

基于空间光调制器的数字全息技术研究

姓名:李昊霖 学号:202111998166 时间:2024/1/7

摘要

本文使用空间光调制器（SLM）开展了数字全息的算法比较研究，本文首先介绍了 SLM 的类型和功能，并重点介绍了液晶型空间光调制器的原理，然后介绍了 SLM 的应用场景和发展前景。之后，文章介绍了数字全息的原理并给出了三种常见的数字全息算法，分别为傅里叶全息算法、随机相位编码全息算法、G-S 全息算法，并使用 SLM 展开全息实验，发现 G-S 算法表现最好。

关键词：数字全息，空间光调制器，G-S 算法

目录

| | |
|---|----|
| 一、空间光调制器介绍 | 1 |
| 1.1 空间光调制器分类 | 1 |
| 1.2 液晶空间光调制器原理 | 2 |
| 1.3 空间光调制器的应用场景及发展前景 | 3 |
| 1.3.1 应用场景 | 3 |
| 1.3.2 发展前景 | 4 |
| 二、数字全息 | 4 |
| 2.1 数字全息简介 | 4 |
| 2.2 菲涅尔衍射积分 | 4 |
| 2.3 计算全息 | 5 |
| 2.3.1 傅里叶全息术 | 5 |
| 2.3.2 随机相位编码全息术（Random Phase Encoding Holography, RPEH） | 7 |
| 2.3.3 G-S（Gerchberg-Saxton）算法 | 8 |
| 2.4 总结 | 10 |

一、空间光调制器介绍

1.1 空间光调制器分类

空间光调制器（Spatial Light Modulator, SLM）是一种便携、高效而强大的光场调控手段，它可以实现光场振幅、相位以及偏振状态的精细调控，SLM 有多种实现方式，原理基于电光、声光或光学非线性效应，以下是一些常见的空间光调制器的工作原理：

- 电光调制器（Electro-Optic Modulator）：这种类型的调制器使用电场来改变材料的折射率，从而调制光的相位。常见的电光材料包括锂钨酸铁钠（LiNbO₃）和硅（Si），当施加电场时，材料的折射率发生变化，导致通过材料的光波的相位发生变化。
- 声光调制器（Acousto-Optic Modulator）：声光调制器利用声波的传播来调制

光的相位，光波通过通过光学晶体，其中通过超声波驱动晶体表面的声波波动，这会引起光的折射率变化。由于声波频率通常远远低于光波频率，这种调制器适用于低频率调制。

- 液晶空间光调制器（Liquid Crystal Spatial Light Modulator）：液晶是一种能够通过应用电场改变分子排列的材料，从而改变光的传播方向的物质。在液晶空间光调制器中，液晶层被放置在光学系统中，通过调整电场来调制光的相位或振幅。
- 超表面空间光调制：超表面利用亚波长结构，极为精细的调节光场的各个参数，而且可以高密度集成，是一种极为强大的调制手段。但缺点是只能实现某种固定的调制，可调性差。

不同类型的空间光调制器在应用中有各自的优缺点，具体的选择取决于特定的应用需求。本文主要研究**液晶空间光调制器**的原理及应用，这也是目前应用最广泛、成本较低的一种调控手段。

1.2 液晶空间光调制器原理

液晶空间光调制器（LC-SLM）是一种基于液晶技术的光学器件，液晶是一种有机分子，具有介于液体和固体之间的特性，具有显著的双折射效应。液晶分子在不同的电场作用下可以改变其排列状态，这是液晶空间光调制器工作的基础。

