

学校代码 10345

研究类型 应用基础研究



浙江师范大学  
ZHEJIANG NORMAL UNIVERSITY

# 硕士学位论文

题 目: 层析法计算机制全息图快速算法研究

学 科 专 业: 光学工程

年 级: 2019 级 学 号: 201920200707

研 究 生: 姚建云 指导教师: 李勇

中图分类号: 0438.1 论文提交时间: 2022 年 3 月 27 日

层析法计算机制全息图快速算法研究

Research on Fast Algorithm of Layer-based Computer

Generated Hologram

作者：姚建云

Author: Yao Jianyun

申请学位类别：工学硕士

Degree Category: Master of Engineering

学科专业：光学工程

Subject: Optical Engineering

学位授予单位：浙江师范大学

Degree Conferring Unit: Zhejiang Normal University

指导教师：李勇

Supervisor: Li Yong

论文提交时间：2022 年 3 月

Thesis Submission Time: April 2022

# 浙江师范大学硕士学位论文答辩委员会决议书

(本表一式1份,存学校档案,编入学位论文)

姓名	姚建云	学号	201920200707
学院	物电学院	导师姓名	李勇
一级学科(专业学位类别)名称	光学工程	二级学科(专业学位领域)名称	
论文题目	层析法计算机制全息图快速算法研究		
<p>答辩委员会对论文的学术评语(论文选题意义;所用资料和数据可靠性;论文创新性成果及学术水平;论文写作规范化和逻辑性;论文存在的主要不足之处;答辩情况等)</p> <p>全息术能够完整记录物光场的振幅和相位信息,再现像可以提供三维视觉所需的所有深度线索,是一种理想的三维显示技术。计算全息三维显示是目前研究的热点。论文选题紧跟研究前沿,研究内容学术价值高,应用前景较强,符合光学工程学位点的选题要求。</p> <p>论文以提高计算机制全息图的计算速度,和提高全息三维显示质量为目标。巧妙利用人眼的低通滤波特性,提出了采用积分项周期延拓的全息图快速算法,大大提高了全息图的计算速度,创新性强。设计了基于多线程的多核CPU全息图并行计算算法,进一步提高了全息图计算速度。设计了基于OpenGL的三维场景渲染与多视角消隐算法,提高了三维显示质量。论文研究工作饱满,研究方案、技术路线科学合理,取得了良好的研究成果。</p> <p>论文写作规范,结构合理,层次分明,数据详实可信。答辩过程中表达清楚,回答问题正确,表明该同学具有扎实的理论基础和良好的科学实验素质。</p> <p>答辩委员会认为该论文达到了硕士学位论文的要求,经无记名投票,一致同意姚建云同学通过硕士学位论文答辩,并建议授予硕士学位。</p>			
<p>答辩平均成绩: <u>93.8</u>                      等级: <input checked="" type="checkbox"/>优秀   <input type="checkbox"/>良好   <input type="checkbox"/>合格   <input type="checkbox"/>不合格</p> <p>答辩表决结果:     <input checked="" type="checkbox"/>通过                      <input type="checkbox"/>不通过</p> <p>授予学位表决结果: <input checked="" type="checkbox"/>同意授予                      <input type="checkbox"/>不同意授予</p> <p>委员(签名):     钱文龙   王世明   任玉君   金伟民</p> <p>答辩委员会主席(签名):     何飞</p> <p style="text-align: right;">2022年5月19日</p>			

注:答辩表决结果以答辩评分表的“总体评价”结果为准,全体委员2/3以上同意视为通过答辩。

---

# 层析法计算机制全息图快速算法研究

## 摘 要

全息术的发明为三维显示技术打开了一扇新的大门。它能够完整记录物光场的振幅和相位信息,再现像可以提供三维视觉所需的所有深度线索。与基于双目视差的三维显示、多视点三维显示、体三维显示等技术相比,全息三维显示有着观感舒适、视差连续、分辨率高等优势。相干性好的激光光源问世以及计算机技术的突飞猛进,使得全息三维显示技术走上了实用化的道路。

传统的光学全息记录过程对实验环境和设备要求较为苛刻,记录介质不能够重复利用且只能记录真实存在的物体。计算全息是通过数值计算得到三维物体的全息图,将计算好的全息图加载到空间光调制器或输出到记录介质上实现对三维物体的再现。此外计算全息还可以实现虚实场景的结合及动态显示,因此得到了广泛而深入的研究。其存在的主要问题之一是计算量大导致了计算耗时。本论文研究如何提高计算机制全息图的计算速度。主要研究内容如下:

首先通过对菲涅尔全息图的计算模型进行分析,发现离散物点菲涅尔衍射公式中积分项具有周期性分布的特点。巧妙利用人眼的低通滤波特性,提出了采用积分项周期延拓的全息图快速算法。在计算全息图时根据空域(物面)和空频域(全息面)抽样间隔、衍射距离、物体大小等参数确定出积分项一个周期的大小并计算其值;再将积分项周期延拓至全息图大小,得到整个全息面上的积分项。这一过程大大减少了积分项的计算量。根据积分项外系数可正交分解特点,对积分项外的系数进行行列分解,仅计算 1 行 1 列系数分量,通过行列系数组合的方法得到每个样点系数的值。采用行列分解后大幅减少了耗时的三角函数运算,进一步提高了全息图的计算速度。

其次讨论了层析法全息图计算中的隐藏面消除问题。提出了大尺寸计算全息图制作中物体多视角消隐方法,提高了三维显示质量。设计了基于 OpenGL 的三维场景渲染与多视角消隐算法。通过 C++编程实现了多种三维模型的加载和场景渲染,并获取不同视角下消隐后的三维数据及纹理。

最后为进一步提高全息图的计算速度,设计了基于多线程的多核 CPU 全息图并

---

行计算算法。采用 C++ 实现了算法。在 Intel 的 CPU 上以 7 核并行计算，将全息图的计算速度提高了 5 倍以上。

**关键词：**三维显示；计算全息；菲涅尔衍射；快速傅里叶变换；并行计算

---

# RESEARCH ON FAST ALGORITHM OF LAYER-BASED COMPUTER GENERATED HOLOGRAM

## ABSTRACT

The invention of holography has opened a new door for 3d display technology. The amplitude and phase information of the object light field can be completely recorded by holography, and all the depth clues of 3D vision are provided by the reconstructed image of hologram. Compared with binocular parallax based 3D display, multi-view 3D display, volume 3D display, holographic 3D display has the advantages of comfortable view, continuous parallax and high resolution. With the advance of laser which have a good coherence and the rapid development of computer technology, holographic 3D display technology has become practical.

The traditional optical holographic recording process requires strict experimental environment and equipment. The recording medium can not be reused and can only record real objects. Computer generated holography (CGH) is to obtain the hologram of 3D object through numerical calculation. The calculated hologram is loaded into the spatial light modulator or output to the recording medium to realize the reconstruction of 3D object. In addition, CGH can realize the combination of virtual and real scenes and dynamic display, so it has been widely and deeply studied. One of the main problems is the large amount of computation leads to time consuming. In this thesis, it is studied how to improve the computational speed of CGH. The main research contents are as follows.

Firstly, by analyzing the calculation model of Fresnel hologram, it is found that the integral term in the Fresnel diffraction formula of discrete object points has the characteristics of periodic distribution. Based on the low-pass filtering characteristics of human eyes, a fast algorithm of CGH using periodic extension of integral term is proposed. In the calculation of hologram, the size of one period of integral term is determined and its value is calculated according to the parameters such as sampling interval, diffraction

---

distance and object size in spatial (object space) and spatial frequency domain (hologram plane). The integral term on the whole hologram plane is obtained by periodically extending the calculated integral term. According to the orthogonal decomposition characteristics of the coefficients of the integral term, the coefficients are decomposed into rows and columns. Only the coefficient components of one row and one column are calculated. The value of the coefficient of each sample point is obtained by the combination of row and column coefficients. After using row column decomposition, the time-consuming trigonometric function operation is greatly reduced, and the calculation speed of hologram is further improved.

Secondly, the problem of hidden surface removal in layer-based hologram calculation is discussed. A multi-view hidden surface removal method is proposed for computing large-scale CGH, which improves the quality of 3D display. A 3D scene rendering and multi-view hidden surface removal algorithm based on OpenGL is designed. The loading of various 3D models and scene rendering are realized through C++ programming, and the 3D data and texture after hidden surface removal from different viewing angle are obtained.

Finally, in order to further improve the calculation speed of hologram, a parallel computing algorithm of hologram with multi-core CPU based on multithreading is designed. The algorithm is implemented in C++. With 7-core parallel computing on Intel CPU, the computing speed of hologram is increased by more than 5 times.

**KEY WORDS:** Three-dimensional display; Computer generated hologram; Fresnel diffraction; Fast Fourier transform; Parallel computing

# 目 录

摘 要 .....	II
ABSTRACT .....	IV
目 录 .....	VI
1 绪论 .....	1
1.1 论文选题背景及意义 .....	1
1.2 计算全息快速算法研究现状 .....	3
1.3 本论文研究内容 .....	4
2 计算全息原理 .....	6
2.1 全息术理论基础 .....	6
2.1.1 波前记录 .....	6
2.1.2 波前重建 .....	7
2.2 计算全息理论基础 .....	8
2.2.1 抽样定理 .....	9
2.2.2 计算全息信息容量 .....	10
2.3 全息图的层析计算方法 .....	11
2.3.1 层析角谱法 .....	11
2.3.2 层析菲涅尔衍射法 .....	11
2.4 本章小结 .....	12
3 周期延拓计算全息快速算法 .....	13
3.1 算法原理 .....	13
3.1.1 离散菲涅尔衍射公式分析 .....	13
3.1.2 积分项系数特点 .....	17
3.2 三维模型处理 .....	17
3.3 算法流程 .....	20
3.4 本章小结 .....	20
4 全息图消隐及多核并行算法 .....	21
4.1 大视角全息图消隐 .....	21
4.1.1 全息图多视角消隐算法 .....	21
4.1.2 算法流程 .....	23
4.1.3 消隐算法实现 .....	23
4.1.4 模型加载及数据输出 .....	25
4.2 全息图多线程并行计算 .....	27



---

4.2.1 并行算法 .....	27
4.2.2 算法实现 .....	29
4.3 本章小结 .....	31
5 实验及结果分析 .....	32
5.1 实验系统 .....	32
5.2 快速算法光学再现结果 .....	33
5.3 快速算法速度分析 .....	35
5.4 消隐与并行加速结果分析 .....	37
5.5 本章小结 .....	39
6 总结与展望 .....	40
6.1 总结 .....	40
6.2 展望 .....	40
参考文献 .....	42
攻读学位期间取得的成果 .....	46
致  谢 .....	47
浙江师范大学学位论文诚信承诺书 .....	48
浙江师范大学学位论文独创性声明 .....	49
学位论文使用授权声明 .....	50

# 1 绪论

## 1.1 论文选题背景及意义

以人类现有的知识背景看，我们生活在三维世界中。有多种描述三维世界的方法，常见的如绘画、建筑图纸、照相术等都是把三维的世界记录成二维的形式。随着科技的不断发展，人类对三维信息的记录和再现提出了更高的要求。为此科学家在三维信息的测量和重建方面付出了诸多心血。在光学三维测量领域发展出了众多测量方案，如飞行时间法、莫尔轮廓术、相位测量轮廓术、傅里叶变换轮廓术、调制测量轮廓术等。与此同时三维显示技术也得到了迅速的发展。三维显示可以真实地再现三维场景，并能添加虚拟物体到场景中。这一特点使其在远程医疗、文化传媒、游戏开发、军事应用等领域备受关注。

生理上，人对三维信息的感知主要来源于调焦 (Accommodation)、汇聚 (Convergence)、运动视差 (Motion parallax) 和双目视差 (Binocular disparity) 等。心理上，大脑可从二维图像中提取心理深度线索，从而获得三维感觉<sup>[1]</sup>。实际中三维显示的目标就是尽可能多地满足生理、心理上三维视觉感知需求。现有的主流三维显示技术主要有基于双目视差的三维显示、多视点三维显示 (Multiview 3D Display)、体三维显示和全息三维显示等等。

基于双目视差的三维显示技术核心思想是先将三维场景记录成两个视角下的二维图像，再将二者分别传递到观察者的左右眼。左右视图的显示方法主要有颜色编码<sup>[2]</sup>、偏振态编码<sup>[3]</sup>、时分复用<sup>[4]</sup>和头戴显示<sup>[5]</sup>等。顾名思义，颜色编码是用不同颜色的光编码左右视图，观看者佩戴相应的滤色片眼镜即可实现左右眼分别获取不同视图。偏振态编码则是将左右视图分别用正交的偏振光编码，观看者佩戴与之偏振态一致的眼镜即可实现左右眼分别获取不同视图。时分复用是利用人眼的视觉暂留效应，在左右镜片之间高速切换控制通光与否，显示器同步加载相应的视图实现左

右眼图像分时获取。头戴显示利用两片体积较小的显示器加载左右视图，通过棱镜折射将视图传递到眼睛。以上方案仅提供双目视差。由于存在辐辏调节冲突，长期观看容易产生头晕、视觉疲劳等不适症状，对青少年的视力发育也会造成一定影响。

多视点三维显示的核心思想是通过有限的“视点”来近似表征连续的光场。多视点三维显示发展出了基于遮挡的多视点显示<sup>[6]</sup>(Occlusion based)，基于折射的多视点显示<sup>[7]</sup>(Refraction based)，基于衍射的多视点显示(Diffraction based)<sup>[8]</sup>，基于投影的多视点显示<sup>[9,10]</sup>(Projection based)和密集型多视点三维显示<sup>[11]</sup>(Super-multiview)等等。想要提高以上技术的视差连续性和图像分辨率，就需要借助高分辨率的显示设备。

体三维显示<sup>[12]</sup>利用空间中的体素发光来重建出三维场景的光场分布。根据体素是否运动可划分为动态体三维显示和静态体三维显示；根据体素能否自发光分为主动式和被动式体三维显示技术。体三维显示技术虽能够提供所有的深度信息，但仍然存在着遮挡效果不理想、系统造价高、结构复杂等问题。

英籍匈牙利物理学家丹尼斯·伽柏(Dennis Gabor)在1948年提出全息术<sup>[13]</sup>后，研究者对全息术进行了广泛、深入的研究。三维显示是全息术的重要应用领域之一<sup>[14-19]</sup>。全息图能够同时记录物光场的振幅和相位信息，其再现像可以提供人眼观察所需的所有深度线索，因此全息技术被认为是理想的三维显示技术<sup>[20,21]</sup>。早期的全息三维显示技术通过激光干涉将物体三维信息记录在介质上(如银盐干版、胶片等)，再通过再现光照明全息图得到原物体的三维再现像。激光全息记录过程中对震动非常敏感，对设备及环境的要求高，处理过程复杂，动态场景记录困难，并且只能记录实际存在的物体。

随着计算机技术与电子技术的不断发展，计算全息(Computer Generated Holography, CGH)应运而生<sup>[22,23]</sup>。它通过计算机数值计算代替光学记录过程来获得全息图，再将全息图加载到空间光调制器或输出到记录介质上实现三维物体的再现。这一技术的出现使得虚实场景结合的全息图制作成为现实<sup>[24-26]</sup>，并且大大推动了动态全息三维显示的发展<sup>[27-30]</sup>。由于现有的空间光调制器分辨率不够高、像素间

隔大等导致大视场三维重建存在困难。同时计算全息计算量巨大,对算法和硬件都有很高的要求。本文研究内容以提高计算机制全息图的计算速度为主要目标。

## 1.2 计算全息快速算法研究现状

庞大的计算量是实现高质量计算机制全息三维显示的瓶颈之一<sup>[31]</sup>。研究者在提高全息图的计算速度方面进行了不懈的研究,提出了一系列的快速算法。常见的 CGH 算法有点元法、面元法及层析法等等。点元法将物体分解为一系列离散发光点,每个点发出球面波。全息图由这些球面波在全息面上叠加后的物光与参考光干涉形成。该方法适合计算复杂场景的全息图。为提高点元法的计算速度,国内外学者提出了诸多算法。如查表法<sup>[32-34]</sup>(Look-up Table,LUT)、差分法<sup>[37]</sup>、分离变量法<sup>[38]</sup>及波前记录面<sup>[39-42]</sup>(Wave Recording Plane,WRP)法等等。查表法预先计算全息图基元并存储,制作全息图时通过查表获得基元信息再叠加,速度快但存储空间占用大。差分法利用全息面等间隔抽样的特点,每个物点在全息面上相邻样点间复振幅相位的差分为常数,相位计算只需要对差分值积分(只有加法运算),避免了原本计算中的平方、除法等运算,提高了全息图的计算速度。该方法计算比较灵活,但是速度比查表法慢。分离变量法根据每个物点在全息面上的复振幅计算可以分解为两个正交的方向,每个方向可以独立计算的特点,预先计算行、列复振幅分量然后在整个全息面上通过行、列分量组合得到完整的复振幅。费时的计算只在行、列分量计算中,而分量组合计算比原本的计算快得多,全息图的计算速度得到了提高。波前记录面法是在全息面与物体之间设置一平面(称为波前记录面),先计算在该平面上的物光波分布,然后利用快速傅里叶变换(Fast Fourier Transform,FFT)计算菲涅尔衍射,得到全息面上的物光复振幅分布。当波前记录面靠近物体时,由于每个物点发出的光波只分布在记录面上的一个小区内,这样大大降低了全息图的计算量。但该方法重建深度有限遮挡效果较差。面元法主要分为基于抽样的面元法<sup>[43-45]</sup>和解析面元<sup>[46,47]</sup>。同一物体的面元数量远小于点元数量,因此计算量比点元法少。但是该方法计算纹理复杂物体的全息图依然耗时。层析法<sup>[48-52]</sup>是将三维物体分为多个相互平行的平面,计算每一层物体衍射到全息面上的复振幅,然后叠加并与参考光干涉,最终得到全息图。由于层析法采用了 FFT,有效地提高了计算速度。但采用

