

# 基于布朗粒子运动的微小位移测量实验设计与验证

2025 年 8 月 18 日

## 摘要

热力学基本规律在宏观体系中已得到广泛验证，但其在微观尺度下的适用性仍需深入探讨。该实验从微观粒子的热运动出发，以布朗运动为典型模型，结合实验与理论分析，通过测量布朗粒子的随机位移行为，验证不可逆过程中的熵产生机制。进一步地，我们建立了布朗运动中扩散系数与熵产生之间的热力学不确定关系，揭示了微观热运动背后的统计规律及其与热力学第二定律的一致性。该实验研究为从微观尺度验证热力学基本规律提供了有效途径，并对理解非平衡态下微观体系的热力学行为具有重要意义。

为了研究布朗粒子的运动，我们设计并搭建了基于离轴全息术的数字全息显微系统，计划实现对微米级粒子三维位置的高精度测量。在确定物体的位置时，我们优化了确定 z 轴位置的算法，重建出聚焦效果最好的全息图并确定位置 z，通过实验验证了该方法的有效性。同时还使用外加电场的方式为布朗运动添加了随机噪声。通过测量不同噪声强度下粒子的扩散系数，验证了噪声强度与扩散系数之间的修正关系。

**关键字：** 布朗运动，熵产生率，热力学不确定关系，离轴全息

## **Abstract**

# 目录

<b>1 实验背景介绍</b>	<b>4</b>
1.1 布朗运动 . . . . .	4
1.2 熵产生 . . . . .	4
1.3 全息术 . . . . .	4
<b>2 实验原理</b>	<b>6</b>
2.1 有效动能温度公式推导 . . . . .	6
2.2 噪声强度与扩散系数的修正关系 . . . . .	7
2.3 同轴全息的原理与重建 . . . . .	8
2.4 离轴全息的原理与重建 . . . . .	9
<b>3 理论模拟</b>	<b>10</b>
<b>4 装置设计</b>	<b>11</b>
4.1 实验仪器 . . . . .	11
4.2 光路图 . . . . .	11
<b>5 实验内容</b>	<b>12</b>
5.1 离轴全息确定物体位置 . . . . .	12
<b>6 误差分析</b>	<b>15</b>
<b>7 结论与展望</b>	<b>16</b>
<b>8 参考文献</b>	<b>17</b>
<b>9 附录</b>	<b>19</b>
9.1 人员分工 . . . . .	19
9.2 部分代码展示 . . . . .	19
9.3 部分全息图与拍摄图展示 . . . . .	21

# Chapter 1

## 实验背景介绍

### 1.1 布朗运动

布朗运动是指悬浮在流体中的微小粒子由于流体分子热运动而产生的随机运动现象。该现象由植物学家罗伯特·布朗于 1827 年首次观察到。(后面这部分要重新写)

### 1.2 熵产生

熵产生是非平衡态热力学的核心物理量，表征系统内部不可逆过程引起的总熵变化速率。其数学表达式基于热力学第二定律，其中总熵由系统熵变与环境熵变共同构成。该概念由伊利亚·普里高津提出理论框架，揭示开放系统中能量耗散与时空结构的形成机制，如贝纳德对流胞的形成正是熵产生驱动的耗散结构典型实例(后面这部分要重新写)

### 1.3 全息术

1948 年，丹尼斯·伽博 (Dennis Gabor) 提出了一种新的两步无透镜成像过程，他称之为波前重现，现在我们把它叫做全息术 [1]。即有一个合适的相干参考波与一个被物体衍射或散射的光波同时存在，那么衍射波或物波的振幅和相位的信息就能够被记录下来，尽管记录介质只对强度有响应。由这样的干涉图样最终可以得到原来物体的像。根据参考光与物光的相对位置关系，全息术分为两种：同轴全息和离轴全息。

1. 同轴全息：同轴全息由盖伯在最初提出时采用，其特点是参考光与物光沿相同方向传播，并在感光面上形成干涉图样。使用这种方法记录全息图时，物体中心、参考光中心和全息图中心三点共线。其光路的示意图如图1.1[2] 所示。