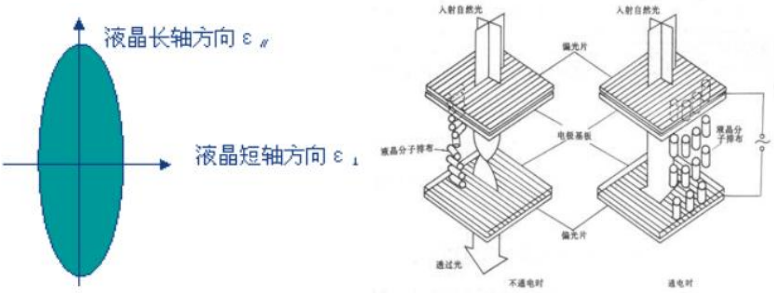


图 1 液晶的双折射与液晶光开关

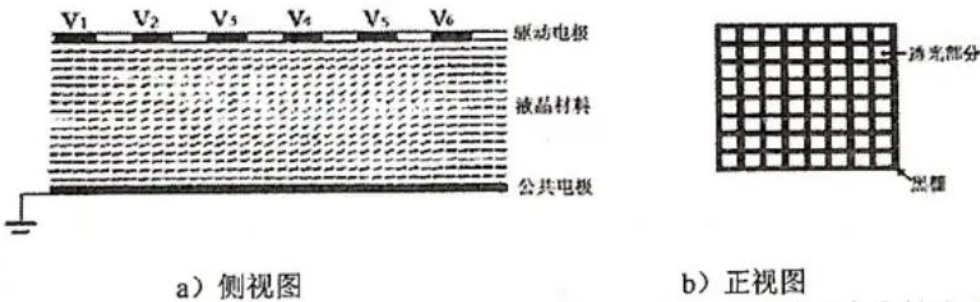


图 2 液晶空间光调制器原理

液晶空间光调制器通常包含两块平行的透明基板，其中涂有透明导电层，以形成电极。这两块基板之间注入了液晶材料，并且表面通常涂有对齐层，以指导液晶分子的初始排列方向。

在没有电场作用时，液晶分子遵循对齐层的导向，具有一定的排列方向。这可以是平行排列、螺旋排列等，具体取决于液晶材料和对齐层的性质。当在液晶层施加电场时，通过透明电极的电场会影响液晶分子的排列。电场使液晶分子发生扭曲或重新排列，改变液晶层的光学性质，尤其是折射率。

相位调制：通过调整电场的强度或方向，可以精确地控制液晶层的折射率。

当光通过液晶层时，由于电场的作用，光的相位会发生变化。这样就可以实现对光波的相位调制，从而产生相位调制的光场。

振幅调制：利用液晶的螺旋排列可以实现旋光性，通过在液晶层前后加两个线偏振片就可以实现振幅调制。假设两个线偏振片透振方向平行，当偏振面旋转 θ 时，透过的光强变为入射的 $\cos^2\theta$ ，可以通过控制电场控制液晶的旋光性，进而控制透射光强实现振幅调制。

1.3 空间光调制器的应用场景及发展前景

1.3.1 应用场景

● 复杂结构光产生

只需改变加载在空间光调制器上的位相图就可实时生成不同拓扑荷数的涡旋光束，还可以生成近似的拉盖尔光束、贝塞尔光束等复杂光束。

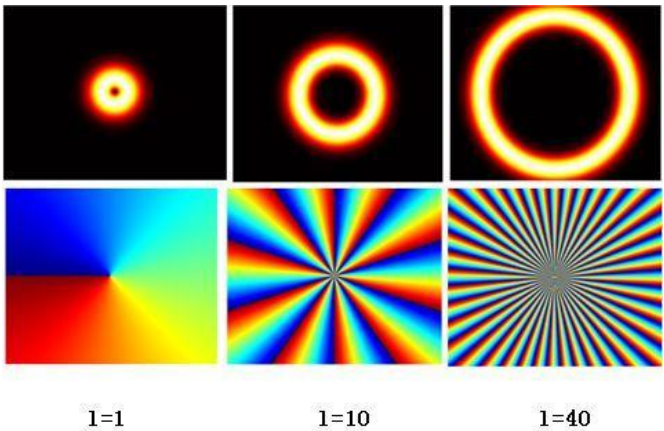


图3 空间光调制器产生涡旋光示意图

- 全息光镊，空间光调制器可精确控制入射激光束的波前相位分布，在囚禁原子的区域得到各种需要的光场分布，即光场强度梯度，为自由操控原子提供了可能，可对多目标粒子在三维空间进行任意操控。
- 光束整形——激光加工，例如平顶光束整形（图4），在激光束整形、激光干涉等应用中，SLM的发展有助于更精确地控制激光光场，提高激光加工的精度和效率。这对于激光医学、激光制造等领域具有重要意义。

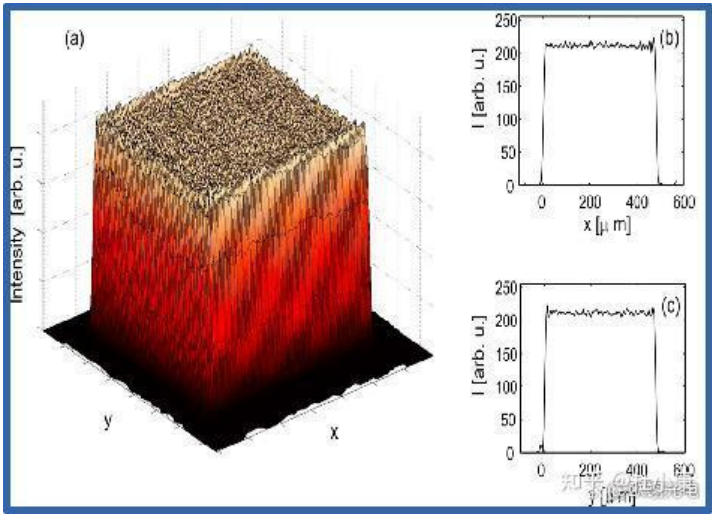


图4 平顶光束整形

- VR 与全息投影可以通过空间光调制实现实时的 3D 全息投影，在 VR/AR 技术上有重大应用前景。
- 光通信：随着光通信需求的增加，SLM 在光通信系统中的应用前景广阔。通过使用 SLM 进行相位调制，可以实现更高速率和更复杂的光通信系统，提高传输效率和容量。