FFT 计算物面到全息面上的衍射时, 抽样间隔需要满足一定的约束条件, 且物面与全息面的抽样点数必须保持一致。这往往就需要对物面的抽样数据进行插值或者补零。

当计算机制全息图的分辨率很高时<sup>[53-56]</sup>, 通常采用并行计算或光场与计算全息结合的方法进行全息图的计算。并行计算方法利用全息图计算中的独立性, 采用多台计算设备同时计算, 设备较为昂贵, 计算时电能消耗大。文献[57]提出采用服务器集群的并行计算方法实现高加速比的全息图计算。文献[58]提出采用 GPU 集群的并行算法, 在由 16 个节点构成、每个节点有 16 个 GPU 的超大规模并行机上实现了  $44000 \times 44000$  像素菲涅尔全息图的快速计算。光场与计算全息结合的方法利用一系列二维光场图像来计算全息图, 其运动视差是离散的。增加运动视差的平滑性需要密集的光场图像, 全息图的计算时间也随之明显增加。文献[59]提出采用频域拼接的方法实现了  $94340 \times 94340$  像素的彩色彩虹全息快速计算。文献[60]提出采用光场图像与全息透镜结合的方法实现了  $200000 \times 200000$  像素的全视差全息图快速计算。文献[61]提出采用角频率切片的由光场图像计算全息图的方法, 实现了  $25000 \times 25000$  像素的全息图快速计算。

### 1.3 本论文研究内容

论文介绍了计算全息的理论基础和常用计算模型。分析了离散菲涅尔衍射积分的特点, 提出了采用积分项周期延拓的全息图快速算法, 大大减少了计算过程中的冗余计算, 提高全息图的计算速度。为了提高全息图的三维重建效果提出了多视角隐藏面消除算法。设计了基于 OpenGL 的三维数据处理程序, 实现任意视角下获取消隐后的三维模型数据及纹理。为进一步提高全息图的计算速度, 设计并实现了采用多核 CPU 的并行计算方法。最后为验证以上算法的可行性及性能, 搭建了实验平台进行实验, 同时对实验结果进行了分析。

本论文的结构如下:

第一章简要总结了三维显示技术的主要类型。通过对国内外的文献调研, 概括了计算全息快速算法的研究现状。分析了现有技术的优缺点。

第二章首先介绍了全息图的记录和再现原理及计算全息的基础—抽样理论和信

息容量。之后介绍了层析法计算机制全息图计算模型。

第三章从菲涅尔衍射积分的计算模型出发,进行分析研究。通过分析发现菲涅尔衍射积分公式离散化后积分项具有周期性,积分项外系数能够行列分解。设计并实现了采用积分项周期延拓的计算全息快速算法。另外,还给出了层析法计算全息计算中三维点云数据的处理方法。

第四章设计了大视角计算机制全息图的多视角消隐算法。从算法的思路和实现两个方面做出详细的介绍。同时为进一步提高全息图的计算速度,设计并实现了基于多核 CPU 的多线程并行计算方法。

第五章为系统搭建及实验。介绍了搭建的实验系统,给出了所提快速算法的光学再现结果。对比了快速算法与传统层析法计算速度。给出了多视角消隐处理后的结果及并行计算的速度提高情况。对实验结果进行了分析和总结。

第六章总结了全文的核心内容,提出了可改进部分和下一步工作的方向。

## 2 计算全息原理

### 2.1 全息术理论基础

全息术的两个基本问题是记录和重建物光波的振幅和相位信息。采用干涉方法记录物光波的波前信息生成全息图，这一过程称之为波前记录。通过衍射来重建全息图所记录的物光波称之为波前重建<sup>[62]</sup>。

#### 2.1.1 波前记录

波前记录所解决的问题是如何完整地记录物光波的振幅和相位信息。目前所有的光学记录介质都只对光强敏感，无法记录光波的相位信息。想要记录相位信息就必须将相位信息转化为强度分布，这一过程称为相位编码。干涉法能将空间相位调制转化为空间强度调制，是一种标准的相位编码方法<sup>[62]</sup>。

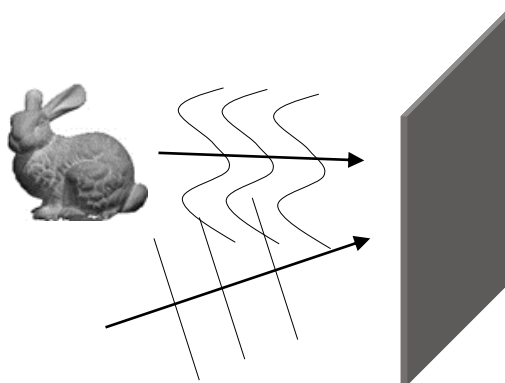


图 2.1 全息术波前记录

如图 2.1 所示，物光与参考光复振幅分布用  $o(x,y)$  和  $r(x,y)$  表示

$$O(x,y) = |O(x,y)| \exp[j\phi_o(x,y)] \quad (2.1)$$

$$R(x,y) = |R(x,y)| \exp[j\phi_r(x,y)] \quad (2.2)$$

记录介质上所得到的光强为物光和参考光相干叠加的结果：

$$\begin{aligned}
I(x, y) &= |O(x, y) + R(x, y)|^2 \\
&= |O(x, y)|^2 + |R(x, y)|^2 + O(x, y)R^*(x, y) + O^*(x, y)R(x, y) \\
&= |O(x, y)|^2 + |R(x, y)|^2 + 2|O(x, y)||R(x, y)|\cos[\varphi_o(x, y) - \varphi_r(x, y)]
\end{aligned} \tag{2.3}$$

由式(2.3)可以看出，其前两项为物光和参考光的强度信息，第三项为干涉项。它包含了物光、参考光及它们的相对相位关系。记录的干涉图包含了物光的振幅和相位信息。

### 2.1.2 波前重建

物光波的振幅和相位信息通过编码记录到全息图上后，就可以通过光学或者数字的方式对光波进行重建。这一过程以全息图对重建光波的衍射为基础<sup>[62]</sup>。用一相干光波照射记录好的全息图，得到的衍射光波可以表示为

$$\begin{aligned}
u(x, y) &= B(x, y)t(x, y) \\
&= t_b B(x, y) + \beta' \left[ |O(x, y)|^2 + |R(x, y)|^2 \right] \\
&\quad + \beta' O(x, y)R^*(x, y)B(x, y) + \beta' O^*(x, y)R(x, y)B(x, y) \\
&= u_1 + u_2 + u_3 + u_4
\end{aligned} \tag{2.4}$$

其中  $B(x, y)$  表示重建光波，利用记录全息图时的参考光波作为重建光波时，式(2.4)中的  $u_3$  项表示为

$$u_3 = \beta' |R(x, y)|^2 O(x, y) \tag{2.5}$$

此项表示了原物光的重建。如图 2.2(a) 所示，人眼可以观察到原始物体的虚像。

如果将重建光场  $B(x, y)$  换成原参考光的共轭光  $R^*(x, y)$  此时式(2.4)中的第  $u_4$  项表示为

$$u_4 = \beta' |R(x, y)|^2 O^*(x, y) \tag{2.6}$$

其为原物光的共轭重建。如图 2.2(b) 所示，透过全息图的衍射光波会在空间中汇聚成一个实像，其深度与原始物体的相反。



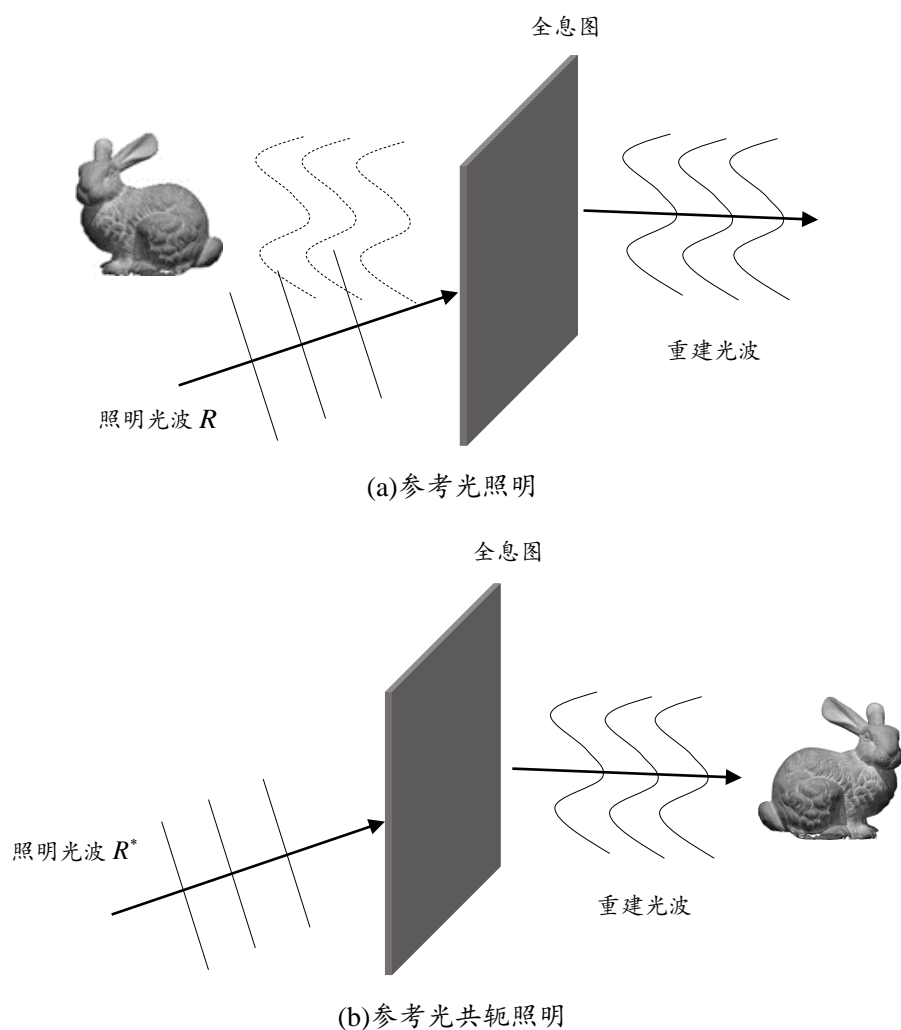


图 2.2 波前重建

## 2.2 计算全息理论基础

随着计算机科学的发展,美国 IBM 公司的德国光学专家罗曼(A.W.Lohmann)做出了世界上第一个计算全息图<sup>[63]</sup>(computer-generated hologram,CGH)。传统的光学全息是采用光学干涉法在记录介质上记录物光与参考光的干涉图样。该方法只能记录实际物体的全息图,而计算全息只要将实际物体或虚构物体的数学模型输入到计算机中,就可以实现全息图的记录,在三维显示领域有着重要的科学意义。计算全息图的制作和再现主要经过抽样、计算、编码、成图、再现五步。

### 2.2.1 抽样定理

抽样定理是计算全息中的重要理论基础<sup>[64,65]</sup>，它是离散信号与连续信号联系的重要桥梁。在自然界中多数信号以连续形式存在，想要在离散系统中实现连续信号处理，就需要对其进行离散化。计算全息术也不例外，采用数字计算机模拟全息图记录的过程中，必然涉及离散信号处理。

现有一维连续信号  $o(x)$  ( $x$  表示时间或空间坐标)，被理想脉冲函数进行规则抽样，抽样间隔为  $\Delta x$ 。抽样后的信号  $o_s(x)$  表示为

$$\begin{aligned} o_s(x) &= o(x) \sum_{m=-\infty}^{+\infty} \delta(x - m\Delta x) \\ &= \frac{1}{\Delta x} o(x) \text{comb}\left(\frac{x}{\Delta x}\right) \end{aligned} \quad (2.7)$$

抽样前后的信号分别表示为图 2.3 的 (a) 和 (b)。图 2.3 (b) 中的虚线表示原连续信号。根据卷积定理  $o_s(x)$  的频谱  $O_s(f)$  为：

$$\begin{aligned} O_s(f) &= O(f) * \text{comb}(f\Delta x) \\ &= O(f) * \left[ \frac{1}{\Delta x} \sum_{m=-\infty}^{+\infty} \delta\left(f - \frac{m}{\Delta x}\right) \right] \end{aligned} \quad (2.8)$$

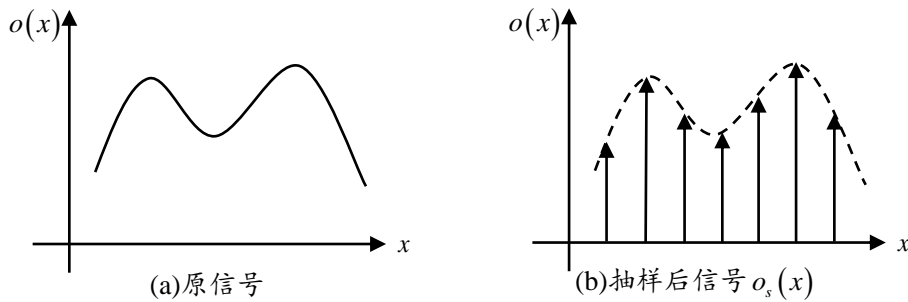
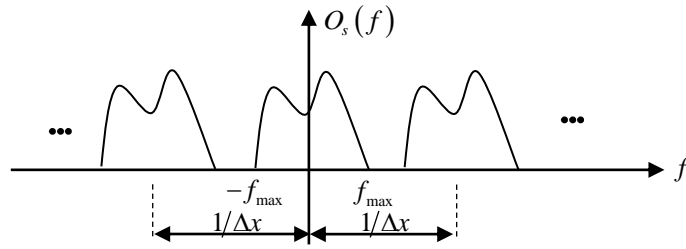


图 2.3 原函数与抽样函数

由  $\delta$  函数的性质，式 (2.8) 可以表示为：

$$O_s(f) = \frac{1}{\Delta x} \sum_{m=-\infty}^{+\infty} O\left(f - \frac{m}{\Delta x}\right) \quad (2.9)$$

其图像如图 2.4 所示， $O_s(f)$  的频谱为原信号  $o(x)$  的频谱  $O(f)$  以  $1/\Delta x$  为周期延拓得到。

图 2.4 抽样后信号的频谱  $O_s(f)$ 

在实际应用中信号带宽都是有限的，其频率最大值为  $f_{\max}$ ，原信号的频谱  $O(f)$  只有在  $-f_{\max} \leq f \leq f_{\max}$  中才有非零值。由图 2.4 可知，为防止延拓后发生混叠，必须满足条件：

$$\Delta x \leq 1/(2f_{\max}) \quad (2.10)$$

此条件规定了抽样后信号不失真时抽样间隔的大小。满足条件时用  $O_s(f)$  乘以矩形窗函数即可得到  $O(f)$  从而恢复出原信号  $o(x)$ 。在计算全息图计算中物面(空域)和全息面(空频域)的抽样均要满足抽样定理。

