

基于点源模型计算全息图快速生成算法的研究进展

金晓宇, 桂进斌*, 刘超, 郑立婷, 楼宇丽

昆明理工大学理学院激光研究所, 云南 昆明 650500

摘要 计算全息图(CGH)的快速生成技术是全息三维显示系统的关键技术,基于点源模型的计算全息图由于模型简单、操作灵活而成为了计算全息方法中一个重要分支。但是使用点源模型计算全息图计算量庞大,为了达到实时显示的要求,提高全息图计算速度的方法被不断提出。对点源模型快速算法的发展历程进行了综述。从计算全息图的原理出发,根据点源模型快速算法的研究方向,将其分为算法和算法与高性能硬件结合两大类。通过分析各类算法的实现方法和存在的问题,指出了其改进的方向,介绍了高性能硬件对计算全息图的贡献,最后对点源模型快速算法的未来研究方向进行了展望。

关键词 计算全息图; 点源模型; 查找表; 波前记录平面; 并行计算

中图分类号 O438.1 文献标识码 A

doi: 10.3788/LOP55.100005

Progress of Fast Generation Algorithm of Computer-Generated Hologram Based on Point Source Model

Jin Xiaoyu, Gui Jinbin*, Liu Chao, Zheng Liting, Lou Yuli

Institute of Laser Research, School of Science, Kunming University of Science and Technology, Kunming, Yunnan 650500, China

Abstract The rapid generation technology of computer-generated hologram (CGH) is the key technology of the holographic three-dimensional display system. The calculation of holograms based on point source model is an important branch of computational holography because of its simple model and flexible operation. However, the calculations of point source model for holograms are enormous. In order to meet the requirements of real-time display, the methods of improving the calculation speed of hologram have been put forward continuously. The development of the fast algorithm of point source model is reviewed. Based on the principle of CGH, according to the research direction of the fast algorithm of point source model, it is divided into two categories: algorithm, and combination of algorithm and high-performance hardware. Through the analysis of the implementation methods and existing problems of all kinds of algorithms, the direction of improvement is pointed out, and the contribution of high-performance hardware to CGH is introduced. Finally, the future research direction of point source model fast algorithm is prospected.

Key words computer-generated hologram; point source model; look-up table; wavefront record plane; parallel calculation

OCIS codes 000.3860; 090.1760; 090.5694

1 引言

全息三维显示技术因具有能够提供三维物体的全部信息,重构具有物理景深效果再现像的特性,被认定为最具有潜力的真三维显示技术,已被广泛应用于干涉计量、三维显示、动态全息计量和全息存

储等诸多领域。1948年,Gabor^[1]提出全息术的概念,可概括为“干涉记录,衍射重现”,但因为没有好的相干光源,在这之后全息术发展十分缓慢。直到20世纪60年代,由于激光器的出现^[2],全息术得到快速发展。1965年,Kozma等^[3]在制作一种复数空间滤波器的研究中,实现了与计算全息原理相似的

收稿日期: 2018-03-20; 修回日期: 2018-05-03; 录用日期: 2018-05-09

基金项目: 国家自然科学基金(61565011,61465005)、云南省自然科学基金(2014FB132)

* E-mail: jinbingui@163.com

方法。同年,美国学者 Aemitage 等^[4]发明了迂回相位方式编码,实现复数波面的相位表示,并利用计算机和绘图仪作出了第一张计算全息图(CGH)。1967年,Lohmann 等^[5]将快速傅里叶算法应用在CGH的计算中,大大缩短了计算全息图的时间。

随着计算机技术与光学技术的快速发展,计算全息的研究也越来越受到学者们的重视。计算全息是通过计算机模拟计算虚拟物体发出的物光波与参考光的干涉,最终通过波前调制设备加载干涉条纹再现三维场景^[6]的一种技术。计算全息技术利用计算机编码可以全面地记录和再现三维物体光波的振幅和相位信息,而且与传统的光学全息术相比,计算全息打破了传统全息术记录光路的限制,还具有低噪声、可重复性好以及可以获得虚拟物体全息图等多方面的优点。广义来说,计算全息属于数字全息的一个分支,它是全息技术与现代计算机、数字图像处理技术、空间光调制技术及自动化控制技术相结合而产生的一种新兴技术^[7]。

在现实中,三维物体 CGH 的计算复杂且计算量巨大,因此,如何快速生成三维物体 CGH 是数字三维全息显示中的关键问题之一^[8]。为了提高全息图计算速度,科研人员们进行了积极的研究,目前计算全息图主要分为两种方法:一是将空间曲面视为不同形状面源集合的面元法^[9-13];二是将空间曲面视为点源集合的点源法^[14-30]。由于具有操作灵活、原理简单、便于表示物体表面特性且重建质量好的优点,点源法计算全息图具有很大的应用潜力。本文从点源法原理出发,分析了基于点源法计算全息图过程中的复杂度和计算难点。重点阐述了提高 CGH 计算速度的算法和硬件,并分析和讨论了算法和硬件性能存在的某些问题,并对今后的研究方向进行了展望。

2 点源模型计算全息图的基本原理

计算机生成的全息图能够正确地记录和重建三维(3D)对象的信息。在点源模型计算全息图中,三维物体被看作理想的漫散射体。将三维物体表面离散成 N 个点的集合,其中物体表面的每个点都可以视为一个独立的点光源,如图 1 所示,它们发出的球面波可以均匀地照射在整个全息图平面上,通过计算各个点光源发出的球面波,跟踪其路径得到每一个采样点的衍射图样。这种计算方法被称为光线跟踪法。