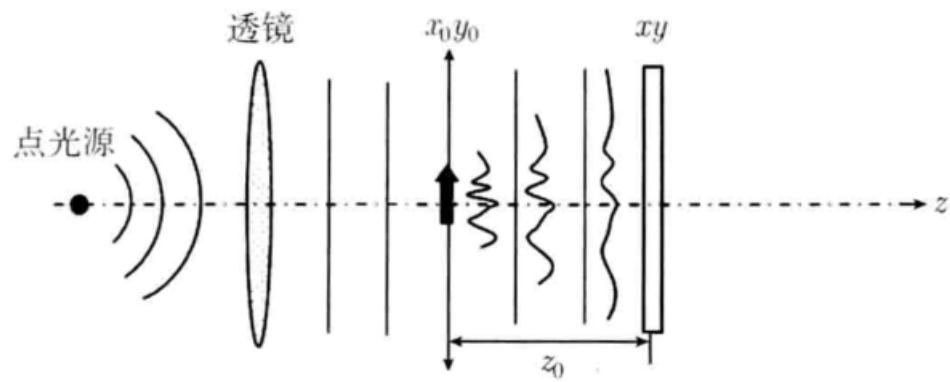


图 1.1: 同轴全息光路示意图

2. 离轴全息: 为克服同轴全息的像面重叠问题, 1962 年刘易斯·莱斯 (Emmett Leith) 与尤里·乌帕特涅克斯 (Jurīs Upatnieks) 提出了离轴全息方法。在该方案中, 参考光与物光以一定角度入射到记录面上, 从而在频谱空间中将真实像、共轭像和直流分量分离开来。其光路的示意图如图1.2[2] 所示。

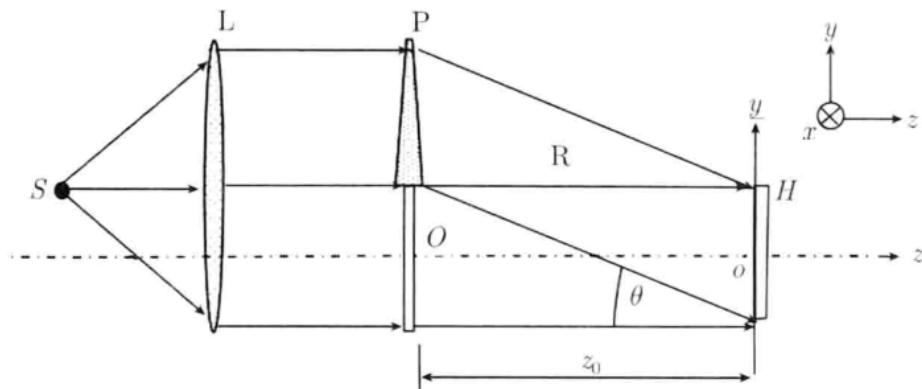


图 1.2: 离轴全息光路示意图

# Chapter 2

## 实验原理

### 2.1 有效动能温度公式推导

考虑被光阱束缚的布朗粒子运动，在自由粒子近似下忽略势能项，其朗之万方程为：

$$m \frac{dv}{dt} = -\gamma v + \xi(t) + \zeta(t), \quad (2.1)$$

其中：

- $v = \dot{x}$  为粒子速度；
- $\xi(t)$  为热噪声，对应环境温度  $T_w$ ；
- $\zeta(t)$  为外部随机力，振幅为  $\sigma$ 。

噪声的统计特性如下：

$$\langle \xi(t)\xi(t') \rangle = 2\gamma k T_w \delta(t-t'), \quad (2.2)$$

$$\langle \zeta(t)\zeta(t') \rangle = \sigma^2 \delta(t-t'), \quad (2.3)$$

$$\langle \xi(t)\zeta(t') \rangle = 0. \quad (2.4)$$

定义总噪声  $\eta(t) = \xi(t) + \zeta(t)$ ，其自相关函数为：

$$\langle \eta(t)\eta(t') \rangle = (2\gamma k T_w + \sigma^2) \delta(t-t'). \quad (2.5)$$