### 2.2.2 计算全息信息容量

计算机分析和处理光场的二维分布时，需满足抽样理论。在对图像抽样时，抽样点过疏将无法保证精度，过密则会带来冗余的计算量和存储量<sup>[63]</sup>。因此合理的选择抽样间隔达到既不丢失信息，又不带来冗余计算是至关重要的。图像在光学系统中的传递受孔径光阑和视场光阑的约束<sup>[64]</sup>。用空间带宽积(SW)来描述光学信道的信息量，其表达式为：

$$SW = \iint dx dy \iint d\xi d\eta \quad (2.11)$$

假定图像在空域和频域所占据的面积都是矩形，其边长分别为  $\Delta x$ ， $\Delta y$ ， $\Delta \xi$ ， $\Delta \eta$ ，则上式可写成：

$$SW = \Delta x \Delta y \Delta \xi \Delta \eta \quad (2.12)$$

当图像发生位移、缩放时空间带宽积保持不变。空间带宽积还确定了图像上可分辨的像素数量，据此可以确定计算全息图上的点数。

## 2.3 全息图的层析计算方法

层析法是将三维物体分解成多个平行平面，通过计算所有平面在全息面上的衍射场并叠加得到总衍射场。这种方法可以有效地利用 FFT 提高计算速度，层析法常用的计算方法有角谱衍射法和菲涅尔衍射法等。

### 2.3.1 层析角谱法

采用角谱法进行衍射计算不引入傍轴近似，可以高精度模拟波前的传播过程。

图 2.5 所示为层析角谱法的计算流程。

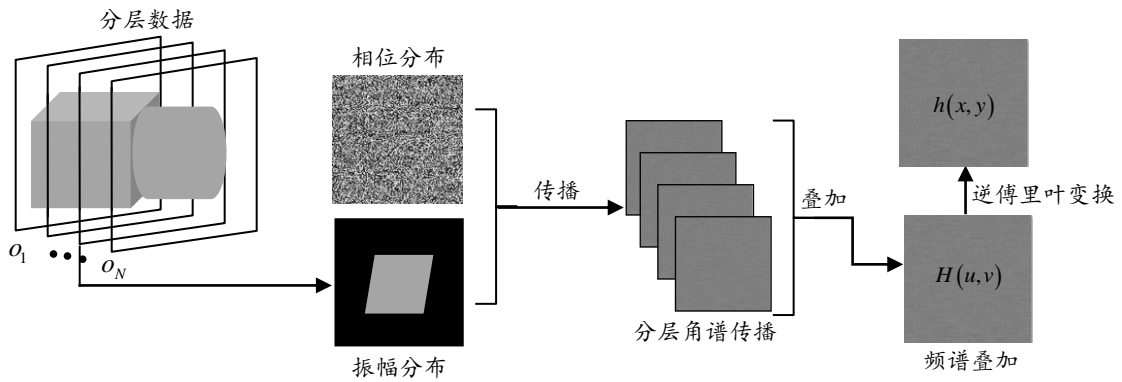


图 2.5 层析角谱法的计算流程

对三维模型进行分层，每一层图像加上随机相位作为输入波前进行角谱传播。得到每一层图像在全息面上的频谱分布  $H_i(u, v)$ ，然后叠加得到整个三维场景在全息面上的频谱分布  $H(u, v)$ ，最后利用逆傅里叶变换计算出全息面上的复振幅分布。计算过程可表示为

$$h(x, y) = F^{-1} \left\{ \sum_{i=1}^N F[o_i(x, y)] \exp(jkz_i \sqrt{1 - \lambda^2 u^2 - \lambda^2 v^2}) \right\} \quad (2.13)$$

式中  $o_i(x, y)$  为第  $i$  层的复振幅分布， $z_i$  是第  $i$  层到全息面的距离，式中的傅里叶变换均可通过 FFT 进行快速计算。

### 2.3.2 层析菲涅尔衍射法

如图 2.6 所示，物体沿  $z$  方向分层， $z_i$  表示第  $i$  层到全息面的距离，根据菲涅尔衍射积分公式可求得该层在全息面上的衍射场分布为

$$u_i(x, y) = \frac{\exp(jkz_i)}{j\lambda z_i} \exp\left[j\frac{k}{2z_i}(x^2 + y^2)\right] \times \iint o(x_0, y_0) \exp\left[j\frac{k}{2z_i}(x_0^2 + y_0^2)\right] \exp\left[-j2\pi\left(\frac{x}{\lambda z_i}x_0 + \frac{y}{\lambda z_i}y_0\right)\right] dx_0 dy_0 \quad (2.14)$$

其中  $k$  为波数,  $\lambda$  为波长,  $(x, y)$  对应全息面坐标,  $(x_0, y_0)$  对应物面坐标。 $o(x_0, y_0)$  表示第  $i$  层数据的光场分布。全息面上总的衍射场分布可以表示为

$$u(x, y) = \sum_{i=1:n} u_i(x, y) \quad (2.15)$$

将其与参考光干涉, 得到全息图。相比于角谱法, 采用菲涅尔衍射计算每层的计算仅做一次 FFT。但是菲涅尔衍射需要满足旁轴近似条件。

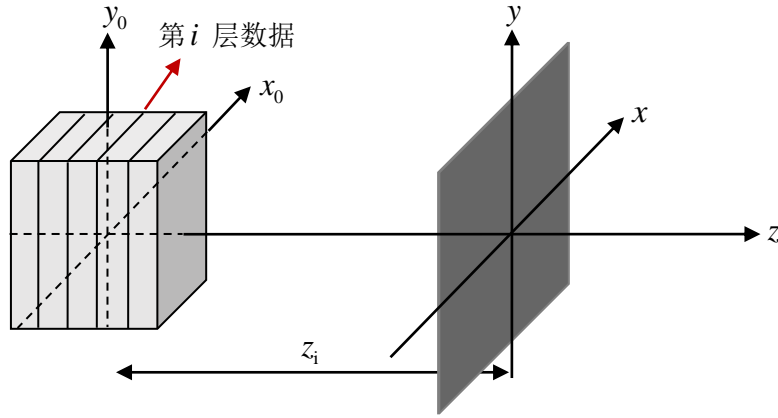


图 2.6 层析法计算模型

层析法的一大优点是分层数确定时计算量不随场景的复杂程度而改变。另外衍射计算可以采用快速傅里叶变换大幅的提升计算速度。需要重点考虑的是层与层之间遮挡的消除<sup>[66,67]</sup>。

## 2.4 本章小结

本章介绍了全息图记录与再现原理及计算全息的两个重要理论基础：抽样定理、信息容量。讨论了层析法计算全息图原理。

### 3 周期延拓计算全息快速算法

#### 3.1 算法原理

##### 3.1.1 离散菲涅尔衍射公式分析

图3.1为层析法计算菲涅尔全息图的流程示意图。 $R$ 表示参考光， $H$ 表示全息面。首先对物体沿 $z$ 轴分层，然后对每一层的数据计算其在全息面上的光场复振幅分布，将得到的所有复振幅分布求和并与参考光干涉得到要记录的全息图。

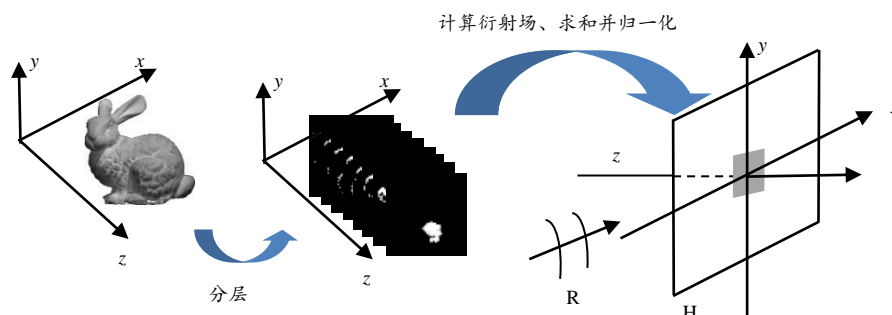


图3.1 层析法菲涅尔全息图计算流程

采用菲涅尔衍射计算时，在计算每一层数据在全息面上的衍射场分布时需要通过菲涅尔衍射积分公式来完成。式(2.14)可以写成

$$u(x, y) = \frac{\exp(jkz)}{j\lambda z} \exp\left[j\frac{k}{2z}(x^2 + y^2)\right] u'(x, y) \quad (3.1)$$

式中 $u(x, y)$ 表示全息面上物光波的复振幅分布， $k=2\pi/\lambda$ ， $\lambda$ 为波长，

$$u'(x, y) = \iint o(x_0, y_0) \exp\left[j\frac{k}{2z}(x_0^2 + y_0^2)\right] \exp\left[-j2\pi\left(\frac{x}{\lambda z}x_0 + \frac{y}{\lambda z}y_0\right)\right] dx_0 dy_0 \quad (3.2)$$

积分项式(3.2)为二维傅里叶变换。采用计算机数值计算时对空域(物面)和空频域(全息面)进行抽样离散化后，利用FFT算法实现。在水平、垂直方向抽样间隔相同的情况下，采用 $\Delta_o$ 和 $\Delta_h$ 分别表示物面和全息面的抽样间隔。对物面离散化后式(3.2)

表示为

$$\begin{aligned}
 \hat{u}'(x, y) &= \iint o(x_0, y_0) \exp \left[ j \frac{k}{2z} (x_0^2 + y_0^2) \right] \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} \delta(x_0 - m\Delta_o) \delta(y_0 - n\Delta_o) \\
 &\quad \exp \left[ -j2\pi \left( \frac{x}{\lambda z} x_0 + \frac{y}{\lambda z} y_0 \right) \right] dx_0 dy_0 \\
 &= \frac{1}{\Delta_o^2} u'(x, y) * \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} \delta(x - m\lambda z / \Delta_o) \delta(y - n\lambda z / \Delta_o) \\
 &= \frac{1}{\Delta_o^2} \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} u'(x - m\lambda z / \Delta_o, y - n\lambda z / \Delta_o)
 \end{aligned} \tag{3.3}$$

由式(3.3)可知, 物面离散化后全息面上的积分项是一个周期函数, 是连续情况下全息面上积分项的周期延拓。离散傅里叶变换 (Discrete Fourier Transformation, DFT) 是取式(3.3)的一个周期并抽样离散化, FFT 是 DFT 的快速算法。因此 FFT 实际隐含了周期性。当全息面或物面水平和垂直方向的抽样点数分别为  $M$  和  $N$  时, 根据 FFT 原理, 物面和全息面的尺寸必须满足

$$M = N = \frac{\lambda z}{\Delta_o \Delta_h} \tag{3.4}$$

则式(3.3)水平和垂直方向的周期分别为  $M\Delta_h$  和  $N\Delta_h$ 。

全息图尺寸与观察视角有关。如图 3.2 所示, 在记录物体大小一定时, 人眼透过全息图观察再现像, 只有部分与瞳孔对应区域的衍射光进入人眼。人眼移动时, 对应的全息图区域也随之移动, 则观察到不同视角处的再现像。由此可见, 在物体大小、全息面与物面距离确定且全息面抽样间隔满足要求时, 全息图越大, 其可观察视角越大。由式(3.4)可知, 全息图越大, 要求物面的抽样间隔越小。人眼是一个低通系统分辨率有限, 物面抽样间隔小到一定程度, 超过了人眼分辨率, 会造成信息冗余, 增加不必要的存储和计算量。因此, 在大视角全息图计算时, 以人眼分辨率为基准确定物面的抽样间隔。物面大小确定后, 这一抽样间隔对应的只是某一视角下与瞳孔对应的小部分全息图。增加视角, 全息图尺寸必须增加。由式(3.4)知, 必须减小物面抽样间隔。

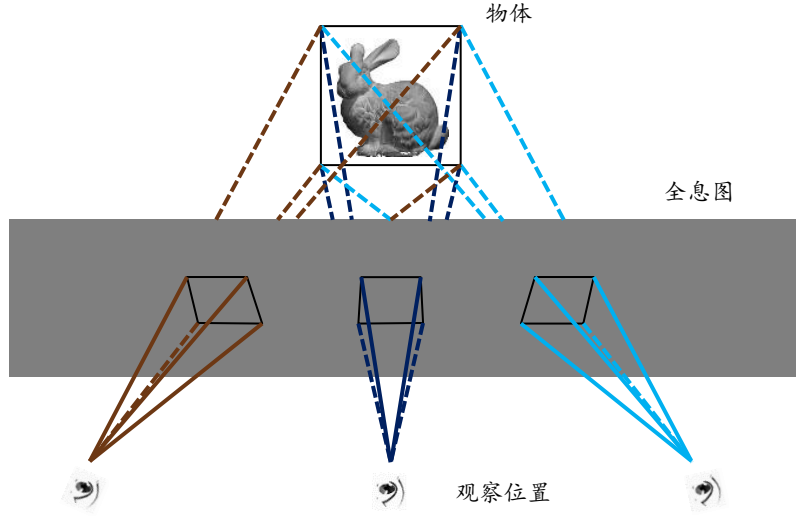


图 3.2 在不同位置透过全息图观察再现像

在数字信号处理中，整数倍插值用以下模型描述：在原有样点之间插零，然后进行数字低通滤波。设抽样后的物面复振幅为  $o(p\Delta_o, q\Delta_o)$ ，在物面相邻样点间插  $k_x, k_y$  个零，数据抽样间隔变为  $\Delta_o/k_x$  和  $\Delta_o/k_y$ ，则插零后的物面复振幅表示为  $o(m'\Delta_o/k_x, n'\Delta_o/k_y)$ 。根据式 (3.3)，积分项变为

$$\begin{aligned} \hat{u}'(x, y) = & \sum_{m'=-\infty}^{\infty} \sum_{n'=-\infty}^{\infty} o\left(\frac{m'\Delta_o}{k_x}, \frac{n'\Delta_o}{k_y}\right) \exp\left\{j\frac{k}{2z}\left[\left(\frac{m'\Delta_o}{k_x}\right)^2 + \left(\frac{n'\Delta_o}{k_y}\right)^2\right]\right\} \\ & \exp\left[-j2\pi\left(\frac{x}{\lambda z} \cdot \frac{m'\Delta_o}{k_x} + \frac{y}{\lambda z} \cdot \frac{n'\Delta_o}{k_y}\right)\right] \end{aligned} \quad (3.5)$$

这是一个周期为  $(\lambda z k_x / \Delta_o, \lambda z k_y / \Delta_o)$  的周期函数。插零前后的物面抽样点关系可表示为

$$o\left(\frac{m'\Delta_o}{k_x}, \frac{n'\Delta_o}{k_y}\right) = \begin{cases} o(p\Delta_o, q\Delta_o), & m' = 0, \pm k_x, \pm 2k_x, \dots; n' = 0, \pm k_y, \pm 2k_y, \dots \\ 0, & \text{其它} \end{cases} \quad (3.6)$$

则式 (3.5) 可表示为

$$\begin{aligned} \hat{u}'(x, y) = & \sum_{p=-\infty}^{\infty} \sum_{q=-\infty}^{\infty} o(p\Delta_o, q\Delta_o) \exp\left\{j\frac{k}{2z}[(p\Delta_o)^2 + (q\Delta_o)^2]\right\} \\ & \exp\left[-j2\pi\left(\frac{x}{\lambda z} \cdot p\Delta_o + \frac{y}{\lambda z} \cdot q\Delta_o\right)\right] \end{aligned} \quad (3.7)$$

式 (3.7) 是周期为  $(\lambda z / \Delta_o, \lambda z / \Delta_o)$  的周期函数，周期为式 (3.5) 的  $1/k_x$  和  $1/k_y$ 。因此，插零后积分项的一个周期内有  $k_x \times k_y$  个周期的原积分项。综上所述，物面插零积分项周期



变大，变大部分是原积分项的周期延拓。插零后积分项一个周期的分布可表示为

$$\hat{u}''(x, y) = \sum_{m=0}^{k_x} \sum_{n=0}^{k_y} u'(x - m\lambda z / \Delta_o, y - n\lambda z / \Delta_o) \quad (3.8)$$

如图3.2所示，人眼透过全息图观察再现像时，只有部分全息图上衍射的光进入人眼。相当于进行了如下运算

$$\begin{aligned} \hat{u}'''(x, y) &= \hat{u}''(x, y) \text{Rect}\left(\frac{x - x_c}{X}, \frac{y - y_c}{Y}\right) \\ &= \text{Rect}\left(\frac{x - x_c}{X}, \frac{y - y_c}{Y}\right) \sum_{m=0}^{k_x} \sum_{n=0}^{k_y} u'(x - m\lambda z / \Delta_o, y - n\lambda z / \Delta_o) \end{aligned} \quad (3.9)$$

式中， $(x_c, y_c)$ 是眼睛对应的全息图窗口中心，**Rect**表示二维矩形窗函数， $X$ 、 $Y$ 分别是窗口 $x$ 、 $y$ 方向的宽度。当窗口宽度等于积分项周期时，一个完整的周期信息进入人眼。此时式(3.9)的逆傅里叶变换就是内插公式形式的抽样定理，而全息图衍射再现就实现了逆傅里叶变换。这样，在全息面积分项周期延拓再利用人眼的低通滤波特性实现物面的插值运算，减少了大量冗余计算并且省去了数字插值中的数字低通滤波运算。