设三维物体的空间坐标为 XOY ,全息图平面的

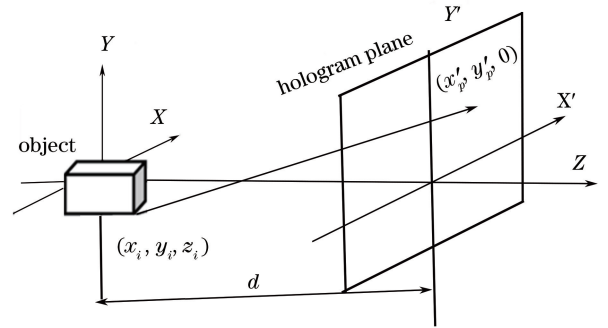


图 1 三维计算全息记录示意图

Fig. 1 Diagram of CGH for recording 3D object

空间坐标为 $X'O'Y'$,三维物体表面每个点光源坐标为 (x_i, y_i, z_i) ,可以得到光线跟踪算法计算全息图平面上的复振幅分布 $I(x'_p, y'_p)$ 的公式为

$$I(x'_p, y'_p) = \sum_{i=1}^N \frac{A_i}{r_i} \exp [j(kr_i + \varphi_i)], \quad (1)$$

式中: A_i 为第 i 点的光波振幅; φ_i 表示初始相位; $k = 2\pi/\lambda$ 表示波数,为表示三维物体是一个漫散射体,一般初始相位为 $(0, 2\pi)$ 范围内的随机相位, λ 为全息图参考光的波长; r_i 为三维物体表面的第 i 点光源 (x_i, y_i, z_i) 距全息图平面上像素点 (x'_p, y'_p) 的距离,可表示为

$$r_i = \sqrt{(x'_p - x_i)^2 + (y'_p - y_i)^2 + (d - z_i)^2}, \quad (2)$$

式中, d 为三维物体所在的空间坐标到全息图平面的垂直距离。对(1)式进行分析可以看出,在全息图计算的整个过程中,需要计算三维物体表面上的每个离散点光源在全息图上所有像素点上的复振幅分布,然后将求得的复振幅信息进行叠加获得最终的全息图。假设三维物体表面具有 N 个点光源,全息图上的抽样点为 $p \times q$,那么获得一幅全息图,(1)式的计算量为 $N \times p \times q$ 次。每次计算包含的计算方法有:复数运算一次、指数运算一次、加法和乘法运算各五次。整个计算过程非常耗时,即使使用高性能计算机,也难以在较短的时间内计算出全息图。

3 点源模型加速计算全息图生成技术

3.1 查找表法

为了解决传统点源模型计算全息图缓慢的问题,1993年,Lucente^[14]首次提出基于查找表(LUT)加速全息图计算的方法。该方法需预先计算三维物体空间中均匀采样点对全息面上每个点的贡献,并将其存放在内存或硬盘里。在计算全息图时,只需根据物点的坐标提取预先保存在 LUT 中

的基本干涉图样(EFP),因此全息图计算就是一个简单的数据读取与叠加过程,可表示为

$$I(x, y) = \sum_{p=1}^N a_p T(x, y; x_p, y_p, z_p), \quad (3)$$

式中: $T(x, y; x_p, y_p, z_p)$ 为预先计算并存储在 LUT 中的 EFP; a_p 取值为 0 和 1, 当为 1 时, 说明 LUT 中该点的 EFP 与点源匹配, 则进行叠加, 当为 0 时, 则说明不匹配。每次读取与叠加只包含一次加法和乘法运算。如果三维物体表面有 N 个离散点, 全息图的大小为 $p \times q$, 则所需的计算次数为 $N \times p \times q$ 。这种方法在离线计算时, 可得到三维物体离散点源的 EFP, 在线计算时只需进行读取和叠加操作, 即可生成三维物体的全息图。同时, 这种方法只保留简单的加法运算, 降低了在线计算的复杂程度, 大大提高了计算全息图的速度。但其缺点是离线计算所构建的 LUT 数据占据大量的存储空间, 当全息图每个像素所占空间为 M 时, 需要的存

储空间为 $N \times p \times q \times M$, 所需的存储空间达到了 GByte 量级。因此该算法难以在并行计算机和现场可编程门阵列(FPGA)芯片上实现。

LUT 快速算法需要大量的存储空间, 不利于实现全息图的实时计算。2008 年, 韩国光云大学 Kim 等^[15]在 LUT 的基础上利用空域卷积提出一种新型查找表(N-LUT)法, 这种方法可以显著降低传统 LUT 占据存储空间的大小, 并保持快速计算速度的优点。N-LUT 方法的流程框图如图 2 所示。第一步, 提取三维物体信息数据, 利用提取到的数据, 将三维物体模型划分为与深度相关的切片点阵平面。第二步, 依次选择这些点阵平面, 通过提取预先存储在 N-LUT 中该切片点阵中心点的衍射图样, 使用空域卷积和加法操作, 生成该切片点阵的衍射场。第三步, 对所有点阵平面执行上述操作, 把得到的结果与参考光进行叠加, 即可生成三维物体的 CGH。第四步, 使用生成的全息图重建三维物体图像。

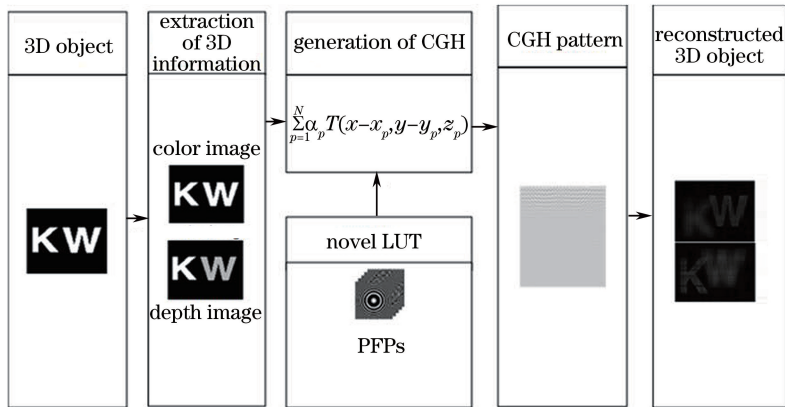


图 2 新型查找表法流程框图

Fig. 2 Flow block-diagram of the N-LUT method

对于 N-LUT 法, 其生成 CGH 的速度比射线追踪法(CRT)快 69.5 倍, 所需存储空间的大小为常规 LUT 法的 1/745, 其计算复杂度与 LUT 方法相同。虽然 N-LUT 法降低了 LUT 的存储空间但却增加了在线计算量, 而且这种方法的 LUT 存储空间仍然很大。同 LUT 法相似, N-LUT 法也难以在 FPGA 和图形处理单元(GPU)的计算芯片上实现。因此, LUT 法的研究重点是压缩波前数据以及如何快速解压缩和读取数据。