朗之万方程的稳态解为：

$$v(t) = \frac{1}{m} \int_{-\infty}^t e^{-\frac{\gamma}{m}(t-\tau)} \eta(\tau) d\tau. \quad (2.6)$$

计算速度方差：

$$\begin{aligned} \langle v^2(t) \rangle &= \frac{1}{m^2} \int_{-\infty}^t \int_{-\infty}^t e^{-\frac{\gamma}{m}(t-\tau)} e^{-\frac{\gamma}{m}(t-\tau')} \langle \eta(\tau)\eta(\tau') \rangle d\tau d\tau' \\ &= \frac{2\gamma k T_w + \sigma^2}{m^2} \int_{-\infty}^t e^{-\frac{2\gamma}{m}(t-\tau)} d\tau \\ &= \frac{2\gamma k T_w + \sigma^2}{2\gamma m}. \end{aligned} \quad (2.7)$$

根据能量均分定理:

$$\frac{1}{2}m\langle v^2 \rangle = \frac{1}{2}kT_{\text{kin}}, \quad (2.8)$$

代入上式可得:

$$T_{\text{kin}} = T_w + \frac{\sigma^2}{2k\gamma}. \quad (2.9)$$

因此, 外部随机力会等效提高布朗粒子的有效动能温度, 其增加量与噪声强度平方成正比, 与阻尼系数成反比。

## 2.2 噪声强度与扩散系数的修正关系

### 2.2.1 噪声强度的定义

从朗之万方程的过阻尼形式出发:

$$\gamma\dot{x} = f(x, t) + \eta(t), \quad (2.10)$$

其中  $\gamma$  为摩擦系数,  $\eta(t)$  为零均值随机噪声, 其完整统计特性为:

$$\langle \eta(t)\eta(t') \rangle = (2\gamma k_B T_w + \sigma^2) \delta(t - t'), \quad (2.11)$$

其中第一项为热噪声, 第二项为外加电场噪声。

### 2.2.2 扩散系数的推导

根据随机微分方程理论:

$$D = \frac{2\gamma k_B T_w + \sigma^2}{2\gamma^2}. \quad (2.12)$$

将第一项识别为 Einstein 关系:

$$D = \frac{k_B T_w}{\gamma} + \frac{\sigma^2}{2\gamma^2}. \quad (2.13)$$

### 2.2.3 引入迁移率

定义  $\mu \equiv 1/\gamma$ , 可得:

$$D = \mu k_B T_w + \frac{\sigma^2 \mu^2}{2}. \quad (2.14)$$

### 2.2.4 量纲验证

- $\gamma$ : [N·s/m] (摩擦系数)
- $\sigma^2$ : [ $N^2 \cdot s$ ] (噪声强度)
- $k_B T_w$ : [N·m] (热能)

检验非热噪声项的量纲:

$$\frac{\sigma^2}{2\gamma^2} \rightarrow \frac{[N^2 \cdot s]}{[N^2 \cdot s^2/m^2]} = [m^2/s], \quad (2.15)$$

与扩散系数的量纲一致。

### 2.2.5 熵产生率公式修正

概率流保持原定义:

$$J(x, t) = \mu f(x, t)p(x, t) - D \partial_x p(x, t), \quad (2.16)$$

其中  $D$  包含了非线性项  $\sigma^2 \mu^2 / 2$ 。熵产生率为:

$$\dot{\Sigma} = \frac{1}{D} \left\langle \left( \frac{J}{p} \right)^2 \right\rangle \geq \frac{\langle \dot{x} \rangle^2}{D}. \quad (2.17)$$

## 2.3 同轴全息的原理与重建

同轴全息 (In-line Holography) 是最早由 Gabor 提出的全息记录方式, 其特点是物光与参考光共轴传播, 通过干涉产生的全息图直接记录在探测平面上。该方法无需复杂的光路调整, 但重建时需解决共轭像与原始像重叠的问题。