由上述分析可知，对于大视角计算全息图的积分项，在抽样间隔满足人眼分辨率时，只需计算其中一个周期，而其他部分直接周期延拓即可。设要计算的全息图大小为 $M \times N$ ，将已知参量代入约束条件式(3.4)中计算出全息面积分项一个周期的像素数 $K \times L$ 。此时 $x$ 、 $y$ 方向上积分项的周期数为 $k_x = \text{round}(N/L)$ 和 $k_y = \text{round}(M/K)$ ，在传统算法中，需要根据计算的全息图大小和衍射距离等参量对物体的分层数据进行重抽样或补零到与全息图相同大小，再对每一层数据做菲涅尔衍射计算求出全息面上的衍射场。此过程中利用FFT计算积分项的计算量为 $MM \log_2 MN$ 。新算法中用FFT计算积分项的计算量变为 $(MN/k_x k_y) \log_2^{MN/k_x k_y}$ 。可见，当需要计算的全息图越大(即包含的周期数越多)时，新算法的计算速度优势越明显。通常物体与全息面水平和垂直方向的抽样间隔相同，要求积分区域必须是正方形。对于非正方形全息图，通常通过物面补零到正方形计算后再裁剪来达到这一要求。在长宽比大的全息图计算时，本算法能够进一步减少冗余计算量。

### 3.1.2 积分项系数特点

用 $u_c(x, y)$ 表示式(3.1)的系数(后续表述中统称为系数), 即

$$u_c(x, y) = \frac{\exp(jkz)}{j\lambda z} \exp\left[j\frac{k}{2z}(x^2 + y^2)\right] \quad (3.10)$$

在计算某一层物体的衍射场时,  $z$ 为定值则 $\exp(jkz)/j\lambda z$ 为常数, 显然后一项可以对 $x, y$ 进行变量分离, 表示为

$$\begin{aligned} u_{cx}(x) &= A \exp\left(j\frac{k}{2z}x^2\right) \\ u_{cy}(y) &= \exp\left(j\frac{k}{2z}y^2\right) \end{aligned} \quad (3.11)$$

式中,  $A$ 表示常数 $\exp(jkz)/j\lambda z$ 。计算 $M \times N$ 大小的全息图时对整个全息面计算 $u_c(x, y)$ 需要进行 $2M \times N$ 次三角函数与 $4M \times N$ 次乘法计算。式(3.11)分离变量后, 系数 $u_c(x, y)$ 在全息面上的分布分解成行列相互独立的分量 $u_{cx}$ 和 $u_{cy}$ 。因此只需要计算长度分别为 $1 \times M$ 和 $1 \times N$ 的两列数据存储在内存中, 将积分项周期性延拓后乘上对应位置处的行、列系数即可得到全息面上的衍射场复振幅分布。此时对系数 $u_c(x, y)$ 的计算变为 $2(M+N)$ 次三角函数与 $2(M \times N + M + N) + M$ 次乘法计算。由于三角函数计算耗时较长, 采用行列分解后明显提高了计算速度。

## 3.2 三维模型处理

在全息图记录前需要完成三维数据的准备工作。其中最重要的是对物体进行分层和隐藏面消除, 以保证对三维物体进行准确的再现。对于点云及纹理图表示的模型, 选用如下方案进行数据准备。

如图 3.1 所示, 将点云数据沿着  $z$  轴分层。 $\Delta z$  范围内的数据作为一层数据。 $\Delta z$  的选择需要考虑记录物体的物理尺寸、人眼在对应观察距离上的分辨率、景深等因素。 $\Delta z$  选择过小会导致层数增加计算时间延长, 选择过大则会导致观察再现像时出现物体分层。对于给定的物体, 沿  $z$  轴逆向观察时, 较远层上的点可能会被前一层的点遮挡。若不进行处理, 全息图的再现像会出现透明感, 不够逼真。另外, 在分层后的每一层数据中有数据的像素点并不是连续分布的。需要先将数据点变成连续

区域方便纹理贴图。首先制作与每层数据对应的二值模板，有数据部分为 1，无数据部分为 0，如图 3.3 所示。之后用大小为  $m \times n$  的矩形窗口在一层数据上进行检测。结果表示为：

$$Q = \sum_{i=1:m, j=1:n} q(i, j) \quad (3.12)$$

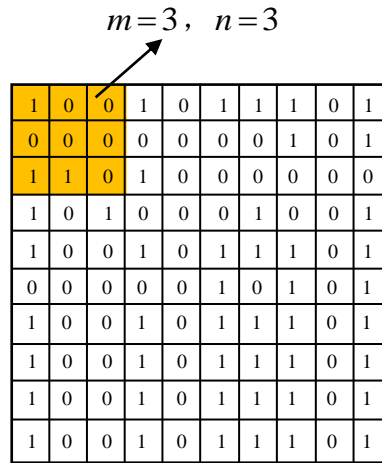


图 3.3 点云数据连续化处理示意图

$q(i, j)$  表示窗口取到的像素值，当窗口内的有效数据量  $Q$  超过设定域值  $a$  后将窗口内的所有值设为 1。之后移动到该层数据的下一窗口位置处理，直到遍历整个矩阵。处理结果理想则结束对该层数据的连续化处理。若效果较差可在此基础上变换窗口初始位置重复上述步骤。根据实际情况调节窗口大小、域值  $a$  和窗口处理次数可将每一层上的有效数据点处理成相对连续的区域。图 3.4 为上述处理方法的两组实验结果，上方两幅图是点云数据分层后的两层。可以明显看到有数据位置是离散分布的，这样不便于纹理贴图。下方两幅图是上方图像对应的处理后的结果，可以看到经过处理变成了连续的分布。

完成连续化处理之后，需要进行隐藏面的消除。用一个与分层数据像素数一致的全 1 矩阵  $P$  记录有数据的像素点位置，用来消除隐藏面。具体过程是第一层数据取反，与矩阵  $P$  相乘得到新的  $P$  矩阵  $P1$ 。实现了对第一层数据上有数据点位置的记录。此时  $P1$  矩阵中的 0 值都是第一层数据上的数据位置。接下来用第二层数据取反和  $P1$  矩阵相乘得到新的  $P$  矩阵  $P2$ ， $P2$  中标 0 的位置是前两层数据中有数据的位置。同时用第二层数据直接与  $P1$  矩阵相乘得到消隐后的第二层数据。以此类推完成对每

层数据的消隐，流程如图 3.5。

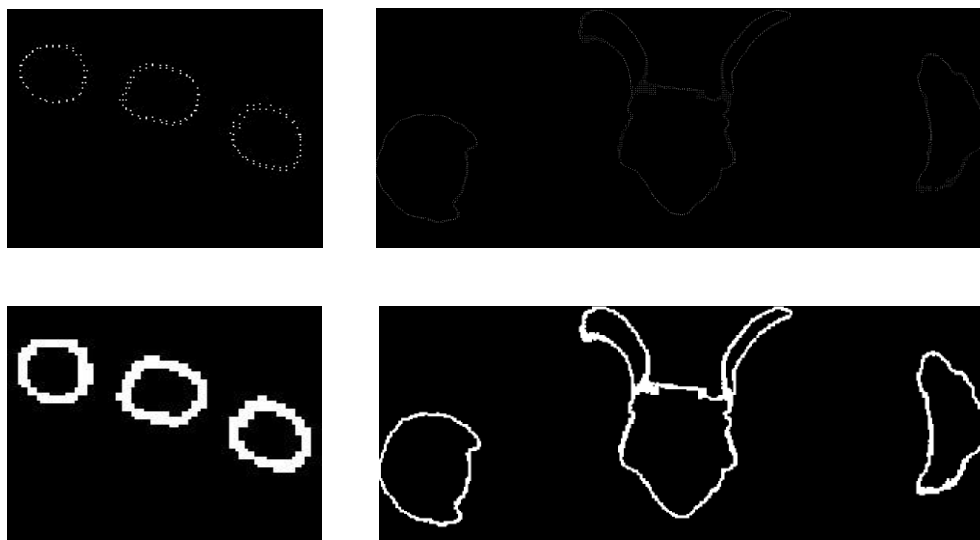


图 3.4 数据连续化处理前后对比

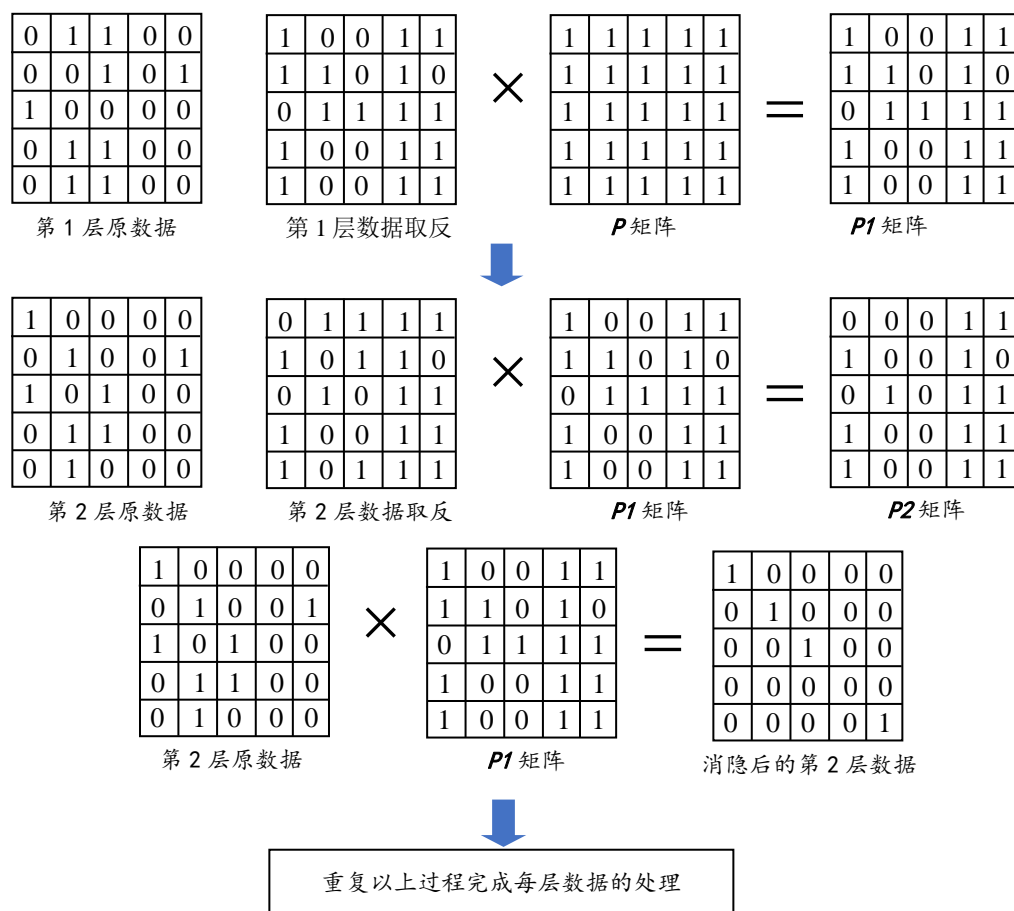


图 3.5 消隐处理示意图

### 3.3 算法流程

根据以上分析，将菲涅尔衍射计算分为系数和积分项两部分。系数部分采用行、列正交分解再合成计算；积分项部分采用计算一个周期再周期延拓；最后两部分合成得到全息面上的物光复振幅分布。快速算法具体流程为：

- 1) 读入点云数据，先进行分层后对数据进行处理；
- 2) 计算一层物体在全息面上积分项的一个周期，将求得的积分项存入内存；
- 3) 根据最终要计算的全息图大小计算该层数据对应系数行、列分量存入内存；
- 4) 将该层的积分项周期延拓，根据对应位置的行号列号获取系数分量相乘得完整系数，再将系数与积分项相乘得到该位置的物光复振幅，并与前一次计算好的复振幅叠加；
- 5) 重复步骤 2)~4) 直到所有层计算完成；
- 6) 将叠加后的复振幅与参考光干涉，经编码得到全息图。

### 3.4 本章小结

为提高大尺寸计算机制全息图的计算速度，本章通过分析基于层析法的菲涅尔全息图计算模型的特点，提出一种新的计算机制菲涅尔全息图快速算法。首先分析得出离散菲涅尔衍射公式中积分项具有周期性，巧妙利用人眼的低通滤波特性，提出周期延拓法计算积分项，大大降低了计算时间。之后根据系数可行列分解的特性，降低了计算过程中的计算量。其次给出了点云表示的三维模型处理方法。最后给出了快速算法流程。

## 4 全息图消隐及多核并行算法

### 4.1 大视角全息图消隐

3.2 节所介绍的方法仅能够对模型进行简单的处理，很难实现高质量的三维重建。消隐也只能是在单一视角下完成。在全息三维显示中，大视角是追求的一个重要目标。前述方法不能够满足实际的观察需求。需要有新的高精度消隐算法。

#### 4.1.1 全息图多视角消隐算法

在三维世界中从不同角度观察场景中的物体时，所观察到的结果会不同。想要通过全息三维显示呈现相同的视觉效果，仅仅在一个视角下作隐藏面消除是远远不够的。有场景如图 4.1 所示，物体 1 是一个红色背景平面，物体 2 是一个蓝色球体。两物体在观察位置 2 处做了隐藏面消除，物体 1 被 2 遮挡的部分用白色代替。在此位置上观察到的结果如图 4.2(b)，在观察位置 1 和观察位置 3 的结果如图 4.2(a) 和 4.2(c)。显然(a)和(c)不符合实际的观察结果。当在位置 1、2、3 分别做消隐后得到的结果是图 4.2 的(d)、(e)、(f)所示，此时与观察实际物体效果相同。

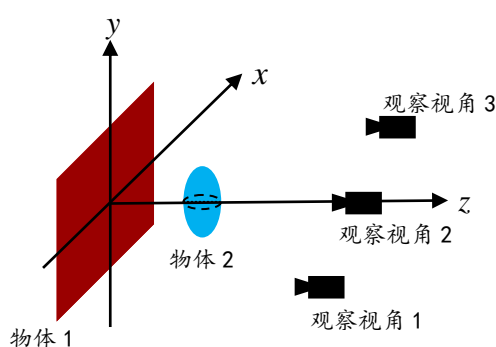


图 4.1 观察角度与物体位置关系

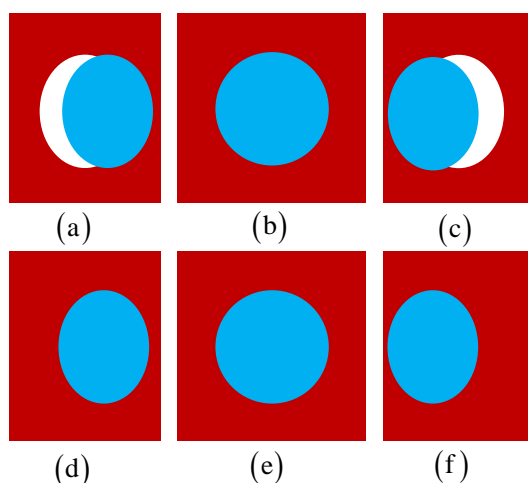


图 4.2 仅位置 2 消隐时，(a)、(b)、(c) 为位置 1、2、3 的观察结果；  
(d)、(e)、(f) 为分别在三个位置消隐后 1、2、3 的观察结果

对大视角的全息图做多视角消隐，首先将全息面分为多个区域(区域大小的选择根据观察角度，全息面与物体之间位置关系等参数确定)。在每个区域中根据对应视角对模型做隐藏面消除。在得到每个区域的消隐数据后，计算不同区域下积分项的