2009 年, 新加坡学者 Pan 等^[16]为了降低 LUT 存储空间使用量, 他们提出分离查找表(S-LUT)算法, 该算法在离线状态下利用两次菲涅耳近似, 计算三维物体具有相同垂直距离 z_j 的每个二维截面上点源对应的水平调制因子

$$H(\Delta x, z_j) = \exp\left(\frac{2\pi i}{\lambda} \sqrt{\Delta x^2 + z_j^2}\right) \quad (4)$$

和垂直调制因子

$$V(\Delta y, z_j) = \exp\left(\frac{2\pi i}{\lambda} \sqrt{\Delta y^2 + z_j^2}\right), \quad (5)$$

并保存在 LUT 中。式中, $\Delta x = x_h - x_j$, $\Delta y = y_h - y_j$, (x_h, y_h) 为全息面坐标, (x_j, y_j, z_j) 为物平面坐标。在线状态下, 对于具有相同 (x_j, z_j) 值的 n 个点源, 它们的水平调制因子是相同的, 因此这 n 个点源的全息图 $I_c(x_h, y_h) |_{(x_j, z_j)}$ 可分为两步得到:

$$S(y_h) = \sum_{j=0}^{n-1} a_j \times V(\Delta y, z_j), \quad (6)$$

$$I_c(x_h, y_h) |_{(x_j, z_j)} = H(\Delta x, z_j) \times S(y_h), \quad (7)$$

式中, $S(y_h)$ 为具有相同垂直距离 z_j 的 n 个点源到全息面的贡献总和。通过把三维物体不同垂直距离

z_j 的二维平面上所有点源的衍射图样相叠加,最终得到三维物体的全息图 $I_c(x_h, y_h)$ 表达式为

$$I_c(x_h, y_h) = \sum I_c(x_h, y_h) |_{(x_j, z_j)}. \quad (8)$$

这种算法利用 X 、 Y 两个方向的调制因子代替了全息图存储在表格中的数据,降低了 LUT 的存储空间和数据循环的次数,大大提高了全息图的计算速度。根据文献[13],S-LUT 法的存储空间为百 MByte 量级,在图形处理单元中实现该算法,可达到每秒计算 2×10^4 个物点的计算速度。这为计算复杂三维物体的全息图提供了一个高效的方法,但

仍然达不到实时计算的要求。

为了进一步减少 LUT 存储空间的使用和提高全息图的计算效率,2013 年,北京理工大学的 Jia 等^[17]在 S-LUT 方法的基础上提出了压缩查找表(C-LUT)法。该算法由两步组成,其算法框图如图 3 所示。第一步:在离线情况下预先计算 X - Y 方向调制因子,并存储在表格中;第二步:在线情况下,首先从表格中提取构建目标物点对应的 X 、 Y 方向的调制因子,然后再乘以对应的 Z 方向调制因子,从而获得全息图。

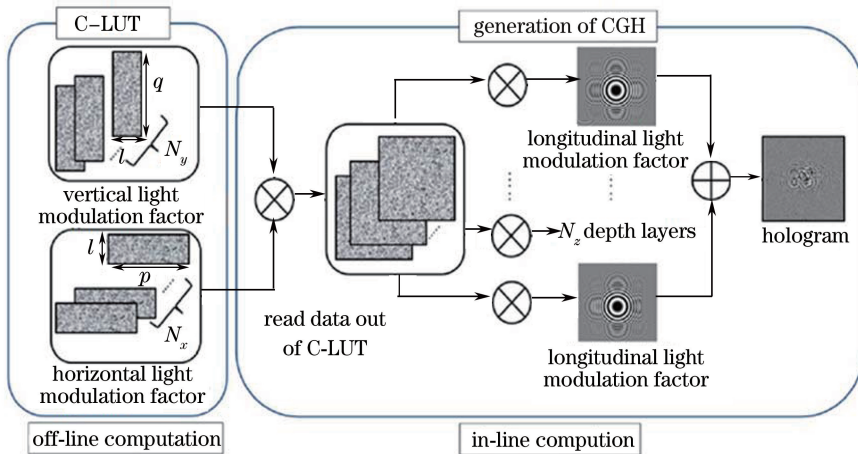


图 3 C-LUT 方法原理框图

Fig. 3 Block-diagram of the C-LUT method principle

C-LUT 法是在夫琅禾费衍射理论的基础上提出的,由于夫琅禾费全息图可以在适当的观测距离内重建与非涅耳全息图相似的强度信息。(1)式可以近似表示为

$$H_{FH}(x'_p, y'_p) = \sum_{j=0}^{N-1} A_j \exp \left[ik \frac{x_p'^2 + y_q'^2}{2(d - z_j)} \right] \times \exp \left(-ik \frac{x'_p x_j}{d} \right) \times \exp \left(-ik \frac{y'_p y_j}{d} \right), \quad (9)$$