### 2.3.1 原理

设物光复振幅为  $O(x, y)$ , 同轴参考光为平面波, 其复振幅可表示为

$$R(x, y) = A_r e^{i\phi}, \quad (2.18)$$

其中  $A_r$  为恒定振幅,  $\phi$  为初始相位。干涉强度分布为

$$I(x, y) = |O(x, y) + R(x, y)|^2 = |O(x, y)|^2 + |A_r|^2 + 2A_r \operatorname{Re}\{O(x, y)e^{-i\phi}\}, \quad (2.19)$$

其中第三项包含物光的振幅与相位信息。由于参考光与物光同轴, 所有频谱成分在频域中重叠于零频附近。

### 2.3.2 重建

同轴全息重建需通过相位恢复算法或多次曝光法分离重叠像。典型数值重建步骤如下 [3]:

- 对全息图  $I(x, y)$  进行二维傅里叶变换:

$$\mathcal{F}\{I(x, y)\} = \mathcal{F}\{|O|^2 + |A_r|^2\} + \mathcal{F}\{OR^*\} + \mathcal{F}\{O^*R\}, \quad (2.20)$$

- 利用迭代相位恢复算法(如 Gerchberg-Saxton 算法)从重叠频谱中提取物光场  $O(x, y)$ , 或通过多幅不同相移的全息图分离各成分:

$$O(x, y) = \frac{1}{4A_r} \sum_{k=1}^4 I_k(x, y) e^{-i\phi_k}, \quad (2.21)$$

其中  $\phi_k$  为第  $k$  次曝光引入的已知相移。

- 对提取的复振幅  $O(x, y)$  进行菲涅尔衍射积分, 实现物平面在任意距离  $z$  的重构:

$$U(x, y; z) = \mathcal{F}^{-1} \left\{ \mathcal{F}[O(x, y)] \cdot e^{i2\pi z \sqrt{\frac{1}{x^2} - f_x^2 - f_y^2}} \right\}. \quad (2.22)$$

同轴全息系统结构简单、光路稳定, 适用于粒子场测量与微小物体成像, 但需通过相位调制或算法处理解决像重叠问题。

## 2.4 离轴全息的原理与重建

离轴全息 (Off-axis Holography) 是一种通过在记录过程中引入参考光与物光的倾斜干涉来实现像分离的数字全息方法。其核心思想是在探测面上形成包含物光波前信息的干涉条纹，并通过空间频域滤波提取所需的复振幅信息，从而实现三维波场的数值重建。

### 2.4.1 原理

设物光复振幅为  $O(x, y)$ ，参考光复振幅为

$$R(x, y) = A_r e^{i2\pi(f_x x + f_y y)}, \quad (2.23)$$

其中  $A_r$  为参考光幅值， $f_x, f_y$  为由倾斜角决定的空间频率分量。干涉强度分布可表示为

$$I(x, y) = |O(x, y) + R(x, y)|^2 = |O(x, y)|^2 + |A_r|^2 + O(x, y)R^*(x, y) + O^*(x, y)R(x, y), \quad (2.24)$$

其中第三项和第四项分别为携带物光复振幅信息的正、共轭像调制项。

由于参考光与物光具有一定倾斜角， $O(x, y)R^*(x, y)$  和  $O^*(x, y)R(x, y)$  在频域中被移至原点两侧，不同于零频项  $|O|^2 + |A_r|^2$ ，从而可在频谱平面中利用带通滤波选取目标像的频谱区域，抑制共轭像与零级像的干扰。

### 2.4.2 重建

首先对干涉图  $I(x, y)$  进行二维傅里叶变换：

$$\mathcal{F}\{I(x, y)\} = \mathcal{F}\{|O|^2 + |A_r|^2\} + \mathcal{F}\{OR^*\} + \mathcal{F}\{O^*R\}, \quad (2.25)$$

通过滤波保留所需的  $\mathcal{F}\{OR^*\}$  项并进行逆傅里叶变换，可得到物光的复振幅分布（除去已知参考光相位）：

$$O'(x, y) = \mathcal{F}^{-1}[\mathcal{F}\{I(x, y)\} \cdot H(f_x, f_y)], \quad (2.26)$$