1.3.2 发展前景

- 高分辨率和高带宽：未来的空间光调制器需要更高的分辨率和更大的带宽，以满足日益增长的数据传输需求和更精细的光场控制。
- 高损伤阈值：在激光制造、激光武器等领域需要极强的激光束，为了适应这些场景应用的需要，SLM 需要不断提高损伤阈值。
- 多波长适应性：随着多波长光源的广泛应用，SLM 需要具备适应不同波长的能力，以适应不同应用场景和光源。
- 集成光学：将 SLM 集成到微纳米光学系统中，实现更小型化、轻量化、低功耗的设备，以满足现代便携式设备和集成光学系统的需求。
- 新型材料和技术：探索新型的液晶材料、非线性光学材料等，以及创新的制备技术，提高 SLM 的性能和稳定性。
- 深度学习与 SLM 的结合：利用深度学习等先进技术，将其与 SLM 结合，实现更高级的光场调制和图像处理，拓展 SLM 在人工智能、虚拟现实等领域的应用。

二、数字全息

2.1 数字全息简介

传统的全息术是利用全息干板对干涉条纹进行记录，便能够实现对物光的相位信息记录的目的。具体方法是首先用激光照射物体并衍射，当物光传播到全息干板时，用另一束与物光相干的参考光同时照射全息干板，此时物光和参考光会在全息干板上发生干涉。然后对全息干板进行曝光，全息干板上记录到的便是物光和参考光相干涉的条纹，此时记录到的干涉条纹的形状、强度、疏密程度便代表物光的振幅和相位信息。

但实际上，物光的产生与干涉是可以利用计算机进行计算与模拟的。所以，如果我们不通过光学方法，而是使用计算方法制作全息图，这就称为计算全息。光学全息记录必须是实际存在的物体，而计算全息则并不需要所记录的物体都是实际存在的物体，其能够生成世间根本不存在的物体（通过计算机虚拟得到的虚拟物体）的全息图，因而具有独特的优点和极大的灵活性。随着计算机技术的飞速发展，计算全息已经在诸如光信息处理、干涉计量等领域得到了广泛应用。由于其编码具有灵活性高和可重复强的优点，已经逐渐所取代传统光学全息。

2.2 菲涅尔衍射积分

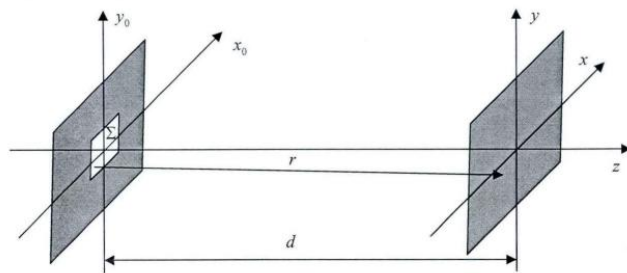


图 5 菲涅尔衍射积分示意图

$$U(x, y) = \frac{\exp(jkd)}{j\lambda d} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} U_0(x_0, y_0) \times \exp \left\{ \frac{jk}{2d} \left[(x - x_0)^2 + (y - y_0)^2 \right] \right\} dx_0 dy_0$$

光的传播是计算全息的基础，不管是物光与参考光的干涉，还是利用全息图再现波前，都离不开菲涅尔衍射积分。不同的全息算法源自于不同的菲涅尔衍射积分计算方法，如傅里叶变换算法（S-FFT）、卷积算法（T-FFT）和角谱衍射算法（D-FFT）。

应当指出，这三种算法只是对于菲涅尔积分的不同计算方式，在连续的情况下三者完全等价。但是，在使用计算机解决问题时，必须要进行离散化，这时三种算法就表现出了各自的特点。其中 S-FFT 算法是最直接，也是较为快速的方法，但在同等取样间隔下表现要比其他算法差。T-FFT 是同等取样间隔下表现最好的算法，但运算速度较慢。D-FFT 是这些算法中最快速的算法，虽然它进行了两次（逆）傅里叶变换，但是由于其整体计算简单，所以在其他的计算步骤上消耗的时间远小于 S-FFT，这导致其计算速度快于 S-FFT。

2.3 计算全息

计算全息算法多种多样，根据不同的原理和要求可以设计不同的全息算法。由于数字全息术通常由空间光调制器（SLM）来实现，而 SLM 多为相位调制型¹，所以数字全息算法的研究集中在相位型全息图。一般来讲，不存在一种方法能完美还原物光波前，所以全息算法使用了不同的方法来近似物光波前，大致分为迭代型和非迭代型。其中非迭代型基于一些近似原理直接计算相位图，而迭代型通过迭代优化的方式不断逼近物光，一般迭代型算法可以达到更好的优化效果。而且利用智能优化算法，比如遗传算法、梯度下降法可以达到更好的优化效果，目前这仍是一个活跃的研究领域。

2.3.1 傅里叶全息术

傅里叶变换全息图是由菲涅尔公式的傅里叶算法直接发展而来的，主要用来生成纯相位全息图，算法流程如下图所示：

- （1）创建物体模型：选择一幅图片或者创建一个物体模型。
- （2）傅里叶变换：对图像进行傅里叶变换，得到频域信息。
- （3）生成全息图：保留相位信息，构建纯相位全息图。
- （4）全息图重建：使用逆傅里叶变换将从纯相位频域信息中重建全息像。

