

文章编号:1672-058X(2013)05-0037-06

三维场景相息图快速生成研究*

陈慧蓉^{1,2}, 付胜豪¹, 王元庆¹

(1. 南京大学 电子科学与工程学院, 南京 210093; 2. 芜湖职业技术学院 电气工程系, 安徽 芜湖 241006)

摘要:相息图以其衍射效率高, 无共轭像等优点而被广泛应用于全息三维动态显示中, 然而相息图计算量大, 生成速度慢而影响了它的实时动态显示; 鉴于此, 提出了一种三维场景相息图快速生成方法; 采用 OpenGL 进行物点离散采样, 逐点计算的优化查表算法进行相息图计算, 将相位因子拆成水平方向和竖直方向因子的乘积, 离线制作的查找表只需存储水平方向和竖直方向因子, 从而使存入 GPU 纹理内存的查找表空间大大减少, 并利用 CUDA 架构合理设计并行计算方案, 使相息图的运算速度进一步加速; 实验表明: 采用此方法能正确生成所需的相息图, 且计算速度比基于 CPU 的计算方法提高了 40 倍左右。

关键词:计算机全息; 相息图; 并行计算; 三维立体场景

中图分类号: TP391

文献标志码: A

0 引言

1948 年, D. Gabor^[1] 提出全息概念以来, 全息术得到了飞快的发展。随着计算机技术的发展, 出现了将计算机技术与全息技术相结合的一种新技术, 也就是计算全息即计算机制全息图 (Computer Generated Holography, CGH)^[1-2], 它克服了传统光学全息的缺点, 可以不受实验条件限制, 记录任何甚至不存在的物体的全息图。相息图是 L. B. Lesem^[3] 等提出的一种全新的计算制全息图形式。相息图具有衍射效率高, 光学再现时没有共轭像或多余的衍射级次等优点, 因此一经提出就得到广泛应用, 特别是利用计算机编码制作相息图, 空间光调制器承载加载相息图实现全息的三维动态显示, 一直是人们研究的热点, 要达到实时动态显示的效果, 关键问题是三维相息图的快速生成。

人们提出了一些提高相息图计算速度的方法。在软件算法的改进方面, 提出了查表算法^[4], 主菲涅尔波带算法^[5], 加速 FFT 算法^[6], 菲涅尔全息图快速算法^[7-9] 等。硬件方面, 提出基于现场可编程门阵列 (Field Programmable Gate Array, FPGA)^[10-11] 硬件开发技术和图像处理单元 (Graphic Processing Unit, GPU) 处理技术^[12-13]。这些方法存在着要么加速效果不明显, 要么对硬件要求较高。采用逐点计算的查表优化算法, 将相位因子的计算分解为水平因子和垂直因子计算结果的组合的形式, 大大减少计算量和存储量。并合理设计 GPU 的 CUDA 并行计算方案, 进一步加快计算速度。实验表明, 该相息图生成方法是行之有效的, 且运算速度比基于 CPU 的计算方法提高 40 倍左右, 随着所取的空间物点数增加, 运算速度提高更大。

收稿日期: 2013-02-08; 修回日期: 2013-03-15.

* 基金项目: 国家自然科学基金重点项目资助 (608320036); 国家质检公益性行业科研专项经费资助项目 (201110233).

作者简介: 陈慧蓉 (1972-), 女, 安徽芜湖人, 副教授, 硕士, 从事全息图像生成与自动控制方面研究.

1 三维场景相息图生成原理

三维场景相息图的生成流程图如图 1 所示。首先要进行三维场景建模,抽样得到三维场景虚构物体的离散样点分布。本文采用计算机进行三维建模,OpenGL(Open Grapgics Library)语言进行三维数据处理。具体方法是:三维立体场景是由 OpenGL 在虚拟坐标中绘制,利用 glReadPixels() 函数在深度缓冲区读取深度信息,获取平面坐标,再利用 gluUnProject() 函数将平面坐标转换为对应三维立体场景的、相对于视点的三维坐标,最后将其修正为实际坐标并进行规整,为相息图计算获取空间物点集数据。