其中  $H(f_x, f_y)$  为空间频域带通滤波器。若需将物平面重构到某一距离  $z$ ，可利用菲涅尔衍射积分进行数值传播：

$$U(x, y; z) = \mathcal{F}^{-1} \left\{ \mathcal{F}[O'(x, y)] \cdot e^{i2\pi z \sqrt{\frac{1}{\lambda^2} - f_x^2 - f_y^2}} \right\}, \quad (2.27)$$

其中  $\lambda$  为波长， $f_x, f_y$  为空间频率坐标。

该方法通过频域分离实现单次拍摄下的相位重建，具有结构简单、计算高效、适应性强等优点，广泛应用于显微成像、位相测量及动态过程监测等领域。

# Chapter 3

## 理论模拟

# Chapter 4

## 装置设计

### 4.1 实验仪器

这里要插入我们装置的图片

### 4.2 光路图

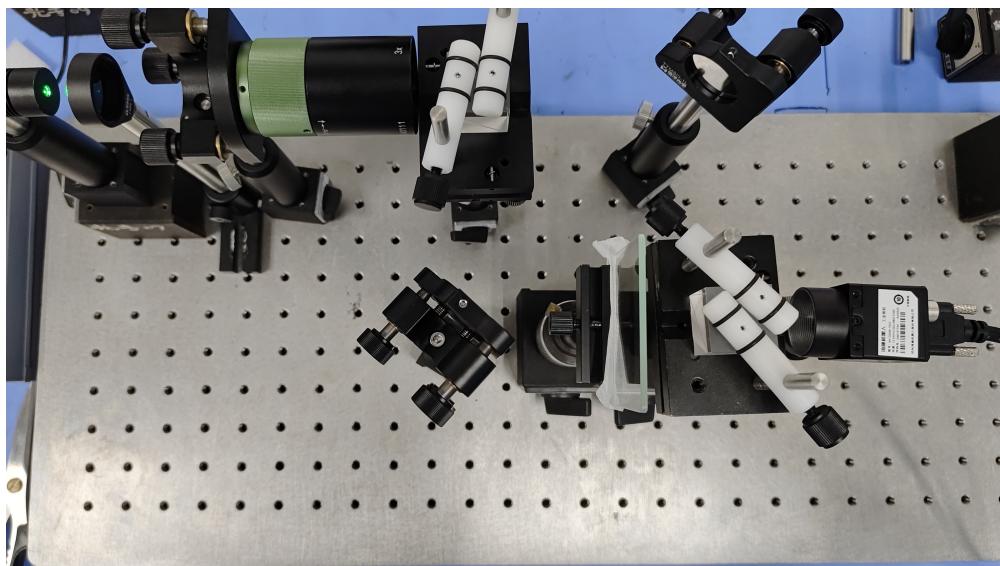


图 4.1: 离轴全息实验装置示意图

# Chapter 5

## 实验内容

### 5.1 离轴全息确定物体位置

#### 5.1.1 光路搭建与调试

基于离轴全息成像原理，实验光路搭建过程如图4.1。搭建过程分为：

**光源调制：**采用波长 520 nm 的半导体激光器（输出功率 20 mW），经扩束系统（ $3\times$  扩束器）形成直径 3 倍扩展的平行光束。

**分光系统：**平行光束入射至第一普通分光棱镜（BS1，50:50 分光比），分束为物光与参考光。物光路经反射镜 M1 转向后，照射分辨率板等样品，参考光路经反射镜 M2 反射后直接传输，两光路光程差控制在相干长度以内。最终会合在第二普通分光棱镜（BS2，50: 50 分光比）处，形成干涉图样。

**离轴干涉：**第二普通分光棱镜（BS2）以  $15^\circ$  偏转角实现离轴配置，物光与参考光干涉后形成离轴全息图，由科学级 CMOS 相机（像素尺寸 2.4 μm， $5472\times3648$  像素）记录。通过调整 CCD 位置，使 CCD 的接收屏位于两束光的重叠处。