式中, X 方向的调制因子定义为 $H(x'_p, x_j) = \exp(-ikx'_p x_j/d)$, Y 方向调制因子定义为 $V(y'_p, y_j) = \exp(-iky'_p y_j/d)$, Z 方向调制因子为 $L(z', z_j) = \exp \left[ik \frac{x_p'^2 + y_q'^2}{2(d - z_j)} \right]$, $L(z', z_j)$ 主要调制

物光在 Z 方向的聚焦位置,相当于一个透镜因子。

通过简化(9)式可得到

$$H_{FH}(x'_p, y'_p) =$$

$$\sum_{j=0}^{N-1} A_j H(x'_p, x_j) V(y'_p, y_j) L(z', z_j). \quad (10)$$

由于三维物体可以近似地看作是由不同深度的二维点阵构成,每一个二维点阵具有 $N_{x,y}$ 个点源, $N_{x,y}$ 个点源在相同深度层 N_z 平面具有相同的轴向调制因子 $L(z', z_j)$,对于每一层的二维点阵,有 N_y 个点分布在同一垂直方向,也就是说,这些点具有相同的水平方向调制因子 $H(x'_p, x_j)$ 。全息图是来自不同垂直方向 N_y 和不同深度层 N_z 上的点源全息图的叠加,最终得到三维物体全息图的表达式为

$$H_{FH}(x'_p, y'_p) = \sum_{iz=0}^{N_z-1} \left\{ \sum_{ix}^{N_x-1} \left[\sum_{iy}^{N_y-1} A_j H(x'_p, x_j) \right] \cdot V(y'_p, y_j) \right\} L(z', z_j). \quad (11)$$

C-LUT 方法由于近似,引入了水平方向和垂直方向的相位调制误差,该调制误差会导致再现像在 X 、 Y 方向上发生位置偏移,而不同物点在 X 、 Y 方

向上的位置误差大小主要由物体深度 z_j 与再现距离 d 的比值决定。文献[17]通过相位预补偿和增大 z_j/d 的比值来消除误差,相位预补偿需要在线计算

相位补偿因子,在一定程度上增加了在线计算时间。但 C-LUT 法相比于 S-LUT 法,其全息图的计算速度仍有很大的提升,同时也降低了 LUT 的存储空间。

根据文献[14-17]比较了 CRT, LUT, S-LUT

表 1 算法的复杂度,操作和内存使用比较

Table 1 Complexity, operation and memory usage comparison among the algorithms

Algorithm	Complexity of in-line	Complexity of off-line	Operation of in-line	Operation of off-line	Memory usage
CRT	$N_x N_y N_z p q$	0	$1 \exp, 1\sqrt{\cdot}, 5 \times, 5+$	0	0
LUT	$N_x N_y N_z p q$	$N_x N_y N_z p q$	$1+, 1 \times$	$1 \exp, 1\sqrt{\cdot}$	$N_x N_y N_z p q$
S-LUT	$N_z [N_x (N_y q + p q) + p q]$	$(N_x p + N_y q) N_z$	$1+, 1 \times$	$2 \exp, 2\sqrt{\cdot}, 8 \times, 2+$	$(N_x p + N_y q) M N_z$
C-LUT	$N_z [N_x (N_y q + p q) + p q]$	$N_x p + N_y q$	$1+, 1 \times 1 \exp, 4 \times, 1+$	$2 \exp, 8 \times$	$(N_x p + N_y q) M$

从表 1 中可以看出, C-LUT 方法所需存储空间最少, 离线下的运算量也最小, 它的计算复杂度和存储空间与物体的层数 N_z 无关, 只与单层的点数有关。如果将 C-LUT 算法与 GPU 和 FPGA 结合, 构建全息快速计算平台, 可进一步提高生成全息图的速度, 在将来应用于实时三维全息显示系统中。

虽然 C-LUT 法计算全息图的速度并不慢, 但是需要对算法中近似引入的相位调制误差进行校正, 增加了算法的复杂性。2014 年, 蒋晓瑜等^[18]提出三角函数查表法(T-LUT)。该算法基于点源模型的基本原理, 通过相关的数学近似和三角恒等变换, 生成一种纯相位 LUT。在 LUT 离线制作中, 将深度信息加入到离线计算中, 降低了在线计算的复杂度, LUT 的存储内存在理论上与 S-LUT 相当。相比于 C-LUT, T-LUT 中存储了深度信息, 因此在线计算三维物体全息图的计算速度上具有优势。文献[15]将该算法在 GPU 上加以实现, 通过利用 NVIDIA 开发的统一计算架构(CUDA)进行并行编程计算, 并进行了三次并行优化。相对于 CRT 法的 GPU 加速运算, T-LUT 速度可提升 30 倍至近千倍。

随着科研人员的不断努力, 利用 LUT 快速计算全息图已经取得了很大的进步。同时, LUT 的计算量也得到大大减少, 由原来的 GByte 量级到现在的 Byte 量级, 有效地解决了点源模型计算全息图计算量大、速度慢的问题。但这种算法仍达不到实时计算的要求, 因此需要对 LUT 方法进行改进, 将 LUT 法与高性能硬件和其他算法有机结合, 进一步提高全息图的计算速度, 在将来达到实时计算的要求。

3.2 波前记录平面法

2009 年, 日本千叶大学 Shimobaba 等^[19]提出了一种基于傅里叶变换的点源模型快速算法——波前记录平面(WRP)法。这种方法是在全息面与物

和 C-LUT 这几种算法的复杂度、运算符和存储空间。假设全息图一个像素所占内存为 M 、三维物体深度为 N_z 、每一层水平方向和垂直方向有 N_x 和 N_y 个取样点, 结果如表 1 所示。

面之间加入一个虚拟平面。计算全息图时, 首先利用光线跟踪法计算每个物点在波前记录面上的复振幅分布, 然后通过菲涅耳或角谱衍射将其传播到全息平面。由于 WRP 离三维物体距离很近, 在衍射极限限制的影响下, 物光波只能穿过波前记录平面上的小区域(半径为 W_1), 使得点源的物光波对 WRP 的贡献面积非常小, 从而全息图的计算复杂度被大大降低, 可以看出 WRP 半径 W_1 的大小决定了全息图的计算时间。