获得三维物体的物点空间信息后,可以开始相息图的计算,计算物光波到相息面光波的复振幅分布。空间物体可看做是 N 个物点的集合,设某一物点 n ,其坐标为 (x_n, y_n, z_n) ,发出的漫射光均匀照射在相息图平面 $X-Y$ 上,物点 n 到相息面 (x, y) 点的距离为 $r_n = \sqrt{(x_n - x)^2 + (y_n - y)^2 + z_n^2}$,物点 n 的衍射光波振幅和随机相位分别为 a_n, φ_n 从物体上各个点发出的光传播一段距离后,在相息图平面上复振幅可表示:

$$u(x, y) = C \sum_{n=1}^N u_n(x, y) = C \sum_{n=1}^N a_n \exp(jkr_n + j\varphi_n) \quad (1)$$

式(2)中 k 为波数。设 $A(x, y)$ 和 $\varphi(x, y)$ 分别表示整个物体的物光波在相息面上的振幅和相位,则式(1)也可表示:

$$u(x, y) = A(x, y) \exp j\varphi(x, y) \quad (2)$$

舍去振幅信息令 $A(x, y) = 1$,取复振幅的相位信息,相息图平面上光波的位相信息:

$$\varphi(x, y) = \tan^{-1} \left(\frac{\sum_{n=1}^N a_n \sin(\varphi_n)}{\sum_{n=1}^N a_n \cos(\varphi_n)} \right) \quad (3)$$

由于位相的周期性,故在具体计算物光波的相位时,用如下所示的式(4):

$$\varphi_k(x, y) = \text{mod } 2\pi[\varphi(x, y)] \quad (4)$$

相位编码时,应用式(4),也就是提取出透过率函数中的相位信息,去模数 2π 的余数作为相息图透过率。最后,经过量化和归一化为需要的灰度级,形成一幅完整的三维场景相息图。将相息图输入至相位型空间光调制器,用一束平行光照射,调制的出射光在空间成像。由于不需要参考光的引入,再现像时也就没有冗余像,得到单一的衍射级,成像像面清晰,衍射效率高。

2 逐点计算的查表优化算法

逐点计算的方法适合三维立体场景的相息图制作,且操作灵活,计算精度高,本文在逐点计算方法的基础上,为了减少计算时间和降低相位因子表格的存储量,对此方法进行了优化。

对式(1)进行菲涅尔近似得:

$$u(x, y) = \sum_{n=1}^N a_n \exp(jkr_n + j\varphi_n) = \sum_{n=1}^N a_n \exp(j\varphi_n) \exp \left[jk \left(z + \frac{(x_n - x)^2 + (y_n - y)^2}{2z} \right) \right] \quad (5)$$

令水平方向因子:

$$H(x) = \exp \left[jk \left(z + \frac{(x_n - x)^2}{2z} \right) \right] \quad (6)$$

竖直方向因子:

$$V(y) = \exp \left(j \frac{(y_n - y)^2}{2z} \right) \quad (7)$$

将式(6)、式(7)带入式(5)得:

$$u(x,y) = \sum_{n=1}^N a_n \exp(j\varphi_{r_n}) H(x) V(y) \tag{8}$$

从式(5)可以看出,相息图计算花费时间在平方和平方根的计算,对于尺寸为 $X \times Y$ 的相息图,计算每个物点的复振幅分布需要乘法运算 $5XY$ 次, n 个物点也就是要 $5nXY$ 次,计算量庞大。

逐点计算方法大量计算量耗在了计算各物点到相息面上的各像素点的距离和相位上,为了加快运算速度,需离线计算好计算量大的相位因子部分,建立起相位因子查找表。这样在线运行时可通过查表来减少计算量。由式(8)可以看出,相位因子可以拆成相乘的水平方向 $H(x)$ 和竖直方向 $V(y)$ 的两个因子,故查找表中只需将水平方向 $H(x)$ 和竖直方向 $V(y)$ 因子存入设备存储区域即 GPU 的内存区域,这样存储的表格数据量大大下降。设场景深度为 K ,存储量由 $X \times Y \times K$ 下降为 $(X + Y) \times K$,也就解决了 GPU 内存有限的问题,为 CUDA 并行计算加速相息图生成提供有力的保证。直接由式(8),计算光波到相息面光波的复振幅分布次数为 $2nXY$ 。