#### 5.1.2 拍摄全息图

首先接通激光器电源，将 CCD 连接到计算机，打开相应拍摄软件。调整 CCD 的位置，使其处于物光和参考光完全重叠的位置。

然后调整 CCD 的曝光时间和增益，使得拍摄的全息图亮度适中，细节清晰。左右和上下移动物体（分辨率板），使用 CCD 拍摄不同位置的分辨率板全息图，记录下每张全息图并保存。

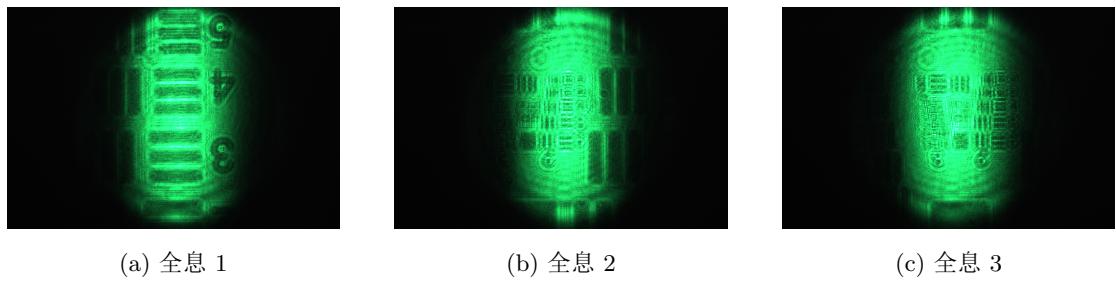


图 5.1: 对分辨率板的不同位置重建并确认距离

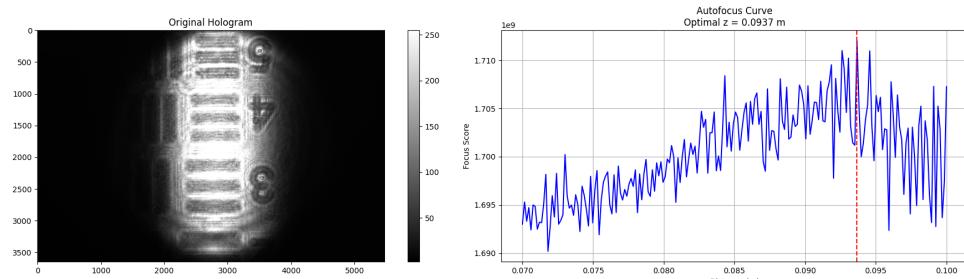
### 5.1.3 重建全息图

将拍摄的全息图导入计算机，使用 Python 进行处理。处理过程如下

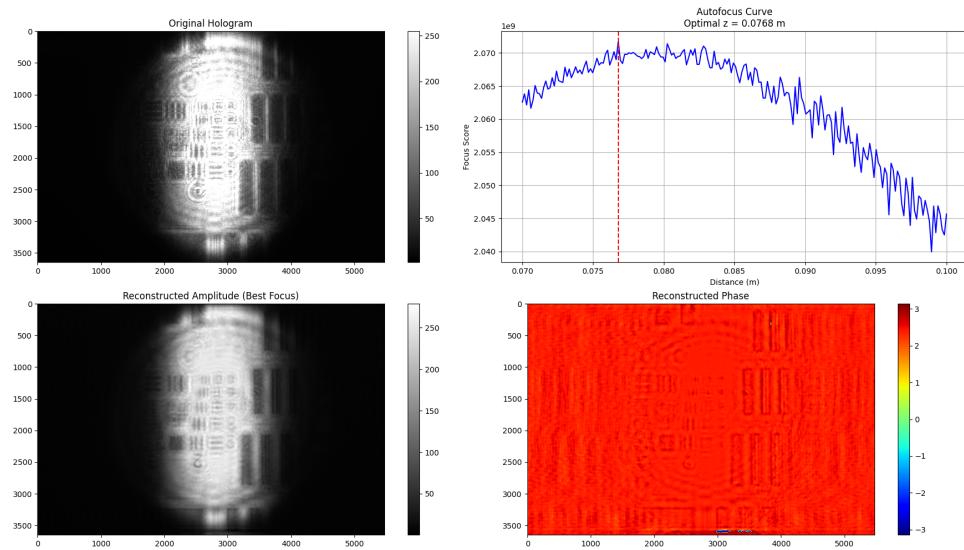
- 首先对全息图进行二维傅里叶变换，得到频域图像。
- 接着在频域图像中选择合适的带通滤波器，滤除零级像和共轭像的干扰。
- 然后对滤波后的频域图像进行逆傅里叶变换，得到物光的复振幅分布。
- 使用菲涅尔衍射积分方法，将物光复振幅传播到不同距离，重建出物体在不同位置的图像。使用算法评估图像的聚焦程度，在搜索范围内检索每一个  $z$  得到一个评价值函数。
- 由评价函数最大值对应的  $z$  值，作为物体的实际位置的  $z$