¹ 即使是振幅型空间光调制器，也往往是基于相位型开发的。例如，使用液晶的旋光性，在出射光上加一偏振片，从而通过调控旋光性调控振幅。但在调控旋光性时，实际也就调控了相位。所以，这样的空间光调制器是振幅相位复合型的，但是又无法单独调控振幅和相位。往往相位型全息图也能在这种 SLM 上运行，但效果可能不如纯相位型 SLM。

傅里叶全息算法的代码见附录 1.1，我们使用该算法对一幅图片进行了全息重建，原图、全息图、重建全息像如下图所示。

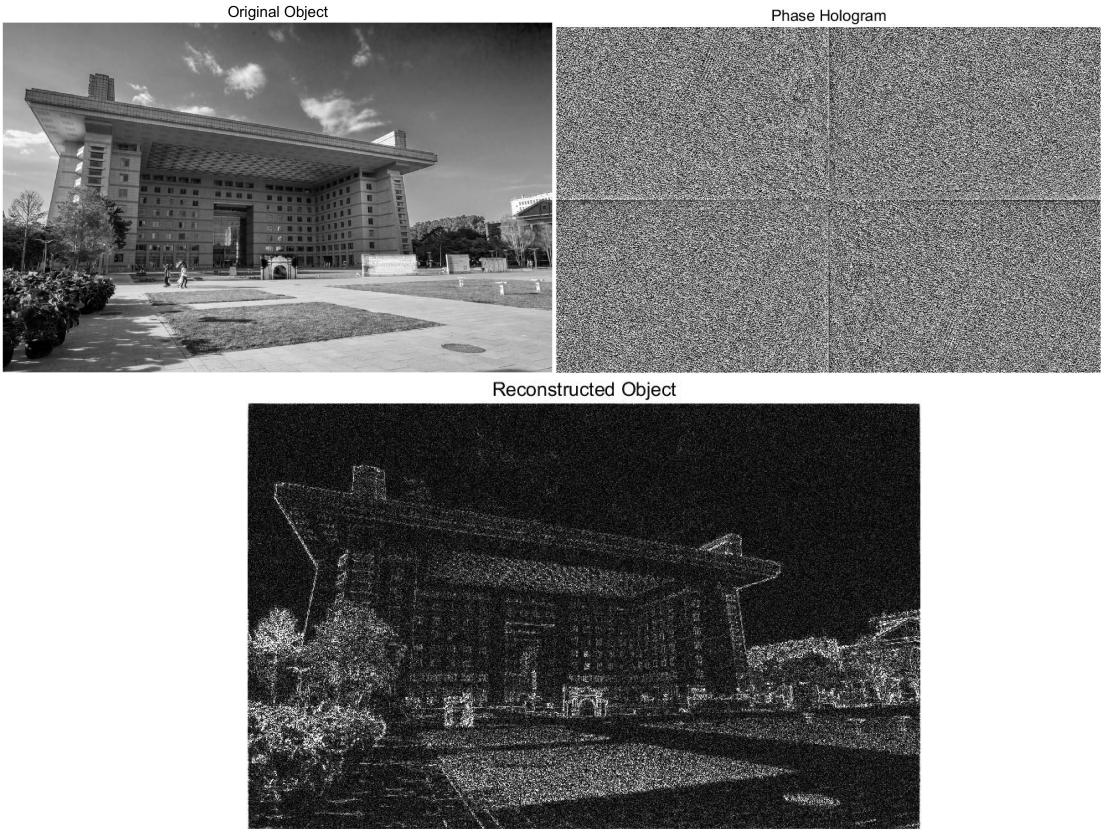


图 6 相位型傅里叶全息术及重建效果图

该算法步骤简单，计算速度快，但是全息重建效果较差，因为在全息图构建过程中简单的丢弃了幅度信息，这导致全息图对于灰度梯度大的地方(亮暗变化明显)重建效果更明显。这一点可以由下面的圆环全息图证明，全息像只显示出了黑白交界处，也就是灰度突变的地方，只能看到圆环的轮廓，这是傅里叶全息算法的不足之处。