通过对相息图的计算过程进行分析,发现有大量关于水平方向和竖直方向的冗余计算,去除这些冗余,就可以进一步提高相息图的运算速度。具体做法:首先计算出同一列的复振幅分布,因为从式(6)看出,对于空间同一列的物点,水平因子是相同的,这样计算次数为 $XY + X$;总的计算光波到相息面光波的复振幅分布为各列物点在相息图上的复振幅叠加,使相息图总的计算次数为 $nX + XY$,与原来的计算次数 nXY 相比,计算速度明显提高。

3 CUDA 并行计算设计

FPGA 技术设计成本高开发周期长^[14],GPU 价格低廉,拥有大量运算单元可实行高效并行计算,因此是相息图进行大量数据运算的首选硬件开发技术。CUDA 是 NVIDIA 公司于 2007 年 6 月推出的通用并行计算架构,可充分利用 GPU 的大规模数据并行计算能力使运算速度加快。故本文在硬件上基于 GPU 合理设计 CUDA 架构实现并行计算方案。

并行计算流程图如图 2 所示,首先离线制作查找表,并将查找表数据从 CPU 拷贝到 GPU 供 CUDA 程序调用,触发 CUDA 核函数,并行计算出各空间物点发出的光波在相息图各像素点处产生的光波的复振幅,然后采用归约求和的方法叠加计算出相息图各像素点处最终的光波复振幅。可见,基于 GPU 的 CUDA 并行加速计算关键是并行块与线程的合理分配,合理设计 CUDA 核函数。



图 1 三维场景相息图生成流程图

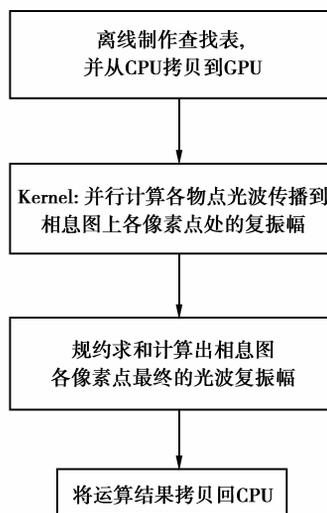


图 2 并行计算流程图

对于 CUDA 核函数, Kernel 设计为二维线程块, 块数为 $X * Y$ (相息图分辨率), 每个块就代表相息图的一个像素点, 每个块中的线程数为 T (注意 T 应为 2 的整数次幂, 这是为了方便归约求和), 每个线程代表一个物点, 在线程内计算出空间物点在该像素点处的复振幅。这样就可以并行计算出空间各物点光波传播到相息图上各像素点处的复振幅。

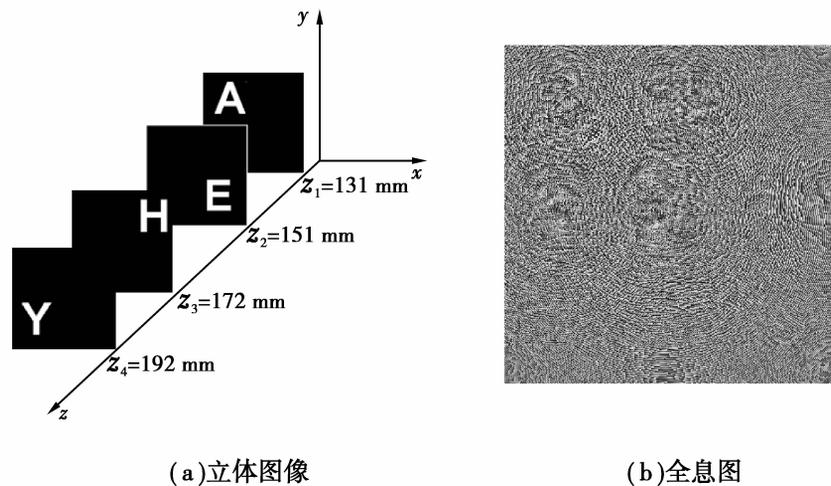


图 3 立体图像及其全息图

将同一像素点处的所有物光波叠加即得到三维场景在相息图各像素点处的复振幅, 对应的也就是将各个块内所有线程运算出的结果分别进行叠加求和。本文采用方法为在每个块内开辟 T 个共享内存 $cach[]$, 用来存储 T 个线程运算结果, 利用 GPU 共享内存存在同一个块内操作具有较高运算效率的特性, 采用归约求和方法在一个线程上迭代完成共享内存中 T 个数据叠加。这样整个空间场景发出的光波在相息图各像素点处的复振幅分布就被计算出来。

