考 文 献

- 1 Tang Chuang, Yao Baoli. Fast algorithm for generating hologram of a 3D object by employing connection between layers [J]. Journal of Optoelectronics·Laser, 2012, 23(11): 2180-2183.
唐 创,姚保利.基于分层技术的三维物体全息图的快速算法[J].光电子·激光,2013,23(11):2180-2183.
- 2 Mark Lucente. Interactive computation of holograms using a look-up table [J]. Electronic Imaging, 1993, 2(1): 28-34.
- 3 Seung-Cheol Kim, Eun-Soo Kim. Computational approaches for fast generation of digital 3D video holograms [J]. Chin Opt Lett, 2009, 7(12): 1083-1091.
- 4 S C Kim, E S Kim. Fast computation of hologram patterns of a 3D object using run-length encoding and novel look-up table methods [J]. Appl Opt, 2009, 48(6): 1030-1041.
- 5 Yuechao Pan, Xuewu Xu, Sanjeev Solanki, *et al.*. Fast CGH computation using S-LUT on GPU [J]. Opt Express, 2009, 17(21): 18543-18555.
- 6 Jia Jia, Yongtian Wang, Juan Liu, *et al.*. Reducing the memory usage for effective computer-generated hologram calculation using compressed look-up table in full-color holographic display [J]. Appl Opt, 2013, 52(7): 1404-1412.
- 7 N Masuda, T Ito, T Tanaka, *et al.*. Computer generated holography using a graphics processing unit [J]. Opt Express, 2006, 14(2):

603-608.

- 8 Tang Wenhua, Li Shuli, Wang Jun, *et al.*. A method of computer-generated hologram based on integral imaging [J]. Chinese J Lasers, 2014, 41(2): 0209022.
唐文华, 李述利, 王 君, 等. 一种基于集成成像生成计算全息图的方法[J]. 中国激光, 2014, 41(2): 0209022.
- 9 Li Junchang, Gui Jinbin, Lou Yuli, *et al.*. Study of creation algorithm of computer-generated hologram of diffuse reflection 3D object [J]. Laser & Optoelectronics Progress, 2013, 50(2): 020903.
李俊昌, 桂进斌, 楼宇丽, 等. 漫反射三维物体计算全息图算法研究[J]. 激光与光电子学进展, 2013, 50(2): 020903.
- 10 X Xu, X Liang, Y Pan, *et al.*. Development of full-color full-parallax digital 3D holographic display system and its prospects [C]. SPIE, 2013, 8644: 864409.
- 11 Dong Jianwen, Chen Dingchen, Pang Xiaoning, *et al.*. Spatial-domain computer generated holographic three-dimensional display [J]. Chinese J Lasers, 2014, 41(7): 0701001.
董建文, 陈定尘, 庞晓宁, 等. 空域计算全息三维成像技术[J]. 中国激光, 2014, 41(7): 0701001.
- 12 Huadong Zheng, Ying Jieyu, Tao Wang, *et al.*. Computer-generated kinoforms of real-existing full-color 3D objects using pure-phase look-up-table method [J]. Opt & Lasers in Eng, 2012, 50(4): 568-573.
- 13 NVIDIA. NVIDIA CUDA C Programming Guide [OL]. <http://docs.nvidia.com/cuda/cuda-C-programming-guide/index.html#abstract>. [2014-01-09].
- 14 N Takada, T Shimobaba, H Nakayama, *et al.*. Fast highresolution computer-generated hologram computation using multiple graphics processing unit cluster system [J]. Appl Opt, 2012, 51(30): 7303-7307.

栏目编辑: 何卓铭