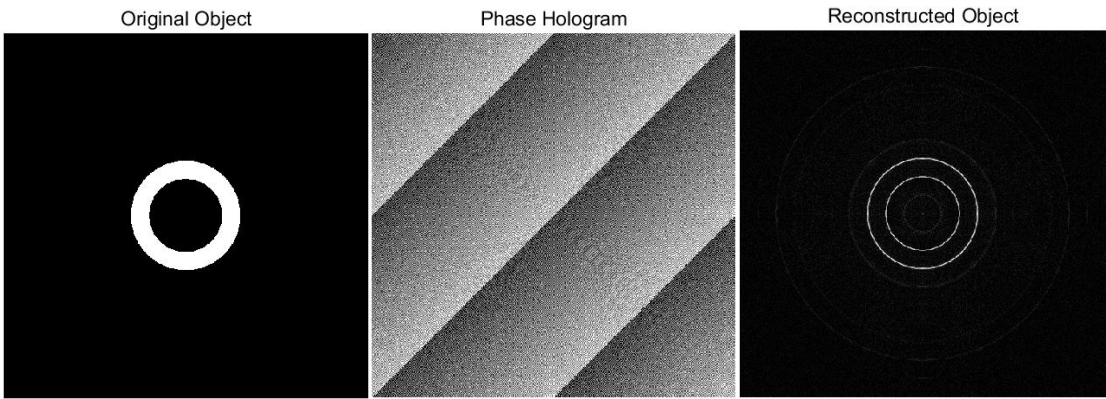


图 7 相位型傅里叶全息术圆环效果图

下面是我们利用空间光调制器得到的实验结果，与模拟基本吻合。因为没有严格共轴，所以出现了共轭像和高阶衍射像。

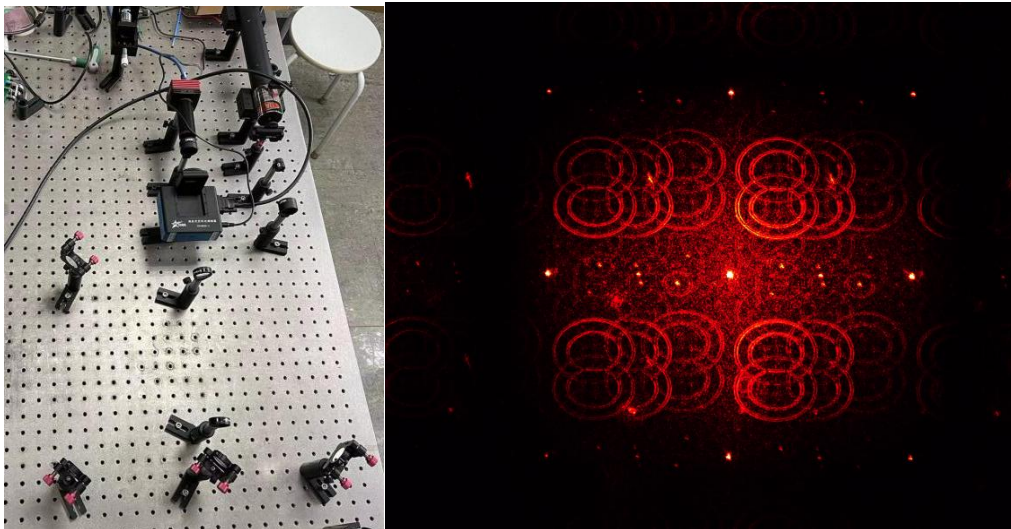


图 8 圆环的相位型傅里叶全息术实验图

2.3.2 随机相位编码全息术 (Random Phase Encoding Holography, RPEH)

随机相位编码全息术利用随机的相位编码方式来记录和再现物体的光学信息。这种方法允许通过编码复杂的相位信息而不需要高分辨率的空间光调制器，从而在光学系统中实现全息图的生成和再现。该算法主要步骤如下：

- (1) 读入图像，处理为灰度图，此时图像为纯实数灰度图。
- (2) 给图像引入随机相位噪声，具体做法是将每个像素的灰度值乘以一个随机相位。
- (3) 进行傅里叶变换。
- (4) 获取相位分布，并对 2π 取余，通过余数绘制纯相位全息图。
- (5) 逆傅里叶变换得到重建的全息像。