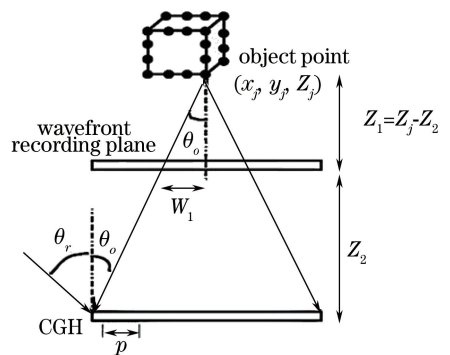


图 4 波前记录平面的计算概要

Fig. 4 Outline of the calculation with the WRP

如图 4 所示, 物光的最大衍射角和参考光的入射角关系可表示为

$$\sin \theta_o - \sin \theta_r = \lambda / 2p, \quad (12)$$

式中, θ_r 为参考光的入射角, θ_o 为三维物体的最大衍射角。为了便于讨论, 假设参考光入射角 $\theta_r = 0$, WRP 上的半径 W_1 可表示为

$$W_1 = |Z_1| \tan [\arcsin(\lambda / 2p)] \approx |Z_j - Z_2| \tan(\lambda / 2p), \quad (13)$$

式中, Z_1 为物点到 WRP 的垂直距离, Z_j 为物点到全息图的垂直距离, Z_2 为 WRP 到全息图的距离, λ 为参考光波长, p 为全息图像素尺寸。当 θ_o 较小时, 傍轴近似可以获得较好的计算结果。根据(13)式可知, WRP 的半径 W_1 主要由 Z_j 和 Z_2 决定。三

维对象点源的半径平均值为 $\bar{W}_1 = (1/N) \sum_j^N W_{1j}$ 。

WRP 上的横截面积的平均值表示为 $\bar{W}_1^2 \pi$ 。当 WRP 靠近三维物体时, \bar{W}_1 减小, 可加快全息图的计算速度。然而当计算具有较大景深的三维物体时, \bar{W}_1 增大, 该方法的计算量也会增加。为了解决这个问题, 需要寻找最合适的 Z_2 , 使 \bar{W}_1 最小化。但目前尚没有寻找最优的 Z_2 的很好的办法, 这是 WRP 法需要解决的一个研究问题。该方法还存在另一个主要问题就是不能记录大于全息图尺寸的三维物体。

WRP 法第一步需要利用光线跟踪法计算每个点源在 WRP 的贡献, 该过程需要大量的计算。为了减少这个过程的计算量, 2010 年, Shimobaba 等^[20]对波前平面法进行了改进, 提出 LUT 法和 WRP 法相结合快速计算全息图的方法。该算法分为两步: 第一步利用 LUT 方法来计算三维物体到 WRP 面的衍射计算, 第二步使用 GPU 计算 WRP 面到 CGH 的衍射计算。与传统 WRP 法相比, 该算法计算复杂度大大降低, 对于 3×10^4 个物点生成尺寸为 $2048 \text{ pixel} \times 2048 \text{ pixel}$ 的相息图可以实现 10 frame/s 的重建。

为了克服传统 WRP 法不能记录大于全息图尺寸的三维物体和提高全息图的计算速度的难点, 2014 年, Phan 等^[21]在 WRP 法的基础上提出了双 WRP 的快速算法, 其中每个 WRP 记录三维物体的一部分波前。与单个 WRP 法相比, 双 WRP 法在计算具有较大景深的三维物体上更有优势, 计算全息图的速度和重建质量都有很好的提升。而且双 WRP 法的两个 WRP 平均生成速度是单个 WRP 法生成一个 WRP 的 2.5 倍。

在 WRP 方法中可以通过将 WRP 靠近三维物体放置来减少计算量。而在文献^[19-21]中, WRP 的数量和间隔是固定的, 且未对 WRP 的排列位置和数量进行优化处理, 因此基于 WRP 法计算全息图的时间仍有很大的提升空间。为了进一步提高全息图的计算速度, 科研人员进行了大量的研究。2017 年, Hasegawa 等^[22]提出一种可以自动优化 WRP 数量和排列位置加速生成 CGH 的方法。该算法根据相关参数以及 WRP 面积阈值的设定, 经过多次迭代寻找最优化的 WRP 数量和排列位置, 大大减少了生成全息图的时间, 而且重建图像的质量与传统方法几乎相同。在该方法中, 迭代运算较

为耗时, 应该寻求一种解析方法, 可根据三维物体和全息图的大小计算出最优的 WRP 位置及数量。同年, Arai 等^[23]提出一种基于小波变换的快速算法。该研究小组将小波变换应用于 WRP 算法中, 使用小波变换完成物点到 WRP 的衍射计算。由于 WRP 距物平面较近, 在第一步中应用小波变换, 可以弥补小波变换随衍射距离增加而导致计算时间增加的缺点; 而且这种方法对常规 WRP 法第二步也进行了改进, 使用两步频限菲涅耳衍射计算 WRP 到全息图的衍射, 两步频限菲涅耳衍射比常规菲涅耳衍射和角谱衍射计算速度更快。由于该算法互补了小波变换和 WRP 法的缺点, 其计算全息图的速度得到了很大的提升。对于大小为 $2048 \text{ pixel} \times 2048 \text{ pixel}$ 像素的全息图, 0.4 s 内可计算一百万个物点。

在上述 WRP 法中, 全息图的计算速度较 LUT 法已经有了很大的提升。但都存在一个问题, 缺乏对三维物体表面点源之间的遮挡的考虑。如果将剔除遮挡技术应用于上述 WRP 法中, 计算速度将会得到进一步提升。