由于每个线程将 $cach[]$ 中的两个值加起来, 这样就有 $T/2$ 个线程并行地进行计算, 并行相加后将结果存回 $cach[]$, 显然并行计算比单线程迭代计算效率要高的多。完成这个步骤后所得结果数据量就是原来的一半, 在下一步中再对这一半数据执行同样的操作, 执行 $\log_2(T)$ 个步骤后, 就能得到 T 个数据叠加即 $cach[]$ 中所有的数据总和存入 $cach[0]$ 中。可见, 计算时间与数组长度成正比, 数组长度越大, 归约求和的方法提高计算时间的效果越明显。

4 实验验证

采用本文提出的算法进行相息图计算程序的编写, 在 PC 机 (处理器为英特尔 Core2Duo E5300, 内存 2 GB, 英伟达 GT240 显卡; WIN7 操作系统, VS2010 编译环境, CUDA4.0 通用开发包) 上进行实验。首先进行本论文所介绍的相息图生成方法的可行性研究, 三维场景如图 3 所示, 空间中四张图像距离相息面分别为 131 mm、151 mm、172 mm、192 mm。采用本方法实现不同深度再现像如图 4 所示。全息图尺寸为 $256 * 256$ 。

最后, 研究全息图尺寸固定, 在满足空间物点采样间隔及相息图采样间隔的条件下, 选取不同的空间物点数, 对基于 CPU 的传统相息图计算方法及本文所提的方法的运算速度进行对比。全息图尺寸为 $512 * 512$, 空间物点数分别为 100、300、1200、3 000 点得到实验数据进行拟合, 所得比较曲线如图 5 所示。从实验数据及比较曲线可以看出, 物点数在 1 200 点时, 相息图的计算速度提高了 40 倍左右, 随着空间物点数的增多, 计算速度提高更多。

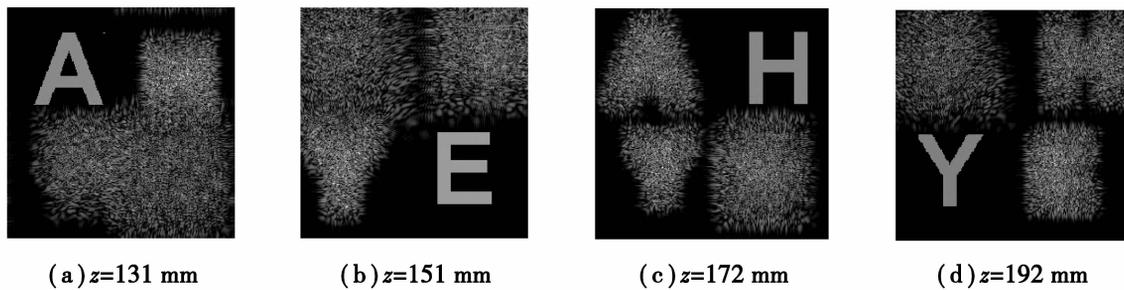


图4 全息图在不同深度上的再现像

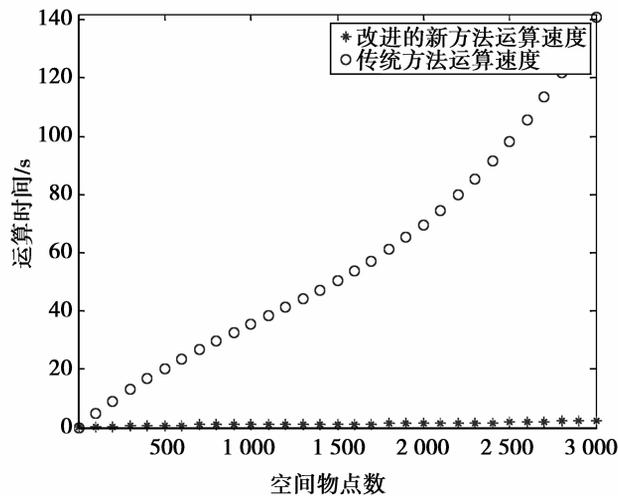


图5 运算速度测试曲线比较

5 结 论

本文介绍了一种三维场景相息图生成方法,其在硬件上采用 CUDA 架构实现并行计算,软件上采用对逐点计算的查表算法进行优化,将相位因子拆成水平方向和竖直方向的两个因子乘积,使存储的表格数据量大大下降,为并行计算提供了保障。实验表明,此方法能正确生成所需的相息图,且测试曲线也表明此方法使相息图的计算速度比基于 CPU 的相息图计算方法提高了 40 倍左右,空间物点数越多,计算速度提高越明显。真正解决了相息图制作过程中出现的计算量大,时间长的问题。