算法代码见附录 1.2，纯相位全息图演示如下：

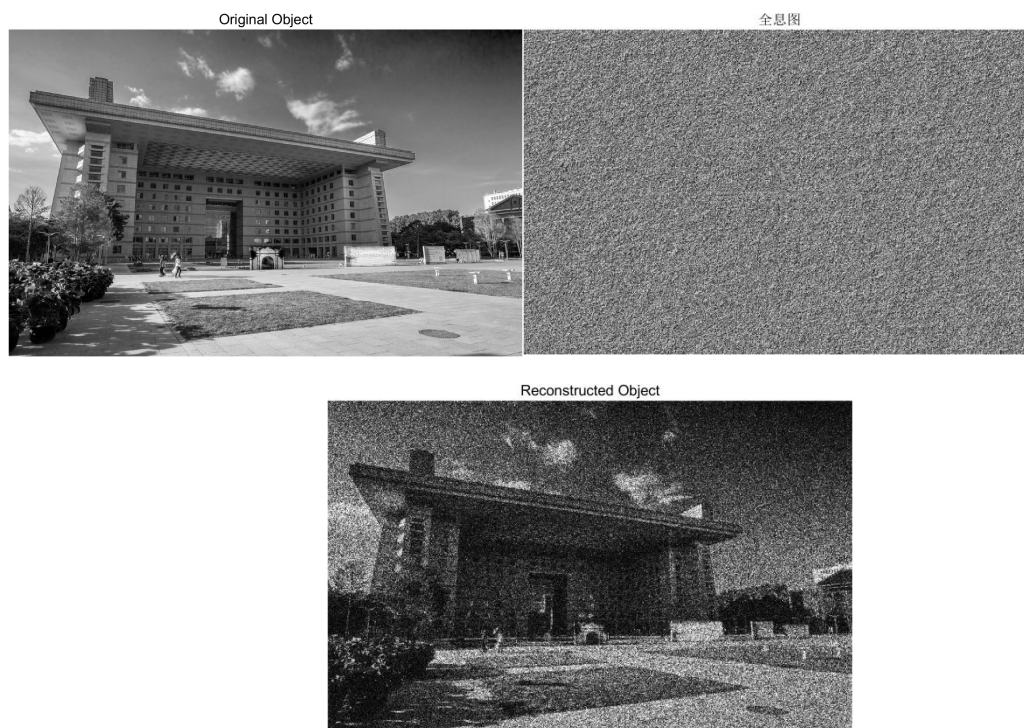


图 9 RPEH 相位型全息效果图

理论上，可以通过将相位分布转化为灰度值来构建振幅型全息图，但是重建全息像效果极差，如下图所示。

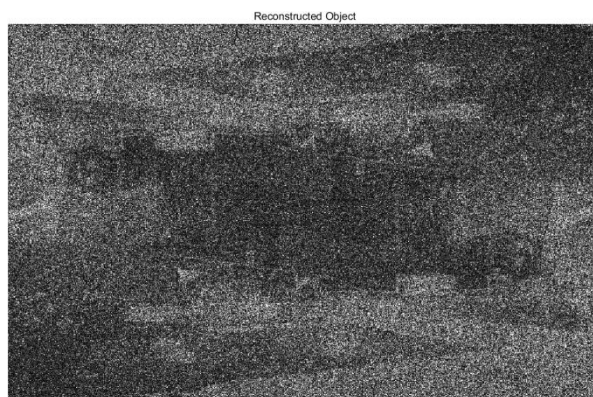


图 10 RPEH 振幅型全息效果图

随机相位编码全息术的优点之一是它不需要高分辨率的光空间调制器，因此可以降低系统的成本。然而，它的缺点是由于相位信息的随机编码，再现图像通常受到噪声和模糊的影响，可能会导致图像的质量降低。因此，在实际应用中，通常使用更复杂的相位编码方式来构建全息图。

下面是我们利用相位全息图构建的五角星全息像，在构建过程中要不断尝试选择合适的随机化参数，相位随机程度过高过低都会对全息效果造成影响。

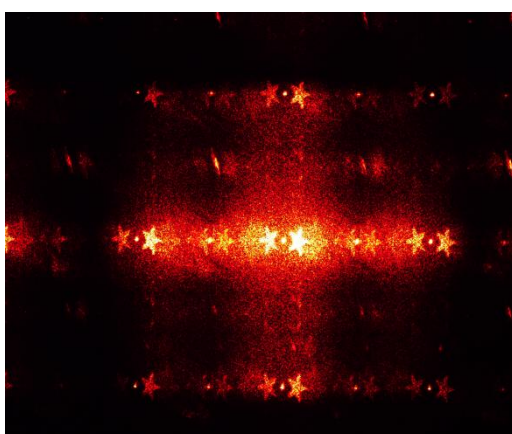


图 11 RPEH 相位型五角星全息实验图

2.3.3 G-S (Gerchberg-Saxton) 算法

Gerchberg-Saxton (GS) 算法是一种用于相位恢复和重构的迭代算法，常用于数字全息术和相位恢复问题。实际上，全息术要求我们只利用纯相位调制恢复物光波前，这一近似问题没有直接的精确求解方法，但可以使用迭代优化的方式得到近似的解。

GS 算法的目标是通过迭代过程逐渐优化相位估计，以最大程度地匹配已知的幅度信息和测量的干涉图案。通过多次迭代，GS 算法可以在幅度已知的情况下，逐步恢复出光波的复杂相位分布，从而实现高质量的图像重构。这一算法的收敛性在 1982 年已由 Fienup 证明²。

需要注意的是，GS 算法并不总是能够收敛到全局最优解，特别是在存在噪声或强烈的非线性变换的情况下。因此，在应用 GS 算法时，合适的参数设置和初

² Fienup J R. Phase retrieval algorithms: a comparison[J]. Applied optics, 1982, 21(15): 2758-2769.