我们将上述过程集成于一个 python 程序，只需要将对应图片放入指定文件夹，运行程序即可完成重建。

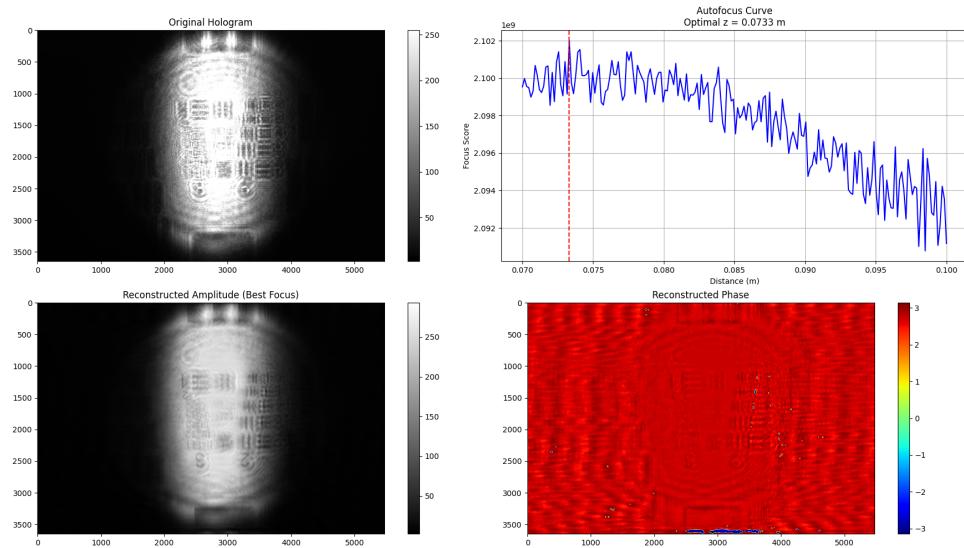
下面是我们使用5.1拍摄的全息图进行重建的结果展示，结果中展示了原全息图、确定距离  $z$  的过程图、在最佳距离  $z$  重建的重建图和重建图对应的相位分布：



(a) 离轴 1



(b) 离轴 2



(c) 离轴 3

图 5.2: 对分辨率板的不同位置重建并确认距离

# Chapter 6

## 误差分析

# Chapter 7

## 结论与展望

## Chapter 8

### 参考文献

- [1] Joseph W. Goodman. *Introduction to Fourier Optics*. Roberts and Company Publishers, 4th edition, 2017.
- [2] 李俊昌. 衍射计算机及数字全息上册. 科学出版社, 1st edition, 2014.
- [3] 李俊昌. 衍射计算机及数字全息下册. 科学出版社, 1st edition, 2014.