3.3 高性能硬件计算

利用改进算法加快全息图的计算速度, 存在着局限性, 高性能硬件与算法相结合的方法是当下提高全息图计算速度的有效手段, 也将是计算全息发展的主要趋势。目前高性能硬件加速全息图生成的方法根据硬件的不同可分为: FPGA 和 GPU。下面将分别对它们进行介绍和分析。

早在 1992 年, 日本学者 Ito 等^[24-28]就使用可编程逻辑器件构建了专门用来加速全息图计算的硬件系统 (HORN)。1993 年, 该研究小组完成了 HORN-1 的制作和设计, HORN-1 主要由 26 块集成芯片构建, 时钟频率为 10 MHz, 采用 16 位 ROM, 其管线结构为并行模式。HORN-1 设计简单, 计算能力有限, 只能计算分辨率为 $400 \text{ pixel} \times 400 \text{ pixel}$ 的全息图, 但是其计算全息图的速度得到了很大的提升, 与当时主流的计算机相比速度提升了 100 倍。随着 FPGA 的迅速发展以及千叶大学研究小组的不断努力, 现今已经研制了 7 个版本的 HORN, 为全息三维显示的应用做出了很大的贡献。其最新系列 HORN-7 可以在 0.4 s 内生成 1.6×10^4 个物点的 2M 像素全息图, 将来可应用于实时全息三维显示。然而, 全息专用计算机的开发周期长且价格昂贵, 而且需要很高的硬件设计能力, 相比之下, GPU 具有高性能、价格便宜及开发周期

短等优点,引起了研究人员的密切关注。学者们纷纷利用 GPU 对全息图计算进行加速。

1995年,MIT实验室^[29]开发了一种计算和显示全息图像的系统,首先将图形工作站用于全息图的计算,并用硬件结构和软件对系统进行实现。如图5所示,该系统通过图像处理系统中的专用流处

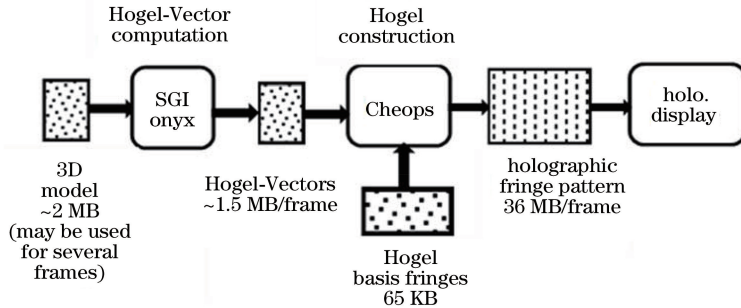


图5 Hogel-Vector 生成和解码

Fig. 5 Hogel-Vector generation and decoding

1999年,Ritter等^[30]提出了基于通用图形语言 OpenGL 对计算机图形硬件进行操作生成全息图的方法。为了在计算机图形硬件中实现二维复值场的叠加,该方法将预先计算出的各个物点的菲涅耳波带的复数值编码到标准图像彩色像素的复杂纹理中,如果实部为正值,则使用红色通道表示,如果为负值,则使用绿色通道表示并存储为绝对值,虚部利用蓝色和阿尔法通道来存储。其中每个纹理都代表单个图形基元的全息图,生成全息图就是应用特殊图形单元累加器对复杂纹理进行叠加。与传统方法相比,这种方法可以将计算全息图的速度提高60~90倍。

2006年,Ahrenberg等^[31]采用 OpenGL 图形库和 OpenGL 着色器语言(GLSL)对 GPU 编程,加速计算全息图,可在0.1s完成1000个物点,生成分辨率为960 pixel×600 pixel的全息图。

2009年,剑桥大学Chen等^[32]提出了一种在图像处理单元上进行并行计算全息图的算法,该算法应用二维高斯插值来创建离散物点构成连续表面,通过预先计算基本物点分布的LUT来代替耗时的计算,由于基本物点分布关于原点对称,LUT只需存储完整物点分布的四分之一。他们还分别利用遮挡剔除、非均匀采样方法减少物点的数量,进一步加速计算全息图的速度。利用这种方法计算全息图在GPU上比在CPU上快14到15倍。虽然该方法提高了全息图的生成速度,但因为在剔除遮挡时进行了两次近似,故降低了重建像的质量,且未对重建像进行评估,也未实现遮挡剔除与非均匀采样相结合的方法。

2010年,Shimobaba等^[33]使用AMD-HD5000

理硬件计算全息图,采用两步计算全息的方法。他们首先利用流处理器对三维场景进行Hogel-Vector编码,然后再进行快速解码得到三维全息图像,并对三维全息图进行实时显示。这种方法提高了传统计算全息图的速度,为全息图的计算提供了一个新的方法。

系列GPU,在OpenCL架构下,快速计算全息图。并与当前主流GPU生产商NVIDIA在CUDA架构下的计算能力进行了比较。实验结果表明,使用AMD生产的GPU在OpenCL架构下的计算全息图的能力更强。这在GPU下开发快速计算全息图算法的学者提供了一个参考。

2012年,Takada等^[34]利用多图像处理单元集群系统实现CGH的计算,多GPU系统能够在0.055s内计算由2048个点组成的三维物体6400 pixel×3720 pixel像素的全息图,且比常规八线程CPU快533倍。

为了在图像处理单元中利用点源模型计算大体积、复杂三维物体的全息图,2016年,Sugawara等^[35]提出了一种基于GPU的三角形补丁模型快速算法,模型如图6所示。