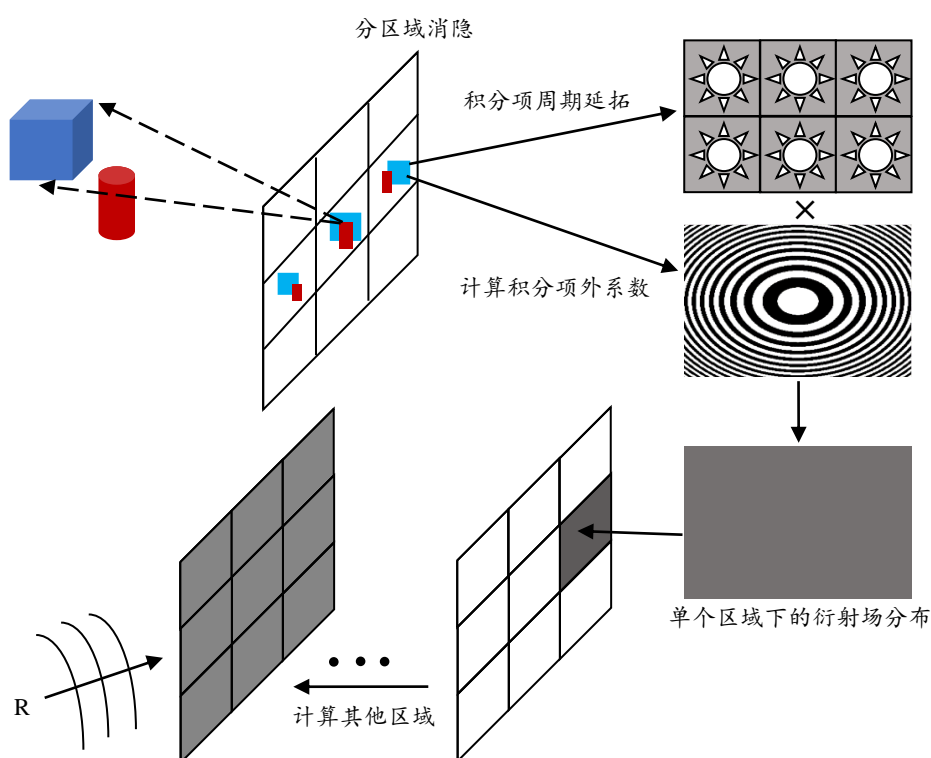


图 4.3 多视角消隐算法流程

一个周期，并将积分项在对应的整个区域周期延拓。计算各区域的积分项外系数。

各区域的系数与积分项分别相乘得到对应的衍射场分布。所有区域的衍射组合，最后与参考光干涉得到全息图。

### 4.1.2 算法流程

算法的流程示意图如图 4.3 所示。综上所述程序设计思路如下：

- 1) 根据参数确定全息图的视角分区；
- 2) 加载模型；
- 3) 根据确定的视角区域调节相机位置，场景渲染；
- 4) 获取当前视角下消隐后数据并存储；
- 5) 重复 3)、4)直到完成所有视角数据的消隐；
- 6) 积分项周期延拓法计算各分区衍射场；
- 7) 与参考光干涉，得到最终的全息图。

### 4.1.3 消隐算法实现

在大视角消隐算法中消隐处理采用 OpenGL 来完成。OpenGL 称为开放图形库，提供用于渲染 2D、3D 矢量图形的跨语言、跨平台的应用程序编程接口(API)。待消隐的物体处在三维空间中，而屏幕却是二维的，因此 OpenGL 的大部分工作是把三维坐标转换为适应屏幕显示的二维坐标。这一处理过程由 OpenGL 的图形渲染管线(Graphics Pipeline)来完成。图形渲染管线的另一个重要作用是计算二维坐标下像素的实际颜色值。

图 4.4 给出了图形渲染管线的流程图。其中顶点着色器、片段着色器允许用户根据不同需要个性化编程，实现更为细致的控制图形渲染管线。而且着色器的实现运行在 GPU 上，可以为开发者节约 CPU 时间。顶点着色器的主要目的是把三维坐标转为另一种三维坐标。片段着色器的主要目的是计算每一个像素的最终颜色，这也是所有 OpenGL 高级效果产生的地方。



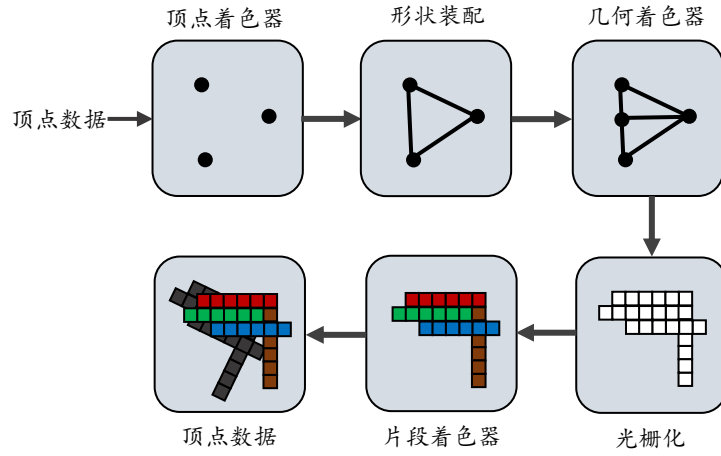


图 4.4 图形渲染管线流程

图 4.5 为顶点着色器中坐标变换的流程图，首先是三维坐标间的转换。表达式为

$$\begin{bmatrix} x_{vi} \\ y_{vi} \\ z_{vi} \\ 1 \end{bmatrix} = \begin{bmatrix} v_{11} & v_{12} & v_{13} & v_{14} \\ v_{21} & v_{22} & v_{23} & v_{24} \\ v_{31} & v_{32} & v_{33} & v_{34} \\ v_{41} & v_{42} & v_{43} & v_{44} \end{bmatrix} \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \\ m_{41} & m_{42} & m_{43} & m_{44} \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix} \quad (4.1)$$

制作好的三维模型处在自身坐标系下，称为局部坐标系。 $x_i, y_i, z_i$  表示三维模型在局部坐标系下的坐标。将局部坐标系转化到真实场景中，得到模型在世界坐标系下的坐标，要通过左乘  $m$  矩阵来实现。再左乘  $v$  矩阵将世界坐标系下的模型转化到观察坐标系中， $x_{vi}, y_{vi}, z_{vi}$  表示三维模型在观察坐标系下的坐标。之后在观察坐标系下做空间裁剪，裁剪用到的模型有透视投影和平行投影两种，后者多用于工业设备，论文选用的是透视投影模型。如图 4.5 所示，裁剪范围由远近平面 (Far plane, Near plane) 和视场角 (FOV) 确定。整个裁剪范围称为平截头体。在平截头体内的数据保留，之外的数据剔除不做进一步的渲染。最后是通过  $p$  矩阵完成视口变换将裁剪空间内模型的三维坐标转换到屏幕上要显示的二维坐标  $u_i, v_i$ 。变换过程表示为：

$$\begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \begin{bmatrix} x_{vi} \\ y_{vi} \\ z_{vi} \\ 1 \end{bmatrix} \quad (4.2)$$

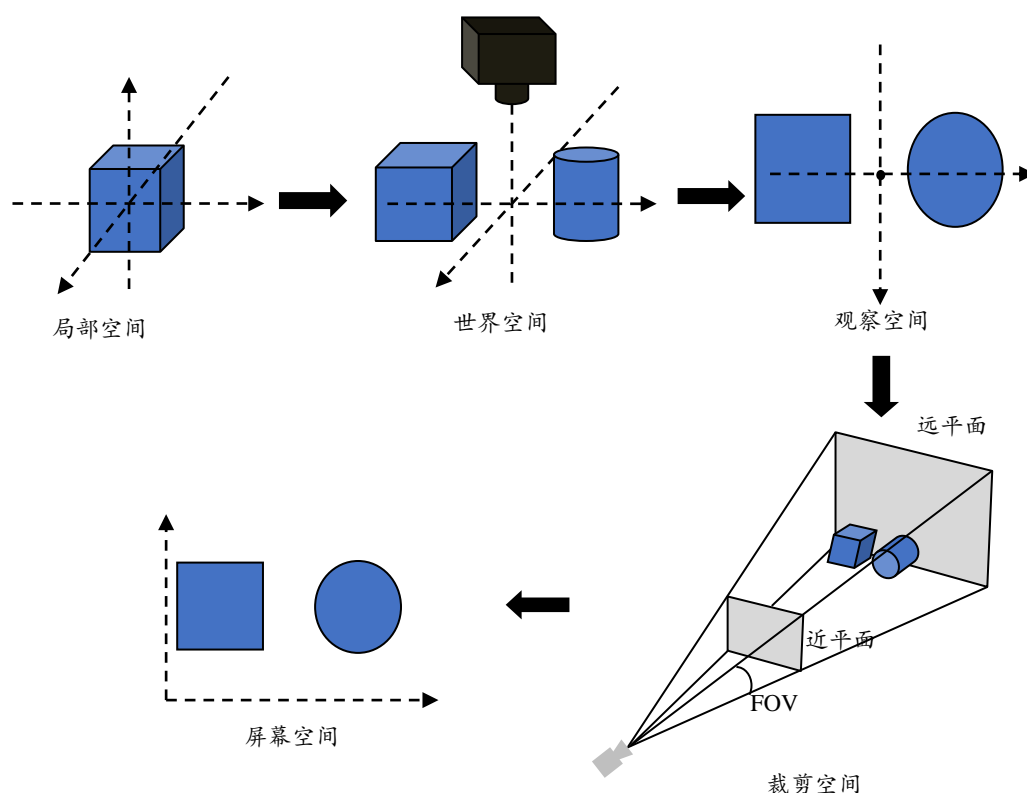


图 4.5 顶点着色器中坐标变换过程

之后是对消隐数据“上色”。顶点着色器处理后的结果作为输出传递到下一个处理流程，形状装配、几何着色器、光栅化的过程无需编程。将流程链接好以后系统自动完成。到达片段着色器后需要用 GLSL 编程来实现不同的片段着色器功能。图 4.6 为片段着色器代码示例。片段着色器可以将环境光、漫反射纹理、高光等预期的渲染效果综合成最终的像素颜色值输出，经最后的测试混合得到显存中待显示的像素数据。

```
#version 330 core           //版本号
out vec4 FragColor;        //输出变量
uniform vec3 lightcolor;    //输入变量
void main()                //主函数
{ FragColor =vec4(lightcolor,1.0); }
```

图 4.6 着色器代码示例

#### 4.1.4 模型加载及数据输出

常见的三维模型格式众多，若针对每一种模型去独立编程解析，是一件繁琐的事。Assimp 库有全面的三维模型加载功能，可以将不同格式下的三维模型解析成通

用的数据类型。用 Assimp 加载三维模型后，与上节所述消隐过程衔接，即可实现各种模型的渲染处理。

渲染好的数据输出是将显存中的数据读取并保存。OpenGL 提供了相应的函数来完成此过程。如图 4.7 所示，通过 `glGetDoublev` 函数获取当前的 **view** 矩阵和 **projection** 矩阵，`glGetIntegerv` 获取 **viewprot** 矩阵。再通过 `glReadPixels` 读取显存中的数据在屏幕坐标系下的坐标、颜色值。最后通过 `gluUnProject` 实现逆视口变换。根据获取的矩阵和二维坐标计算出对应像素的三维坐标并存储在申请好的内存中。最后将内存中的三维坐标和颜色值输出到文档中。至此完成了对三维模型的消隐处理，并将处理后的数据输出供后期全息图的制作。

```
glGetDoublev( GL_MODELVIEW_MATRIX, getmodelview);
glGetDoublev( GL_PROJECTION_MATRIX, getprojection);
glGetIntegerv( GL_VIEWPORT, getviewprot);
glReadPixels(int(wordx),int(wordy),widthnow,heightnow, GL_DEPTH_COMPONENT, GL_FLOAT, depthz );
glReadPixels(int(wordx),int(wordy),widthnow,heightnow, GL_RGB, GL_UNSIGNED_BYTE, image );
```

图 4.7 数据输出代码示例

图 4.8 所示为加载模型后获取三维数据和颜色值的结果。左图为颜色信息，右图为深度信息。图 4.9 是通过改变相机观察位置获取另一视角下的数据。

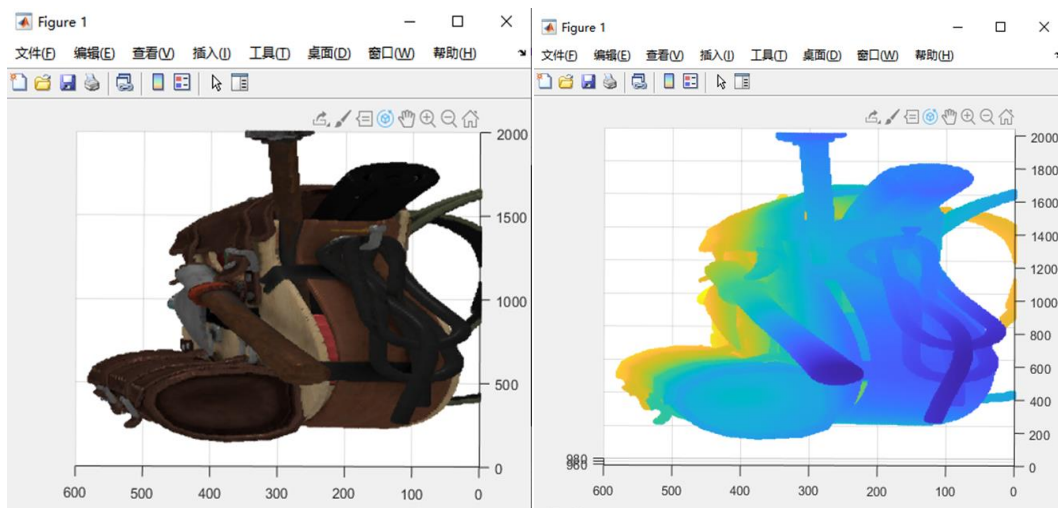


图 4.8 数据恢复结果

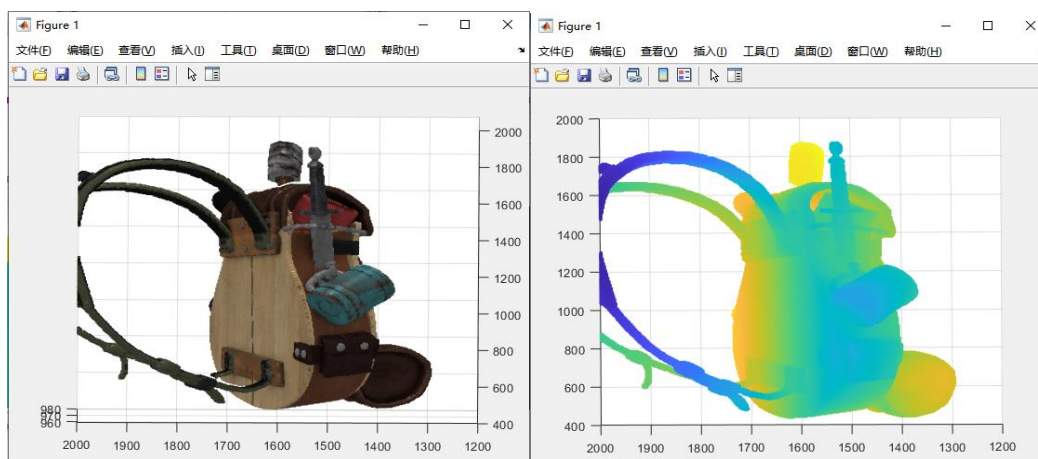


图 4.9 另一视角下的数据恢复结果

## 4.2 全息图多线程并行计算

目前计算机已普遍采用多核 CPU，具备并行计算能力。要充分发挥 CPU 的计算能力，需要采用并行计算算法。多线程和多进程是现代操作系统处理并发的两种基本形式。

从概念上讲，进程 (Process) 是指计算机中已运行的程序，是操作系统进行资源分配和调度的基本单位。作为程序执行的实体，每个进程中可以在内存和 CPU 允许的前提下包含多个线程，它是程序运行的真正实例。多个进程之间可以同步或异步的方式独立运行，以达到充分发挥计算机计算性能的目标。线程指已运行进程的可调度单元，有时也被称为轻量进程。在基于线程的并发过程中，计算任务被分配给多个线程来共同完成，操作系统根据指令完成线程间的调度。由于多个线程共享同一进程的进程空间，可以通过进程中的共享资源来完成线程间的通信。使用共享资源时，需要格外留意线程间数据冲突的问题导致程序无法顺利执行。

### 4.2.1 并行算法

在 4.1 节所述的多视角消隐算法中，各个区域上衍射场的计算过程彼此独立。此处有天然的可并行性。在计算单个区域内衍射场时，计算好区域内积分项的一个周期，后续的积分项延拓和衍射场计算也相互独立，亦可做并行计算。根据上述特点设计并行计算程序，存在数据划分粗略、内存占用大等不足。

在多视角消隐算法中，从分层数据读入到对应衍射场计算完成要经过数据补零、积分项变换、计算积分项外系数和参考光、衍射场计算四个步骤。数据补零耗时极短，无需做并行计算。其余步骤除积分项变换以外，均根据相应的公式逐点计算，所有点的运算彼此相互独立。积分项变换可先一维行变换再一维列变换，各行各列的变换也相互独立。根据以上特点设计并行计算程序可以细化数据划分。具体思路如下：

设单个区域内全息图总点数为  $M \times N$ ，创建了  $i$  个子线程。第 1 到  $(i-1)$  个子线程各自需要计算的点数为  $\text{round}(M \times N / i)$  ( $\text{round}$  表示取整运算)，第  $i$  个子线程计算的点数为  $\text{round}(M \times N / i) + (M \times N) \% i$  ( $\%$  表示取余数)。处理流程如图 4.10 所示。