# Chapter 9

## 附录

### 9.1 人员分工

---

#### 队员 分工

---

队员一 负责搭建光路，全息重建的代码，实验报告的撰写等。

队员二 负责热学相关部分原理的推导和微球布朗运动的模拟等。

队员三 负责制作 ppt 和辅助光路搭建，查阅文献等。

队员四 负责后期数据处理，误差分析，查阅文献等。

---

### 9.2 部分代码展示

#### 9.2.1 重建离轴全息

```
1 import torch
2 import torch.fft
3 import cv2
4 import numpy as np
5 import matplotlib.pyplot as plt
6 import time
7
8 # 读取彩色 BMP，只取绿色通道
9 holo_color = cv2.imread(hologram_path, cv2.IMREAD_COLOR)
10 if holo_color is None:
11     raise FileNotFoundError(f"无法读取全息图文件: {hologram_path}")
12
13 holo = holo_color[:, :, 1].astype(np.float32) # 绿色通道
14 Ny, Nx = holo.shape
15
16 # 将数据转移到GPU
17 holo_gpu = torch.from_numpy(holo).to(device)
```

```

18
19 # 计算频谱
20 H = torch.fft.fft2(holo_gpu)
21 H_shifted = torch.fft.fftshift(H)
22 spectrum = torch.abs(H_shifted).cpu().numpy()
23
24 # 自动寻找实像频谱峰值位置
25 center = (Nx // 2, Ny // 2)
26 mask = np.ones_like(spectrum, dtype=bool)
27 mask[center[1] - 30:center[1] + 30, center[0] - 30:center[0] + 30] = False
28 max_pos = np.unravel_index(np.argmax(spectrum * mask), spectrum.shape)
29 filter_center = (max_pos[1], max_pos[0]) # (x, y)
30 print(f"Auto-detected filter center: {filter_center}")
31
32 # 生成圆形滤波器
33 Y, X = torch.meshgrid(torch.arange(Ny, device=device),
34                         torch.arange(Nx, device=device), indexing='ij')
35 R = torch.sqrt((X - filter_center[0]) ** 2 + (Y - filter_center[1]) ** 2)
36 mask_circle = (R < filter_radius).float()
37
38 # 滤波
39 H_filtered = H_shifted * mask_circle
40 H_unshifted = torch.fft.ifftshift(H_filtered)
41 field_filtered = torch.fft.ifft2(H_unshifted)
42
43 # 定义清晰度评价函数
44 def tenengrad(image):
45     """GPU加速"""
46     # Sobel算子
47     sobel_x = torch.tensor([[1, 0, -1], [2, 0, -2], [1, 0, -1]],
48                           dtype=torch.float32, device=device).view(1, 1, 3,
49                           3)
50     sobel_y = torch.tensor([[1, 2, 1], [0, 0, 0], [-1, -2, -1]],
51                           dtype=torch.float32, device=device).view(1, 1, 3,
52                           3)
53
54     img = image.unsqueeze(0).unsqueeze(0)
55
56     grad_x = torch.nn.functional.conv2d(img, sobel_x, padding=1)
57     grad_y = torch.nn.functional.conv2d(img, sobel_y, padding=1)
58
59     grad_mag = grad_x ** 2 + grad_y ** 2
60     return torch.sum(grad_mag)
61
62 for i, z_test in enumerate(z_values):
63     # 计算传播函数 - 修复维度问题
64     kz = torch.sqrt(k_sq - KX ** 2 - KY ** 2 + 0j)

```

```

63 H_prop = torch.exp(1j * z_test * kz)
64
65 # 传播重建
66 field_fft = torch.fft.fft2(field_filtered)
67 field_recon = torch.fft.ifft2(field_fft * H_prop)
68 amp = torch.abs(field_recon)
69
70 # 只计算中心ROI区域
71 roi_amp = amp[roi_y_start:roi_y_end, roi_x_start:roi_x_end]
72
73 # 计算清晰度得分
74 score = tenengrad(roi_amp)
75 all_scores.append(score.item())
76
77 # 更新最佳结果
78 if score > best_score:
79     best_score = score
80     best_z = z_test.item()
81     best_amp = amp.detach().clone()
82
83 # 使用最佳距离重建最终结果
84 kz_best = torch.sqrt(k_sq - KX ** 2 - KY ** 2 + 0j)
85 H_prop_best = torch.exp(1j * best_z * kz_best)
86 field_recon_best = torch.fft.ifft2(torch.fft.fft2(field_filtered) *
87                                     H_prop_best)
88 amp_best = torch.abs(field_recon_best)
89 phase_best = torch.angle(field_recon_best)

```

Listing 9.1: 离轴全息重建

### 9.3 部分全息图与拍摄图展示