始估计的选择非常重要。

算法流程如图：

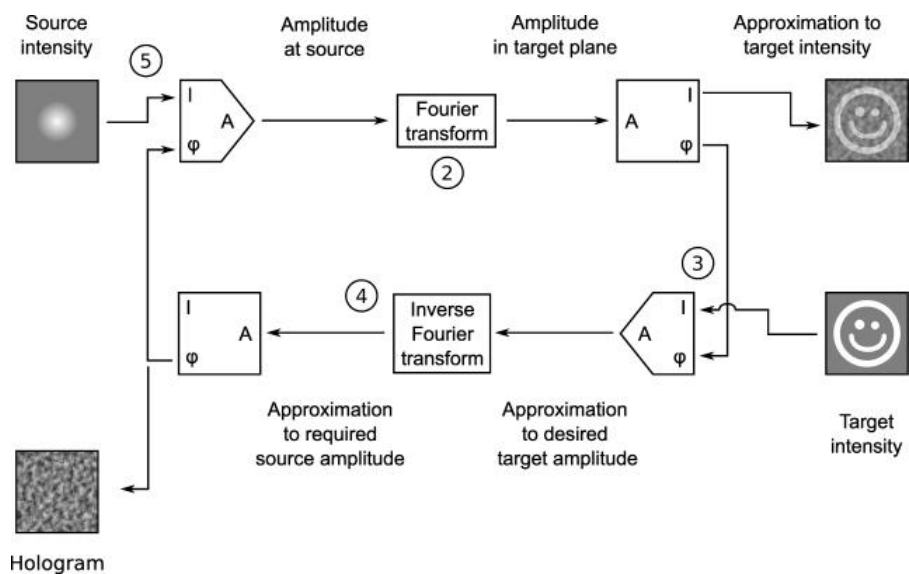


图 12 相位型全息 GS 算法流程图

我们提供了该算法的 Python 代码，见附录 1.3，该算法演示如下，迭代 50 次后可见其效果较好。

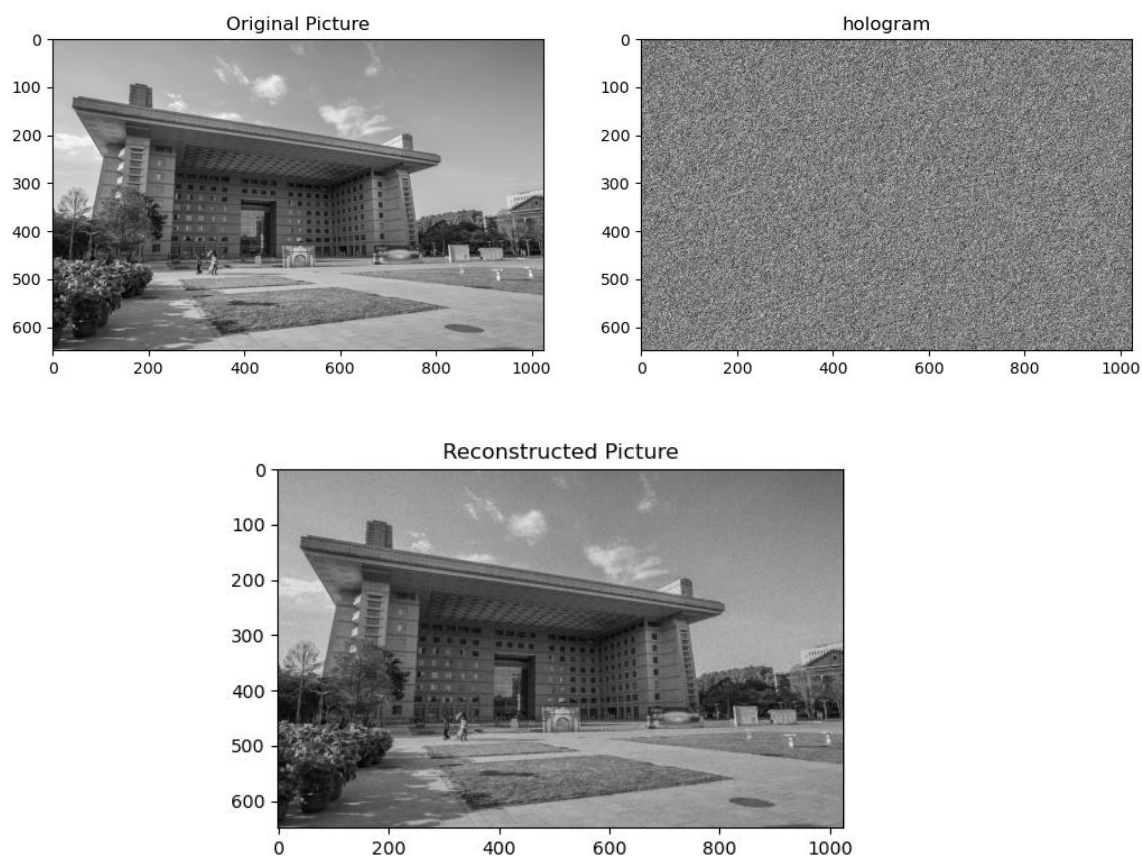


图 13 相位型 GS 全息算法效果图

下面是我们开展实验的光路以及实验结果：

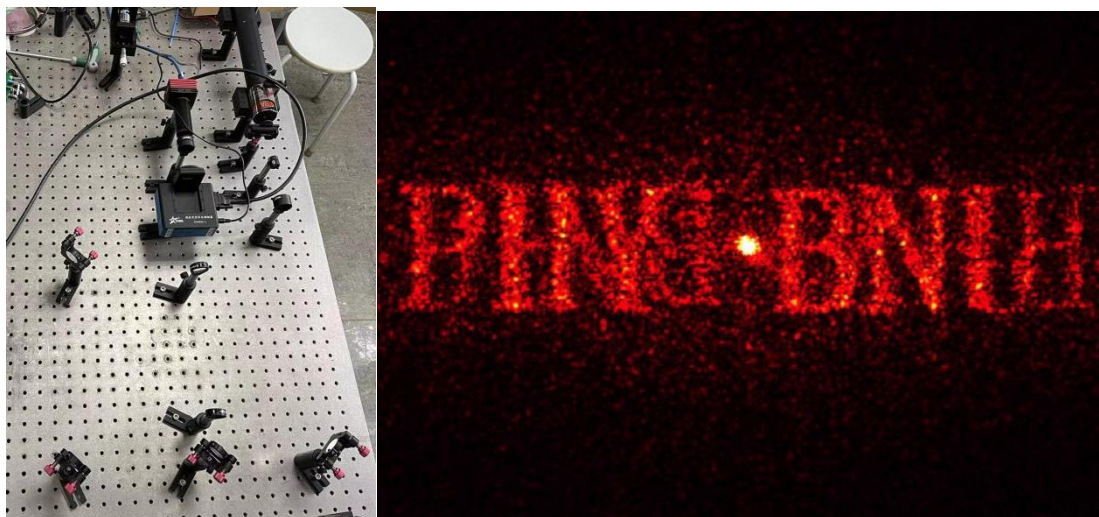


图 14 相位型全息 GS 算法效果图

2.4 总结

数字全息技术是一种利用数字计算和空间光调制器实现全息成像的方法。在数字全息中，记录下来的全息图被数字化并经过计算处理，以重构出原始的物波场。这种方法不仅提供了更高的成像精度和稳定性，同时也便于数据的存储、传输和处理。

在数字全息中，傅里叶全息术和随机相位编码全息术是最为常见的两种算法。傅里叶全息术基于傅里叶变换的原理，将物体的频域信息转换为全息图的强度分布。而随机相位编码全息术则通过引入随机相位编码来提高物波场的重建质量。此外，G-S（Gerchberg-Saxton）算法等优化算法也被应用于数字全息中，以提高重建精度和稳定性，让全息算法有更好的性能表现。

数字全息技术具有广泛的应用前景，尤其是在生物医学、光学计量、显示技术等领域。例如，在生物医学领域，数字全息技术可用于细胞和组织的显微成像，以研究生物结构和功能。在光学计量领域，数字全息可以用于表面形貌和光学元件的测量，以实现高精度的光学元件加工和检测。在显示技术领域，数字全息可以实现三维立体显示，提供更真实的视觉体验。

随着数字全息技术的不断发展，我们可以预见到以下一些趋势：

- 高分辨率和高帧率成像：随着对生物医学和高速动态场景的需求增加，数字全息技术将向着高分辨率和高帧率的方向发展。这需要更高效的算法和更快的计算能力来实现。
- 多波长和多视角成像：数字全息技术可以结合多波长和多视角的成像方法，以获取更多的物波场信息。这将有助于提高重建精度和稳定性。
- 智能化和自动化：数字全息技术将与人工智能、机器学习等先进技术相结合，实现智能化和自动化的物波场重建。这可以大大提高重建效率和精度，同时降低对人工干预的需求。
- 跨界融合与创新应用：数字全息技术可以与其他领域的技术相结合，开拓出更多创新应用。例如，与虚拟现实、增强现实、智能传感器等技术结合，可以实现更丰富的交互体验和应用场景。