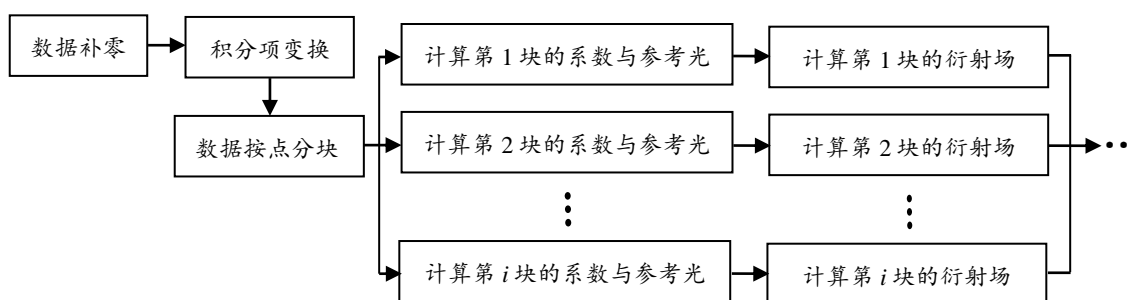


图 4.10 衍射场计算并行处理流程

二维 FFT 实际是先对矩阵的所有行作一维 FFT，再对所有列作一维 FFT。全息图计算过程中待变换的数据可表示成如图 4.11 (a)，其中浅色区域表示有数据，黑色区域表示为补零的部分。若直接用二维的 FFT 做变换，不但行变换时无数据区域要进行变换增加冗余计算，而且无法并行处理。为提高计算速度，可先在有数据的所有行方向上用一维 FFT 得到图 4.11 (b) 所示结果，再对得到的数据在所有列方向上做变换得到图 4.11 (c) 所示结果。行方向上的变换，各行之间相互独立因此可以按行

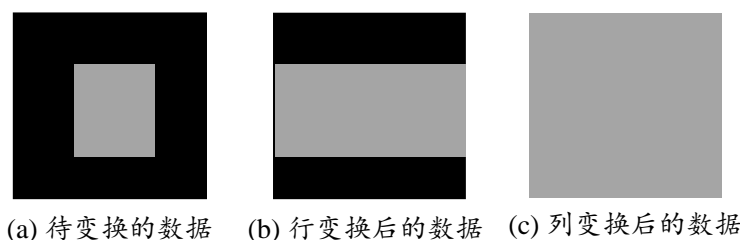


图 4.11 数据变换前后示意图

分配给子线程同时计算从而提高计算速度，列方向上亦是如此。处理流程如图 4.12

所示。

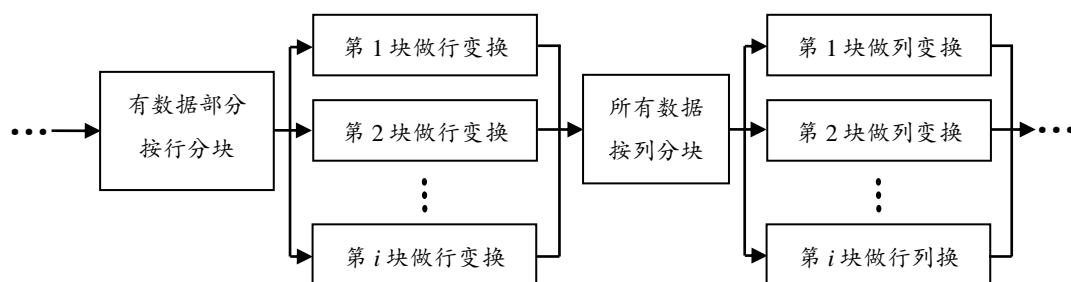


图 4.12 积分项变换并行处理流程

所有层的衍射场计算完成后叠加得到全息面上完整的物光场分布。之后还要经过与参考光干涉、全息图归一化两步得到全息图。此过程的并行处理流程与图 4.10 相似，按点数将数据分块，由子线程完成计算。

综合以上内容可知需要六步来完成相应的计算。分别是：数据补零、积分项变换、积分项外系数与参考光计算、全息面衍射场计算、全息图计算、全息图归一化。

数据补零实现的是根据约束条件式 (3.4) 确定全息面上一个周期积分项的点数  $M$ 、 $N$ ，将分层数据加上随机相位再补零到  $M \times N$ ，后乘上 (4.3) 式中的二次项  $\phi(x_0, y_0)$ ，得到待变换的积分项。

$$\phi(x_0, y_0) = \exp \left[ j \frac{k}{2z} (x_0^2 + y_0^2) \right] \quad (4.3)$$

积分项外系数与参考光计算实现的是计算与层深度对应的积分项外系数行列分量和参考光。衍射场计算则是将算好的积分项周期延拓，乘上由行列分量组成的完整系数得到该层数据在全息面上衍射场的分布。全息图计算是将衍射场与参考光干涉，最后对干涉场归一化，得到最终的全息图。

## 4.2.2 算法实现

综合以上分析，获取消隐数据后计算一个视角下全息图的并行算法程序框图如图 4.13 所示。将三维数据分层后转化为二进制数据读入主程序。参数设置主要完成波长、衍射距离、物面全息面的抽样间隔等基本参量的赋值。

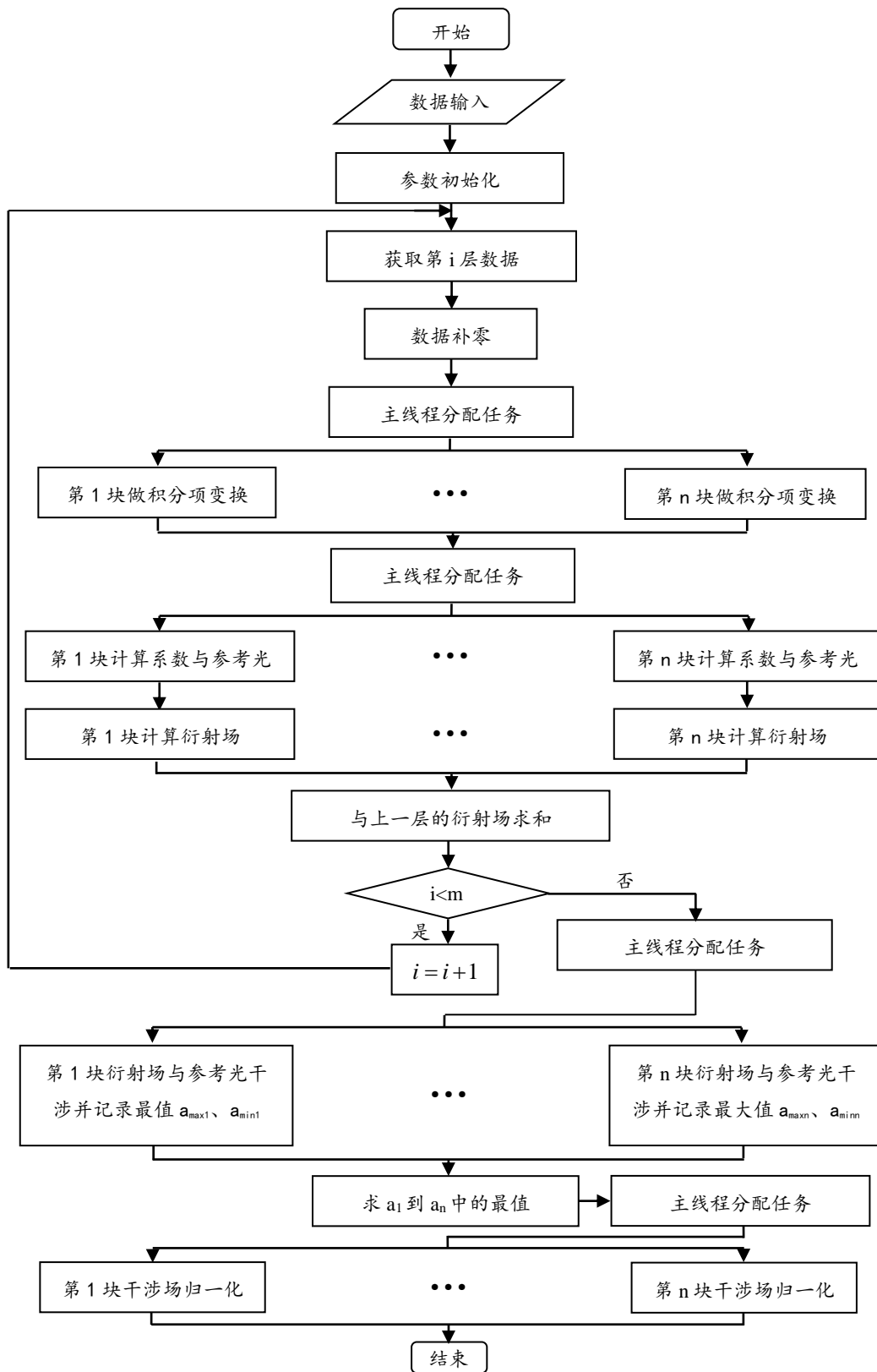


图 4.13 并行计算程序框图

主线程中先要完成全息面上衍射场的计算。首先获取第 1 层数据，根据参数进

行补零。创建  $n$  个子线程，将数据按行(列)数分块给子线程完成积分项变换。得到算好的积分项后，主线程按点数分块，子线程依次完成系数和参考光计算、衍射场计算。得到该层在全息面上的衍射场后与上一层的衍射场求和作为最终结果( $i=1$  时无需求和)。判断当前一层是否为最后一层数据，若不是则继续计算下一层反之则进入全息图计算。子线程运行期间主线程阻塞，待子线程计算结束主线程继续下一步计算。

得到全息面上完整的衍射场后主线程按点数分块，进行物光与参考光的干涉计算，同时记录每块数据下干涉场的最值。回到主线程后比较各块干涉场的最值取其中的最值，用于全息图的归一化。主线程继续按点数分块交给子线程进行归一化处理。结束后存储全息图、销毁子线程、程序结束。

### 4.3 本章小结

本章先分析单视角消隐所带来的问题，从而引出了多视角消隐算法。随后提出了基于 OpenGL 的多视角消隐算法实现思路。介绍了最终所实现的功能，给出了相应的实验结果。另外还提出了基于多核 CPU 的多线程并行计算方法，给出了程序的设计思路，实现流程。



## 5 实验及结果分析

### 5.1 实验系统

在实验中用到的实验系统主要有：全息图计算所使用的计算机环境、全息图输出和光学再现系统。其中计算机 CPU 是 Intel(R) Core(TM) [i7-10700K@3.80GHz](#)，内存为 16GB。周期延拓的全息图快速计算方法、三维模型的消隐算法、多视角消隐的多线程并行计算均用 C++编程实现。全息图输出所用到的是如图 5.1 所示实验室自主研发的全息打印机，打印机分辨率为 318nm。

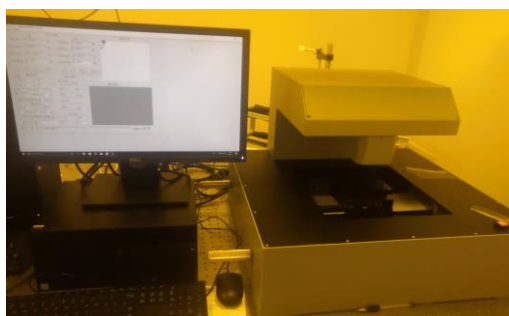


图 5.1 实验室研制的全息打印机

搭建的光学再现系统分两种，图 5.2 是针对输出到光刻胶版上的大尺寸全息图所搭建的再现光路示意图。图 5.3 是用来直接加载全息图的 LCOS 系统示意图，主要用于加载小尺寸全息图，快速验证算法准的确性。LCOS 分辨率为  $4096 \times 2160$ ，像

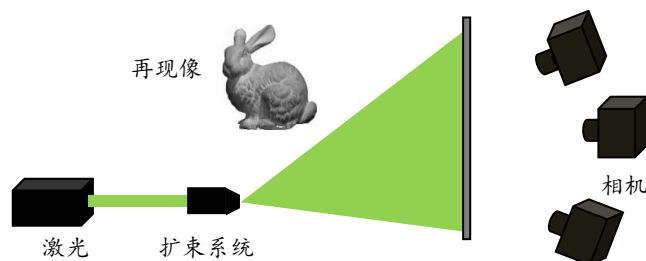


图 5.2 大尺寸全息图再现光路示意图

素间隔为  $4.06\mu\text{m}$ 。激光器发出的激光经扩束得到发散的球面波，经透镜变成汇聚的球面波(此处选用两个透镜为了更方便的调节球面波的汇聚点)。计算机将全息图加

载到 LCOS 后, 反射光经分束棱镜后汇聚在重建像面, 完成三维物体的再现。

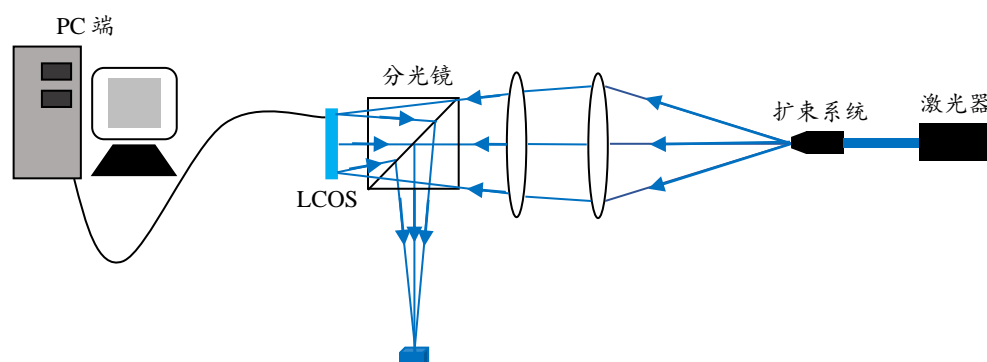


图 5.3 用 LCOS 加载全息图的再现光路示意图

## 5.2 快速算法光学再现结果

为验证提出周期延拓算法的正确性, 设计了程序进行实验。对比了新算法与传统层析法在再现像清晰度、视差效果、深度再现效果的差异; 如图5.4所示, 实验所选用的三维模型点云数据是由斯坦福大学三维点云数据库提供, 模型包含35947个物点。实验中物体在 $x$ 、 $y$ 、 $z$ 三个方向上的实际尺寸分别设置为46.8mm、46.4mm、36.3mm。将点云数据分为40层, 为了更好地观察视差效果, 在第41层处放置了棋盘格。中心层到全息面的距离为310mm。



图 5.4 三维物体模型

为观察采用本算法所制作全息图的再现效果, 计算了像素数为 $54179 \times 399087$ 的全息图。用全息打印机将全息图输出到光刻胶版上。物空间内有效样点数为 $520$ 点 $\times 515$ 点 $\times 41$ 层。物体的抽样间隔为 $0.09\text{mm}$ , 积分项一个周期的抽样点数为 $4927 \times 4927$ 。将打印好的全息图放在如图5.5所示的再现光路中观察再现像, 激光波长为 $473\text{nm}$ , 拍摄实验结果所用相机为华为荣耀20手机所搭载的相机。激光器发出的激光通过扩束和滤波后得到球面波照射到全息图上。

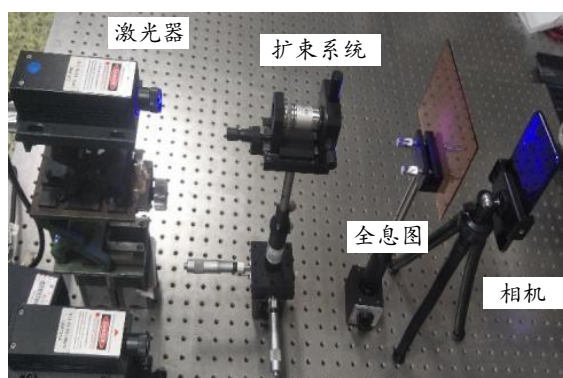


图 5.5 再现光路

图5.6(a)为全息图的局部，图5.6(b)和(c)分别为传统算法与快速算法的再现效果，未观察到二者的差异。图5.7为相机在三个不同视角下拍摄到的再现像。红色框标示区域中，物体与棋盘格之间的遮挡关系正确。当多角度连续拍摄或人眼直接平移观察时可以看到连续的视差变化。