参考文献:

- [1] LEE S. Hampled fourier transform hologram generated by computer[J]. *App. l Opt.*, 1970, 9(3): 639-643
- [2] 虞祖良, 金国藩. 计算机全息图[M]. 北京: 清华大学出版社, 1984
- [3] LESEM L B, HIRSCH P M, JORDAN J A. Generation of discrete point holograms[J]. *J. Opt. Soc. Am.*, 1968, 58: 729A
- [4] LUCENTE M. Interactive computation of holograms using a look-up table[J]. *J Electron Imaging*, 1993, 2(1): 28-34
- [5] 张晓洁, 刘旭, 陈晓西. 利用菲涅尔波带法计算三维全息[J]. *光电工程*, 2004, 31(12): 58-60
- [6] ITO T, SHIMOBABA T. One-unit system for electroholography by use of a special-purpose computational chip with a high-resolution liquid-crystal display toward a three-dimensional television[J]. *Optics Express*, 2004, 12(9): 1788-1793
- [7] 金洪震, 李勇, 王辉. 利用差分方法实现相息图快速计算[J]. *光电工程*, 2001, 28(4): 30-33
- [8] MATSUSHIMA K. Fast computation of Fresnel holograms employing difference[J]. *App. l Opt.*, 2000, 39(35): 6587-6594

- [9] 李勇,许富洋,金洪震.一种菲涅尔全息图的快速算法[J].光子学报,2010,39(3):529-531
- [10] ITO T,MASUDA N,YOSHIMURA K,et al. A special purpose computer for electroholography HORN-5 to realize a real-time reconstruction[J]. Opt. Express,2005,13(6):1923-1932
- [11] 简献忠,张晗,范建鹏,周志刚.基于FPGA技术的计算机全息研究[J].激光技术,2012,36(1):26-28
- [12] MASUDAN,ITO T,TANAKA T,et al. Computer generated holography using a graphics processing unit[J]. Opt. Express,2006,14(2):603-608
- [13] AHRENBERG L,BENZIE P,MAGNORM,et al. Computer generated holography using parallel commodity graphics hardware[J]. Opt. Express,2006,14(17):7636-7641
- [14] THOMAS L,MATS G. A hardware acceleration platform for digital holographic imaging[J]. Signal Processing Systems,2008,52(3):297-311

Research on Fast Kinoform Generation for Three-dimension Scene

CHEN Hui-rong^{1,2}, FU Sheng-hao¹, WANG Yuan-qing¹

(1. School of Electronic Science and Engineering, Nanjing University, Nanjing 210093, China;

2. Department of Electrical Engineering, Wuhu Vocational College of Technology, Anhui Wuhu 241006, China)

Abstract: Kinoform, with the advantage of high efficient diffraction and non-conjugate image and so on, is widely used in three-dimension dynamic display of holography, however, the kinoform, with large calculation and slow generation speed, affects its real-time dynamic display. According to this, a kind of fast kinoform generation method for three-dimension scene is proposed. By using OpenGL to conduct discrete object point sample collection, by performing kinoform calculation through optimized look-up table algorithm of pointwise calculation, phase factor is decomposed into the product of horizontal direction and vertical direction, offline-made search table only needs to store the factors of horizontal direction and vertical direction, so that the space of search table stored in CPU texture memory is largely reduced, CUDA framework is rationally used to design and calculate the scheme to further speed up the calculation rate of the kinoform. The experiment shows that this method can correctly generate the demanded kinoform and can make the calculation speed increase 40 times or so more than the calculation method based on CPU.

Key words: computer holography; kinoform; parallel computation; three-dimension scene

责任编辑:代小红