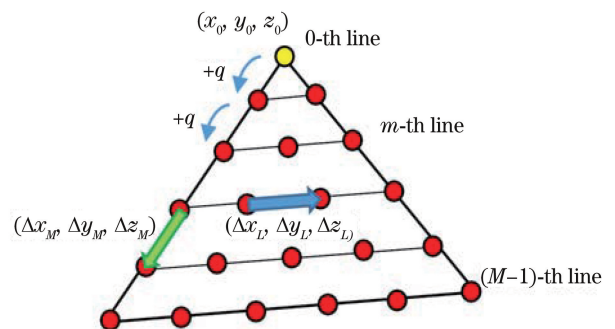


图6 三角形补丁模型的坐标

Fig. 6 Coordinates of the triangular patch model

该方法通过公式推导得到物光在全息图上的分布表达式为

$$U(\xi, \eta) = C \exp(-jkr_0) \left[\frac{1 - \exp(-jkl\Phi_x)}{1 - \exp(-jk\Phi_x)} \right] \left[\frac{1 - \exp(-jkM\Phi_y)}{1 - \exp(-jk\Phi_y)} \right], \quad (14)$$

其中,

$$\begin{cases} r_0 = z_0 + \frac{(x_0 - \xi)^2 + (y_0 - \eta)^2}{2z_0} \\ \Phi_x = \Delta z_L - \frac{\Delta z_L [(x_0 - \xi)^2 + (y_0 - \eta)^2]}{2z_0^2} - \frac{\Delta x_L [(x_0 - \xi)^2 + (y_0 - \eta)^2]}{z_0} \\ \Phi_y = \Delta z_L - \frac{\Delta z_M [(x_0 - \xi)^2 + (y_0 - \eta)^2]}{2z_0^2} - \frac{\Delta x_M [(x_0 - \xi)^2 + (y_0 - \eta)^2]}{z_0} \end{cases}, \quad (15)$$

式中, $(\Delta x_L, \Delta y_L, \Delta z_L)$ 为水平间隔, $(\Delta x_M, \Delta y_M, \Delta z_M)$ 为垂直间隔, (x_0, y_0, z_0) 为三角形补丁模型的初始点。由于(1)式没有点或者线的循环操作, 复杂度为 $O(PN^2)$, P 为三角形补丁的个数, 所以 P 的个数远小于物体点源个数。当点数较多时, 提出的方法比传统的基于点的方法要快得多。该方法结合了点源模型和快速傅里叶方法, 便于在 GPU 中实现。实验证明, 这种方法的计算速度与三维物体点源个数无关, 只与三维物体三角形补丁的个数有关。因此随着这种算法的发展, 该算法可成为复杂三维物体实时全息计算的基础。

随着高性能硬件的快速发展, 利用硬件和算法结合的方法可以大幅度提高全息图计算速度, 实现 CGH 的实时再现。但是上述文献中仍存在一些问題, 如未对三维物体的色彩问题进行讨论, 而且计算模型也过于单一。在以后的研究中可以考虑使用复杂的基元(如三角形)代替点源来描述三维对象, 同时利用贴图技术或者彩色全息图合成技术, 实现彩色复杂物体的实时全息三维显示。

4 结 论

对基于点源模型快速计算全息图的算法和高性能硬件进行了总结。虽然在算法上取得很大的进步, 但三维全息显示技术尚未在本质上得到突破, 其主要原因是计算全息图的速度仍达不到实时显示的要求。从目前的研究进展看, 提高全息图的计算速度可分为两个主要的研究方向, 一是开发快速算法, 二是使用高性能计算机。由于全息图计算具有高度并行性和 GPU 具有开发周期短、价格便宜等优点, 利用 GPU 计算全息图逐渐成为主要的计算模式。为了将来的实际应用, 需要改进算法和提高硬件设备计算性能, 并将多种算法和硬件结合到单个计算系统中。根据不同的三维物体, 选择最优化的计算方法, 从而实现三维物体的实时全息三维显示。

参 考 文 献

- [1] Gabor D. A new microscopic principle[J]. Nature, 1948, 161(4098): 777-778.
- [2] Zhu Y F. The direction of commercial laser holography[J]. Printing Technology, 2000(12): 76-77.
朱云峰. 商用激光全息发展方向[J]. 印刷技术, 2000(12): 76-77.
- [3] Kozma A, Kelly D L. Spatial filtering for detection of signals submerged in noise[J]. Applied Optics, 1965, 4(4): 387-392.
- [4] Armitage J D, Lohmann A W. Character recognition by incoherent spatial filtering[J]. Applied Optics, 1965, 4(4): 461-467.
- [5] Lohmann A W, Paris D P. Binary Fraunhofer holograms, generated by computer[J]. Applied Optics, 1967, 6(10): 1739-1748.
- [6] Goodman J W. Introduction to Fourier optics[M]. Qin K C, Liu P S, Chen J B, *et al.*, Transl. 3rd ed. Beijing: Electronic Industry Press, 2011.
Goodman J W. 傅里叶光学导论[M]. 秦克诚, 刘培森, 陈家璧, 等, 译. 3 版. 北京: 电子工业出版社, 2011.
- [7] Li J C. Diffraction calculation and digital holography[M]. Beijing: Science Press, 2014.
李俊昌. 衍射计算及数字全息[M]. 北京: 科学出版社, 2014.
- [8] Jia J, Wang Y T, Liu J, *et al.* Progress of dynamic 3D display of the computer-generated hologram[J]. Laser & Optoelectronics Progress, 2012, 49(5): 050002.
贾甲, 王涌天, 刘娟, 等. 计算全息三维实时显示的研究进展[J]. 激光与光电子学进展, 2012, 49(5): 050002.
- [9] Matsushima K, Nakahara S. Region segmentation and parallel processing for creating large-scale CGHs in polygon source method[J]. Proceedings of SPIE,