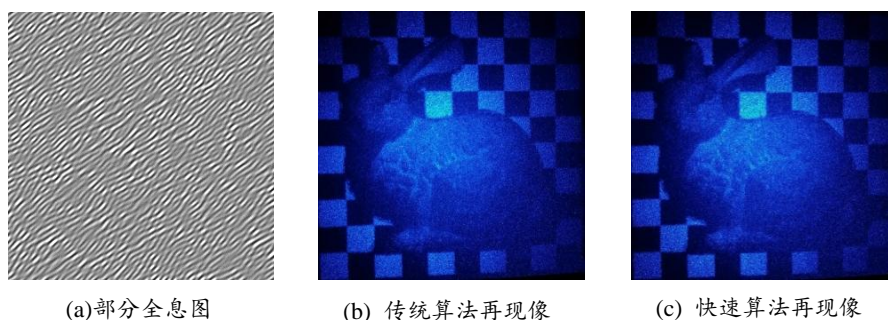


图5.6 部分全息图及两种算法的再现结果

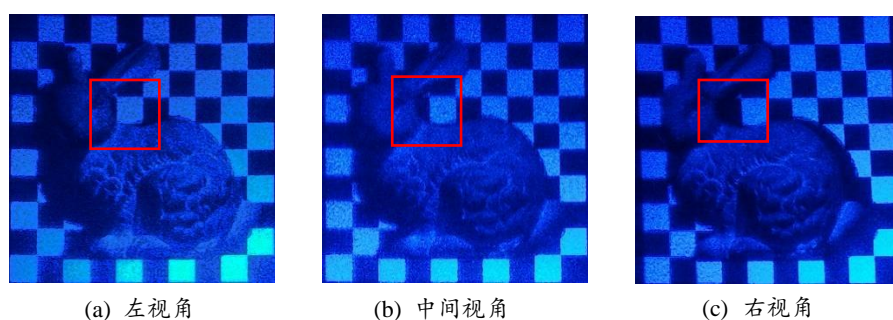


图5.7 不同视角下的再现像

观察不同深度的再现像时，在图5.5光路基础上在全息图右侧加入焦距为30cm的透镜，将再现像成像为等大的实像并用毛玻璃承接。经过测量得知加入透镜后整个成像系统在实像位置附近的景深约为14mm，因此选择间隔大于景深，且特征较为明显的两个深度进行拍摄。图5.8为两个不同深度处的再现像。图中深度为换算成虚像

的深度，再现的深度信息准确。

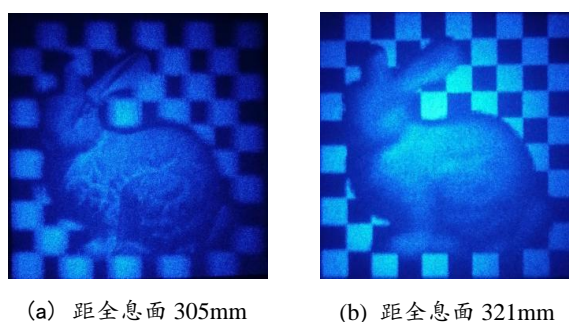


图 5.8 不同深度再现像对比

用层析法计算三维物体的全息图时，再现像的质量与数据分层间隔大小密切相关。层间隔满足人眼分辨率的情况下所观察到的再现像在一定视角范围内都是连续的，当不满足人眼分辨率时，大角度倾斜观察容易出现再现像的分层。在其他参数不变的前提下，将三维物体拉近到距离全息面100mm附近计算、制作全息图，在右边大角度倾斜观察得到如图5.9所示结果。由于观察距离变小，此位置上人眼分辨率更高，再现像在大视角下观察出现了明显的分层。要避免这一现象的出现需要根据人眼分辨率、物体的三维形貌、观察视角等选择合理的层间隔。

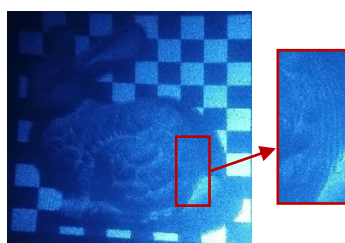


图5.9 再现像分层现象

### 5.3 快速算法速度分析

为研究快速算法的计算速度提高情况，选用上述三维数据模型在同一PC机上用三种算法计算其菲涅尔全息图，取50次运行时间的平均值。在全息面抽样间隔318nm时，积分项一个周期的抽样点数为 $4927 \times 4927$ ，当周期数在3以上后传统算法计算过程中会由于计算机物理内存不足而需要读取硬盘，计算速度会明显降低，此时的计算速度不具可比性。因此采用较小的积分项周期进行计算，全息面的抽样间隔为 $4.06\mu\text{m}$ （实验室中LCOS的像素尺寸），物体距离全息面900mm。根据3.1节所述

求出全息面上积分项一个周期内包含的像素数为 $1174 \times 1174$ 。计算了九组周期数从 $k_x=2, k_y=2$ 到 $k_x=10, k_y=10$ 的全息图。每一组分别采用快速算法周期延拓加系数行列分解(方法1)、快速算法仅积分项周期延拓(方法2)、及传统算法(方法3)三种方法计算,得到如表1所示的结果。图5.10为不同算法计算时间与积分项周期数关系曲线。由结果可见系数行列分解后的快速算法速度最快,仅周期延拓的快速计算次之,且均快于传统算法。整体看所计算的全息图越大时快速算法的速度优势越明显,当全息图分辨率达到 $1174 \times 1174$ 像素时计算速度达到传统算法的13倍。

表1 不同算法计算时间对比

全息图分辨率(像素)	积分项周期数 $k_x \times k_y$	计算用时/s		
		方法 1	方法 2	方法 3
2348×2348	2×2	5.4	6.1	12.1
3522×3522	3×3	9.2	11.1	27.3
4696×4696	4×4	14.5	18.8	48.7
5870×5870	5×5	21.2	27.6	75.1
7044×7044	6×6	29.6	37.9	110.6
8218×8218	7×7	39.5	50.7	164.9
9392×9392	8×8	50.7	65.7	293.2
10566×10566	9×9	62.3	82.1	573.2
11740×11740	10×10	77.4	101.0	1021.0

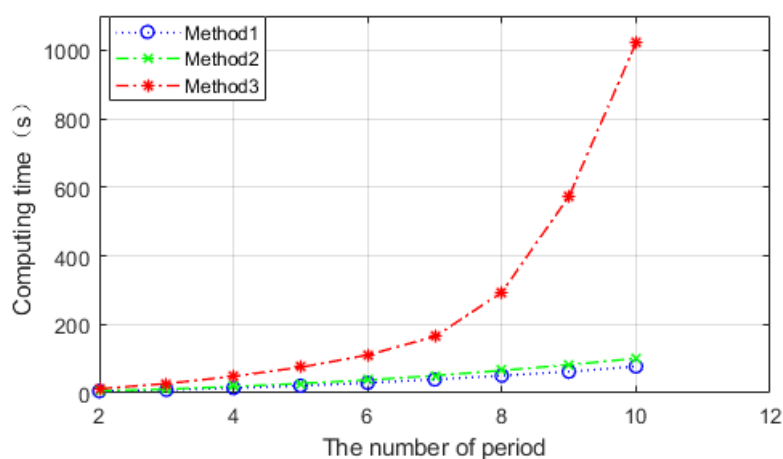


图 5.10 计算时间与周期数关系

另外需要说明的是在表1传统算法用时的数据中,全息图像素数增加到 $9392 \times 9392$ 后计算用时会发生激增。这是由于传统层析法计算全息图时每一层数据都要插值或补零到与全息图相同的像素。计算量随着全息图尺寸的增加而呈平方式的增长,从而导致计算用时激增。相比于传统算法,快速算法在计算完成每一层在全



息面上一个周期的积分项后，即可通过周期延拓来得到整个全息面的积分项。计算量随着全息图尺寸增大而平缓增长。

为探究数据层数对速度提高倍数的影响，进行了层数从 10 到 50，以 5 层为增量的全息图计算速度提高倍数对比实验。每组实验用两种算法计算周期数从  $k_x=2, k_y=2$  到  $k_x=8, k_y=8$  的不同尺寸全息图，并得到快速算法的提速倍数。得到了分层数对快速算法的速度提高倍数无影响的结论。表 2 给出了其中 5 组数据。

表2 层数与速度提高倍数关系

周期数 $k_x \times k_y$	50 层	40 层	30 层	20 层	10 层
2×2	2.3	2.2	2.2	2.2	2.3
3×3	2.9	3.0	2.9	3.0	2.9
4×4	3.3	3.4	3.2	3.5	3.2
5×5	3.6	3.5	3.4	3.7	3.7
6×6	3.8	3.7	3.9	3.9	4.0
7×7	4.1	4.1	4.1	4.3	4.3
8×8	5.8	5.8	5.9	5.9	5.7

#### 5.4 消隐与并行加速结果分析

为探究采用 4.1 节所述的消隐算法能否改善显示效果，设计了如下实验。将图 5.3 所示的三维模型在左中右三个视角下做消隐处理。用中间视角的消隐数据计算左右视角下的全息图作为对照，另分别用左右视角下的消隐数据计算对应视角下的全息图。将制作好的全息图加载到 LCOS 上进行光学再现。结果如图 5.11 所示，图 5.11 (a) 和图 5.11 (b) 的左边为单视角消隐的结果，右边为多视角消隐后的结果。各

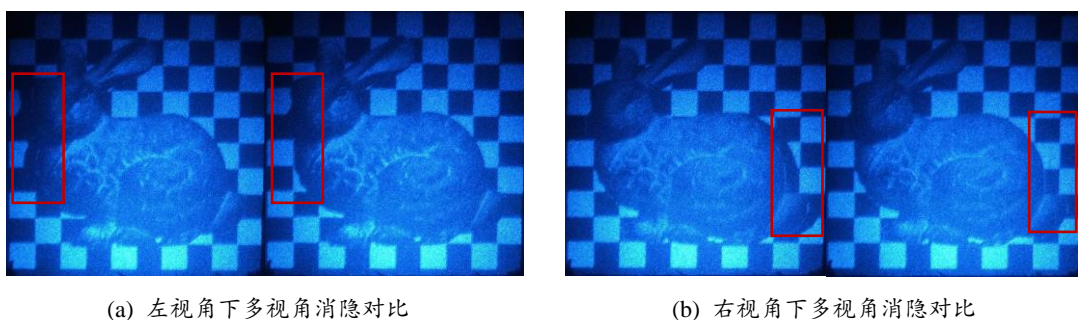


图5.11 多视角消隐前后对比

视角下由红色框标示区域对比可看出，多视角消隐处理后的遮挡关系更接近观察实

物。证明消隐处理有效，可以提高全息三维物体显示的效果。

为探究 4.2 节所采用的多线程并行计算方法的加速效果，设计实验对比了不同线程数下计算速度的结果。首先验证积分项变换部分是否能够达到预期的提速效果。选用大小为  $7500 \times 7500$  的矩阵，有效数据在矩阵中央大小为  $1000 \times 1000$ 。首先用 `fftw_plan_dft_2d` 直接对矩阵做二维傅里叶变换记录其 50 次用时的平均值作为对照组数据。实验组数据先在有数据的部分按行做变换，再在整个矩阵上做列方向上的变换，变换过程用 `fftw_plan_dft_1d` 完成。各方向上的变换依次采用不同数量的线程完成计算，统计每组数据的 50 次平均值作为最终结果。数据如表 3 所示。

表 3 积分项变换并行前后用时对比

对照组数据	线程数						
	1	2	3	4	5	6	7
0.715s	0.453s	0.231s	0.164s	0.139s	0.115s	0.101s	0.092s

由数据可见由于在行方向上减少了无数据部分的计算，单线程的计算速度比直接作二维变换快，随着线程数的增加计算速度得到了进一步提升。证明此改进方案在层析法计算菲涅尔全息图的过程中有效。

之后验证 4.2 节所提并行算法的整体提速效果。选用图 5.4 所述相同的三维数据在同一 PC 机上用不同线程数来计算其菲涅尔全息图，取 50 次运行时间的平均值。全息面的抽样间隔为  $4.06\mu\text{m}$ ，物体距离全息面 900mm，全息面上积分项一个周期内包含的像素数为  $1174 \times 1174$ 。计算了线程数从 1 到 7 变化的七组数据，其中每组计算了 8 种不同尺寸的全息图。实验结果如表 4 所示。将表中数据绘制成折线图 5.12，从图中数据可见计算用时与线程数成反比关系，符合理论推导。

表 4 不同线程数计算时间对比

全息图分辨率 (像素)	线程数						
	1	2	3	4	5	6	7
2348×2348	4.5	2.4	1.7	1.3	1.1	1.0	0.9
3522×3522	6.9	3.6	2.5	1.9	1.6	1.4	1.3
4696×4696	10.2	5.2	3.7	2.8	2.3	2.0	2.0
5870×5870	14.4	7.4	5.0	3.9	3.2	2.9	2.7
7044×7044	19.6	9.9	6.8	5.2	4.3	3.6	3.5
8218×8218	25.7	13.0	8.8	6.7	5.5	4.8	4.5
9392×9392	32.7	16.5	11.2	8.5	7.0	5.9	5.8
10566×10566	40.7	20.6	13.8	10.5	8.7	7.5	7.4

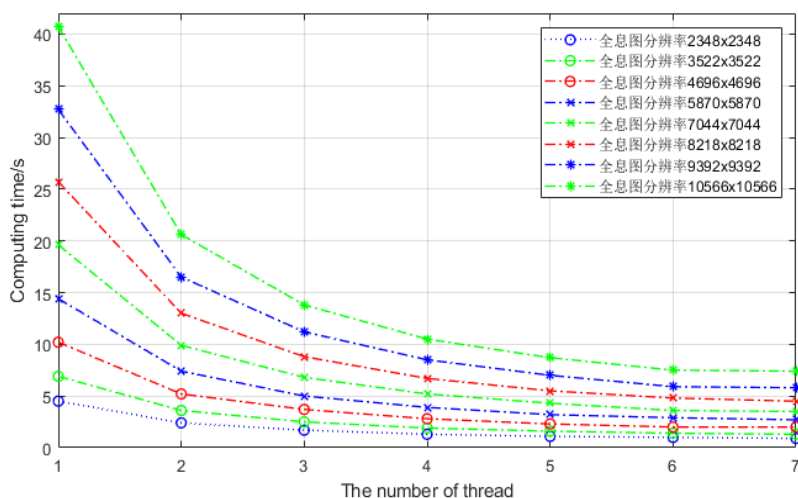


图 5.12 计算时间与线程数关系

从理论上讲计算量不变线程数为  $m$  时计算用时应当是单线程的  $1/m$ 。从实验结果看在线程为 2 时, 8 组数据的用时都在单线程状态下的 52% (理论值 50%) 左右, 接近理论值。线程数为 4 时, 用时是单线程的 28% (理论值 25%) 左右。线程数为 7 时, 用时是单线程的 18% (理论值 14%) 左右。由此可见随线程数的增加提速效果较理论值逐渐变差。这与程序的执行效率、设计思路、数据量等均有联系。可通过进一步改进程序设计思路来提高效率达到更好的提速效果。

## 5.5 本章小结

本章介绍了实验所用到的系统, 给出了相关算法的实验结果, 并进行了分析。由实验结果可知, 提出的算法在保证良好再现效果的同时显著将层析法计算机制菲涅尔全息图的计算速度提高了13倍。在大尺寸全息图的计算上有着良好的表现。多视角消隐处理后进一步提高了三维显示效果。采用多线程并行计算进一步提高了全息图的计算速度。计算相同大小的全息图, 7线程的情况下可以提高计算速度5倍以上。程序设计仍有提升空间, 以达到更高的提速效果。



## 6 总结与展望

### 6.1 总结

论文阐述了全息术的基本原理及关键技术。重点分析了计算全息快速算法及存在的一些问题，由此引出研究的主要内容：基于层析法的计算机制菲涅尔全息图快速计算。从原理的角度分析了菲涅尔衍射积分公式中积分项的周期性，以及系数项可行列分解的特性，依据这些特性提出了一种新的快速计算方法。用 C++ 编程实现了该算法。通过实验验证了该算法，对比了传统层析法与所提算法的计算速度。实验结果表明快速算法在速度上明显优于传统的层析法，当其他参数不变要计算全息图的尺寸越大，快速算法的速度提高越明显。

此外为进一步提高全息图的三维再现效果和计算速度，首先用 C++ 编程，通过 OpenGL 实现了对三维模的场景渲染、多视角消隐处理、数据输出等预处理。其次采用所提出的新算法，用 C++ 编程实现了多线程并行计算。实验对比了不同线程数下快速算法的计算速度，实验表明 7 线程的情况下可以提高计算速度 5 倍以上。同时也发现线程数越多则提速效果有所下降，与理论值相差越大。

### 6.2 展望

随着社会发展二维显示已无法充分满足人们工作、生活、娱乐的要求。高新技术产业、医疗、军事工业等对高质量的真三维显示有着迫切的需求。同时 5G 技术、元宇宙等新技术新领域的出现，给真三维显示带来了更多的市场动力。全息术作为一种理想的三维显示技术，必将受到更加广泛关注。论文提出的计算全息图快速算法，在提高全息图的计算速度、实现大场景多视角动态三维显示方面有一定的应用前景。今后将在以下几个方面进一步研究、改进：

1. 多线程程序执行效率有待进一步提高，以实现更充分的利用计算机 CPU；
2. 与近眼显示技术相结合，实现高帧率的动态显示；

3. 对程序做进一步整合，形成一套完整的计算全息数据处理、全息图生成软件；
4. 设计一个可供多人观看的计算全息三维显示系统。

## 参考文献

- [1] JASON G. Three-dimensional display technologies[J]. *Advances in Optics and Photonics*, 2013, 5(4): 456-535.
- [2] Sorensen S E B, Hansen P S, Sorensen N L. Method for recording and viewing stereoscopic images in color using multichrome filters: U.S. patent 6,687,003[P]. 2004-02-03.
- [3] Edirisinghe E A, Jiang J, Stereo imaging, an emerging technology. *Proceedings of SSGRR, L'Aquila*, July 31-August 6, 2000.
- [4] COLTHEART M. The persistences of vision[J]. *Phil. Trans. R. Soc.*, 1980, B 290: 57-69.
- [5] CHENG Dewen, WANG Yongtian, HUA Hong, et al. Design of an optical see-through head-mounted display with a low f-number and large field of view using a freeform prism[J]. *Applied Optics*, 2009, 48(14): 2655-2668.
- [6] PETERKA T, KOOIMAMA R L, SANDIN D J, et al. Advances in the Dynallax Solid-State Dynamic Parallax Barrier Autostereoscopic Visualization Display System[J]. *IEEE Transactions on Visualization and Computer Graphics*, 2008, 14: 487-499.
- [7] PARK J H, KIM Y, KIM J, et al. Three-dimensional display scheme based on integral imaging with three-dimensional information processing[J]. *Optics Express*, 2004, 12(24): 6020-6032.
- [8] WOODGATE G J, EZRA D, HARROLD J, et al. Observer tracking autostereoscopic 3D display systems[J]. *Proc. SPIE*, 1997, 187: 3012.
- [9] COSSAIRT O S, NAPOLI J, HILL S L, et al. Occlusion-capable multiview volumetric three-dimensional display[J]. *Applied Optics*. 2007, 46(8): 1244-1250.
- [10] TAO Yuhong, WANG Qionghua, Gu Jun, et al. Autostereoscopic three-dimensional projector based on two parallax barriers[J]. *Optics Letters*, 2009, 34(20): 3220-3222.
- [11] KAJIKI Y, YOSHIKAWA H, HONDA T. Hologramlike video images by 45-view stereoscopic display[J]. *Proc. SPIE* 3012, 154 (1997).
- [12] SMALLEY D, NYGAARD E, SQUIRE K, et al. A photophoretic-trap volumetric display[J]. *Nature*, 2018, 553: 486-490.
- [13] GABOR D. A new microscopic principle[J]. *Nature*, 1948, 161(4098): 777-778.
- [14] YARAS F, KANG H, ONURAL L. State of the art in holographic displays: a survey[J]. *Journal of Display Technology*, 2010, 6(10): 443-454.
- [15] LI Xin, LIU Juan, ZHAO Tao, et al. Color dynamic holographic display with wide viewing angle by improved complex amplitude modulation[J]. *Optics Express*, 2018, 26(3): 2349-2358.
- [16] SHI L, LI B, KIM C, et al. Towards real-time photorealistic 3D holography with deep neural networks[J]. *Nature*, 2021, 591(7849): 234-239.
- [17] WU Jiachen, LIU Kexuan, SUI Xiaomeng, et al. High-speed computer-generated holography using autoencoder-based deep neural network[J]. *Optics Letters*, 2021,

- 46(12): 2908-2911.
- [18]许富洋,杨鑫,等. 高分辨率多视点动态全息3D显示[J]. 中国激光, 2021,48(1):158-165.
- [19]李勇,卢世江,宋芳,等. 真实场景的三维视频采集及显示[J]. 光子学报, 2011, 40(11):1706-1710.
- [20]HE Zehao, SUI XiaoMeng, JIN GuoFan, et al. Progress in virtual reality and augmented reality based on holographic display[J]. Applied Optics, 2019, 58(5): A74-A81.
- [21]Chenliang Chang, Kiseung Bang, Gordon Wetzstein, et al. Toward the next-generation VR/AR optics: a review of holographic near-eye displays from a human-centric perspective[J]. Optica, 2020, 7(11): 1563-1577.
- [22]KOZMA A, KELLY D L. Spatial filtering for detection of signals submerged in noise[J]. Applied Optics, 1965, 4(4): 387-392.
- [23]BROWN B R, LOHMANN A W. Computer-generated binary holograms[J]. IBM Journal of Research and Development, 1969, 13(2): 160-168.
- [24]ANMA Y, MATSUSHIMA K, Nakahara S. Hybrid CGH by Digitized Holography : CGH for Mixed 3D Scene of Virtual and Real Object[M]. 2011.
- [25]KYOJI M, NORIAKI S. Full-color digitized holography for large-scale holographic 3D imaging of physical and nonphysical objects[J]. Applied Optics, 2018, 57(1): A150-A156.
- [26]胡杰康,王辉,李勇,等.基于数字全息和计算全息的动态三色全息三维显示[J].浙江师范大学学报:自然科学版,2019(2):135-142.
- [27]MELANIA P, PASQUALE M, ANDREA Finizio. Synthesis and display of dynamic holographic 3D scenes with real-world objects[J]. Optics Express, 2010, 18(9): 8806-8815.
- [28]WERONIKA Z, TOMASZ K, MALGORZATA K. Multi-SLM Color Holographic 3D display based on RGB spatial filter[J]. Journal Of Display Technology, 2016, 12(12): 1724-1731.
- [29]LI Jin, SMITHWICK Quinn, CHU Daping. Full bandwidth dynamic coarse integral holographic displays with large field of view using a large resonant scanner and a galvanometer scanner [J]. Optics Express, 2018, 26(13): 17459-17476.
- [30]PI Dapu, LIU Juan, YU Shuang. Speckleless color dynamic three-dimensional holographic display based on complex amplitude modulation[J]. Applied Optics, 2021, 60(25): 7844-7848.
- [31]CORDA R, GIUSTO D, LIOTTA A, et al. Recent Advances in the Processing and Rendering Algorithms for Computer-Generated Holography[J]. Electronics, 2019, 8(5): 556-573.
- [32]LUCENTE M E. Interactive computation of holograms using a look-up table[J]. Journal of Electronic Imaging, 1993. 2(1): 28-34.
- [33]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.
- [34]PAN Y, XU X, SOLANKI S, et al. Fast CGH computation using S-LUT on GPU[J]. Optics Express, 2009, 17(21): 18543-18555.

- [35]JIA Jia, WANG Yongtian, LIU Juan, 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.
- [36]KIM S C, KIM E S, Fast computation of hologram patterns of a 3D object using run-length encoding and novel look-up table methods[J]. Applied Optics, 2009, 48(6): 1030-1041.
- [37]YOSHIKAWA H. Fast computation of Fresnel holograms employing difference[J]. Optical Review, 2001, 8(5): 331-335.
- [38]李勇, 许富洋, 金红震, 等. 一种菲涅尔全息图的快速算法[J]. 光子学报, 2010, 39(3): 529-532.
- [39]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.
- [40]ARAI D, SHIMOBABA T, MURANO K, et al. Acceleration of computer-generated holograms using tilted wavefront recording plane method[J]. Optics Express, 2015, 23(2): 1740-1747.
- [41]WENG J, SHIMOBABA T, OKADA N, et al. Generation of real-time large computer generated hologram using wavefront recording method[J]. Optics Express, 2012, 20(4): 4018-4023.
- [42]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.
- [43]PAN Yijie, WANG Yongtian, LIU Juan, et al. Fast polygon-based method for calculating computer-generated holograms in three-dimensional display[J]. Applied Optics. 2013, 52(1): A290-A299.
- [44]MATSUSHIMA K. Computer-generated holograms for three-dimensional surface objects with shade and texture[J]. Applied Optics. 2005, 44(22): 4607-4614.
- [45]MATSUSHIMA K, NAKAMURA M, NAKAHARA S. Silhouette method for hidden surface removal in computer holography and its acceleration using the switch-back technique[J]. Optics Express, 2014, 22(20): 24450-24465.
- [46]AHRENBERG L, BENZIE P, MAGNOR M, et al. Computer generated holograms from three dimensional meshes using an analytic light transport model[J]. Applied Optics, 2008, 47(10): 1567-1574.
- [47]PAN Yijie, WANG Yongtian, LIU Juan, et al. Improved full analytical polygonbased method using Fourier analysis of the three-dimensional affine transformation[J]. Applied Optics, 2014, 53(7): 1354-1362.
- [48]ZHAO Yan, CAO Liangcai, ZHANG Hao, et al. Accurate calculation of computer-generated holograms using angular-spectrum layer-oriented method[J]. Optics Express, 2015, 23(20): 25440-25449.
- [49]JIA Jia, SI Jhen, CHU Daping. Fast two-step layer-based method for computer generated hologram using sub-sparse 2D fast Fourier transform[J]. Optics Express, 2018, 26(13): 17487-17497.
- [50]OKADA N, SHIMOBABA T, ICHIHASHI Y, et al. Band-limited double-step Fresnel diffraction and its application to computer-generated holograms[J]. Optics Express, 2015, 21(7): 9192-9197.
- [51]CHEN J S, CHU Daping. Improved layer-based method for rapid hologram

- generation and real-time interactive holographic display applications[J]. *Optics Express*, 2015, 23(14): 18143-18155.
- [52]曹良才,何泽浩,赵燕,等.面向层结构的角谱传播计算全息算法[J].*红外与激光工程*,2018,47(6):11-19.
- [53]KOKI W, MASAHIRO Y, BAHRAM J. High-resolution three-dimensional holographic display using dense ray sampling from integral imaging[J]. *Optics Letters*, 2012, 37(24): 5103-5105.
- [54]LI Zhaohui, ZHANG Jianqi, WANG Xiaorui, et al. High resolution integral holography using Fourier ptychographic approach[J]. *Optics Express*, 2014, 22(26): 31935-31947.
- [55]CHEN Ni, REN Zhenbo, Edmund Y. Lam. High-resolution Fourier hologram synthesis from photographic images through computing the light field[J]. *Applied Optics*, 2016, 55(7): 1751-1756.
- [56]JAEHONG L, HOMIN K, HAN-JU Y. Out-of-core GPU 2D-shift-FFT algorithm for ultra-high-resolution hologram generation[J]. *Optics Express*, 2021, 29(12): 19094-19112.
- [57]李勇,王辉,应朝福,等.基于高速局域网的计算机制全息并行计算算法及实现[J].*激光与红外*.2002,32(5):333-335.
- [58]JESSIE J B, SHINPEI W, KANEMITSU O, et al. Decomposition method for fast computation of gigapixel-sized Fresnel holograms on a graphics processing unit cluster[J]. *Applied Optics*, 2018, 57(12): 3134.
- [59]YANG Xing, WANG Hui, LI Yong, et al. Large scale and high resolution computer generated synthetic color rainbow hologram[J]. *Journal of Optics*, 2019, 21(9): 025601-025611.
- [60]YANG Xing, XU Fuyang, ZHANG Heng, et al. High-resolution hologram calculation method based on light field image rendering[J]. *Applied Sciences*, 2020, 10(3): 819-828.
- [61]PARK J H. Efficient calculation scheme for high pixel resolution non-hogel-based computer generated hologram from light field[J]. *Optics Express*, 2020, 28(5): 6663-6683.
- [62]金国藩.计算机制全息图[M].北京:科学出版社,2020:2-6.
- [63]苏显渝.信息光学[M].北京:科学出版社,2011:158-162.
- [64]Robert M, Gray J W G. *Fourier Transforms: An Introduction for Engineers*[M]. New York: Springer, 1995.
- [65]Goodman J W. *Introduction to Fourier Optics*[M]. New York:McGraw-Hill,1996.
- [66]CHEN Jhensi, CHU Daping, Quinn Y. Smithwick. Rapid hologram generation utilizing layer-based approach and graphic rendering for realistic three-dimensional image reconstruction by angular tiling, *Journal of Electronic Imaging*[J], 2014, 23(2): 023016.
- [67]CHEN Jhensi, CHU Daping. Improved layer-based method for rapid hologram generation and real-time interactive holographic display applications[J]. *Optics Express*, 2015, 23(14): 18143-18155.

---

## 攻读学位期间取得的成果

- [1] 姚建云,应晓霖,许富洋,吴琼,张海花,李勇. 积分项周期延拓快速计算大尺寸菲涅尔全息图[J]. 光子学报, 2022,51(1):356-366.
- [2] 应晓霖,姚建云,张晓松,楚冬娅,李勇. 采用LD的光源步进条纹投影三维测量系统[J].光电工程,2022,48(11):43-51.
- [3] 刘子陌,姚建云,许富洋,杨鑫,宋强,马国斌,李勇. 计算机集成彩色彩虹全息三维显示[J]. 光学学报, 2021, 41(10):96-102.
- [4] 许富洋,杨鑫,姚建云,刘子陌,宋强,李勇. 高分辨率多视点动态全息3D显示. 中国激光, 2021, 48(01):158-165.

---

## 致 谢

论文撰写已接近尾声，在此首先要感谢我的导师李勇教授，他学识渊博在科研的路上给了我极大的帮助。从研一入学就被老师严谨求实的治学态度和一丝不苟的工作理念深深吸引。学术上李老师总能在我们困惑不解时给我们指明方向，在他精心的培养和指导下学到了许多，解决问题的能力也有了很大的提高。毕业论文完成的过程中老师总能及时的指出问题和改进方向，使我少走了许多弯路。回想起老师研一时俯身在实验台前为我们示范光路调校技巧依旧记忆犹新，老师陪我熬过的数不清的夜晚是他对学生浓浓的关爱。老师的心血和付出渗透在论文中的字里行间。能成为李老师的学生我非常荣幸，他的品格和工作态度值得我终身学习。

同时要感谢实验室吴琮老师在实验设备上给我的帮助。也要感谢金伟民和马丽红老师在光学全息和光路搭建方面给予的帮助，感谢施逸乐老师和许富洋老师在实验室使用方面的帮助，感谢杨鑫师兄和张海花师姐在生活和科研上给我的关怀。

感谢实验室的兄弟姐妹：张广汇、宋仁杰、应晓霖、孙雪茹、刘子陌、张晓松、楚冬娅、彭先君、胡涛，有你们的陪伴使我的研究生生活增添了色彩。大家一起讨论问题，共同成长给了我许多启发。

还要感谢父母三年的支持，你们默默的付出才使我能全身心的投入学习，顺利完成学业。你们养我成人，我陪你们到老。

从一封邮件到一纸证书，三年时光惜之不及。是巧合也罢是注定也好，这三年很充实，很快乐。师大里恬静淡雅的初阳湖面、古香古色的红楼、生机勃勃的草地树林、繁华热闹的北门、书香四溢的图文到处都充满着回忆。景难舍情更浓，祝愿师大教育事业蒸蒸日上。



## 浙江师范大学学位论文诚信承诺书

我承诺自觉遵守《浙江师范大学研究生学术道德规范管理条例》。我的学位论文中凡引用他人已经发表或未发表的成果、数据、观点等，均已明确注明并详细列出有关文献的名称、作者、年份、刊物名称和出版文献的出版机构、出版地和版次等内容。论文中未注明的内容为本人的研究成果。

如有违反，本人接受处罚并承担一切责任。

承诺人（研究生）：姚建云

指导教师：李勇

## 浙江师范大学学位论文独创性声明

本人声明所呈交的学位论文是我个人在导师指导下进行的研究工作及取得的研究成果。论文中除了特别加以标注和致谢的地方外，不包含其他人或其他机构已经发表或撰写过的研究成果。其他同志对本研究的启发和所做的贡献均已在论文中作了明确的声明并表示了谢意。本人完全意识到本声明的法律结果由本人承担。

作者签名: 姚建云 日期: 2022年5月31日

## 学位论文使用授权声明

本人完全了解浙江师范大学有关保留、使用学位论文的规定，即：学校有权保留并向国家有关机关或机构送交论文的复印件和电子文档，允许论文被查阅和借阅，可以采用影印、缩印或扫描等手段保存、汇编学位论文。同意浙江师范大学可以用不同方式在不同媒体上发表、传播论文的全部或部分内容。

保密的学位论文在解密后遵守此协议。

作者签名：姚建云 导师签名：李勇 日期：2022年5月31日