- 2009, 7233: 72330E.
- [10] Nishi H, Higashi K, Arima Y, *et al.* Smooth shading of specular surfaces in polygon-based high-definition CGH [C] // 2011 3DTV Conference: The True Vision-Capture, Transmission and Display of 3D Video, May 16-18, 2011, Antalya, Turkey. New York: IEEE, 2011: 12070126.
- [11] Zhang Y P, Zhang J Q, Chen W, *et al.* Fast computer generated hologram algorithm of triangle mesh models [J]. Chinese Journal of Lasers, 2013, 40(7): 0709001.
张亚萍, 张建强, 陈伟, 等. 基于三角模型的计算全息快速算法 [J]. 中国激光, 2013, 40(7): 0709001.
- [12] Liu C, Gui J B, Li J C, *et al.* Fast generation algorithm of computer-generated hologram based on triangular surface light source frequency spectrum analytic solutions [J]. Laser & Optoelectronics Progress, 2018, 55(1): 010901.
刘超, 桂进斌, 李俊昌, 等. 基于三角形面光源频谱解析的计算全息图快速生成算法 [J]. 激光与光电子学进展, 2018, 55(1): 010901.
- [13] Matsushima K, Nakahara S. New techniques for wave-field rendering of polygon-based high-definition CGHs [J]. Proceedings of SPIE, 2011, 7957: 79571A.
- [14] Lucente M E. Interactive computation of holograms using a look-up table [J]. Journal of Electronic Imaging, 1993, 2(1): 28-34.
- [15] Kim S C, Kim E S. Effective generation of digital holograms of three-dimensional objects using a novel look-up table method [J]. Applied Optics, 2008, 47(19): D55-D62.
- [16] Pan Y C, Xu X W, Solanki S, *et al.* Fast CGH computation using S-LUT on GPU [J]. Optics Express, 2009, 17(21): 18543-18555.
- [17] Jia J, Wang Y T, Liu J, *et al.* Reducing the memory usage for effective computer-generated hologram calculation using compressed look-up table in full-color holographic display [J]. Applied Optics, 2013, 52(7): 1404-1412.
- [18] Jiang X Y, Cong B, Pei C, *et al.* A new look-up table method of holographic algorithms based on compute unified device architecture parallel computing [J]. Acta Optica Sinica, 2015, 35(2): 0209001.
蒋晓瑜, 丛彬, 裴闯, 等. 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 [J]. 光学学报, 2015, 35(2): 0209001.
- [19] Shimobaba T, Masuda N, Ito T. Simple and fast calculation algorithm for computer-generated hologram with wavefront recording plane [J]. Optics Letters, 2009, 34(20): 3133-3135.
- [20] Shimobaba T, Nakayama H, Masuda N, *et al.* Rapid calculation algorithm of Fresnel computer-generated-hologram using look-up table and wavefront-recording plane methods for three-dimensional display [J]. Optics Express, 2010, 18(19): 19504-19509.
- [21] Phan A H, Piao M L, Gil S K, *et al.* Generation speed and reconstructed image quality enhancement of a long-depth object using double wavefront recording planes and a GPU [J]. Applied Optics, 2014, 53(22): 4817-4824.
- [22] Hasegawa N, Shimobaba T, Kakue T, *et al.* Acceleration of hologram generation by optimizing the arrangement of wavefront recording planes [J]. Applied Optics, 2017, 56(1): A97-A103.
- [23] Arai D, Shimobaba T, Nishitsuji T, *et al.* An accelerated hologram calculation using the wavefront recording plane method and wavelet transform [J]. Optics Communications, 2017, 393: 107-112.
- [24] Shimobaba T, Masuda N, Sugie T, *et al.* Special-purpose computer for holography HORN-3 with PLD technology [J]. Computer Physics Communications, 2000, 130(1/2): 75-82.
- [25] Shimobaba T, Hishinuma S, Ito T. Special-purpose computer for holography HORN-4 with recurrence algorithm [J]. Computer Physics Communications, 2002, 148(2): 160-170.
- [26] Ito T, Masuda N, Yoshimura K, *et al.* Special-purpose computer HORN-5 for a real-time electroholography [J]. Optics Express, 2005, 13(6): 1923-1932.
- [27] Ichihashi Y, Nakayama H, Ito T, *et al.* HORN-6 special-purpose clustered computing system for electroholography [J]. Optics Express, 2009, 17(16): 13895-13903.
- [28] Masuda N, Hirai D, Okada N, *et al.* Special purpose computer for phase modulation type electroholography [C] // Forum on Information Technology 2012, September 4-6, 2012, Hosei University, Tokyo, Japan. Tokyo: IECIE, 11(1): 285-286.
- [29] Watling J A, Lucente M E, Sparrell C J, *et al.* Hardware architecture for rapid generation of electroholographic fringe patterns [J]. Proceedings of SPIE, 1995, 2406: 172-183.

- [30] Ritter A, Böttger J, Deussen O, *et al.* Hardware-based rendering of full-parallax synthetic holograms [J]. *Applied Optics*, 1999, 38(8): 1364-1369.
- [31] Ahrenberg L, Benzie P, Magnor M, *et al.* Computer generated holography using parallel commodity graphics hardware [J]. *Optics Express*, 2006, 14(17): 7636-7641.
- [32] Chen R H Y, Wilkinson T D. Computer generated hologram from point cloud using graphics processor [J]. *Applied Optics*, 2009, 48(36): 6841-6850.
- [33] Shimobaba T, Ito T, Masuda N, *et al.* Fast calculation of computer-generated-hologram on AMD HD5000 series GPU and OpenCL [J]. *Optics Express*, 2010, 18(10): 9955-9960.
- [34] Takada N, Shimobaba T, Nakayama H, *et al.* Fast high-resolution computer-generated hologram computation using multiple graphics processing unit cluster system [J]. *Applied Optics*, 2012, 51(30): 7303-7307.
- [35] Sugawara T, Ogihara Y, Sakamoto Y. Fast point-based method of a computer-generated hologram for a triangle-patch model by using a graphics processing unit [J]. *Applied Optics*, 2016, 55(3): A160-A166.