总之，数字全息技术作为光学成像领域的重要分支，其发展前景广阔。随着技术的不断进步和应用需求的增加，数字全息技术将不断突破自身的限制，向着更高分辨率、更高帧率、更智能化和自动化的方向发展。同时，数字全息技术也将与其他领域的技术相结合，开拓出更多创新应用，为人类的生活和工作带来更多的便利和价值。

参考文献

- [1] 苏衍峰. 基于空间光调制器的动态全息三维显示技术研究[D]. 苏州大学, 2020. DOI:10.27351/d.cnki.gszhu.2019.000510.
- [2] 李俊昌, 衍射计算及数字全息(上册), 科学出版社, 2014.
- [3] 钟锡华, 现代光学基础, 北京大学出版社, 2003.

附录一

代码运行前请将下面图片设为路径 C:\Users\De'l'l\Desktop\bnu.png



附录 1.1 傅里叶全息算法 Matlab 代码

```
1. % 参数设置
2. clear
3. Uo=imread("C:\Users\De'l'l\Desktop\bnu.png"); %读入图像
4. Uo=double(Uo(:,:,1));
5. [image_sizea,image_sizeb] = size(Uo); % 图像大小
6. wavelength = 632.8e-9; % 波长(米)
7. k = 2 * pi / wavelength; % 波数
8.
9. % 创建物体模型
10. object_model = Uo;
11. % 傅里叶变换
12. object_fft = fftshift(fft2(object_model));
13.
14. % 生成纯相位全息图
```

```

15. phase_hologram = angle(object_fft);
16.
17. % 反向傅里叶变换（将幅度设为常数）
18. reconstructed_object = ifft2(exp(1j * phase_hologram));
19.
20. % 显示结果
21. figure;
22. subplot(1, 2, 1);
23. imshow(phase_hologram, []);
24. title('Phase Hologram');
25.
26. subplot(1, 2, 2);
27. imshow(abs(reconstructed_object)*200);
28. title('Reconstructed Object');
29.
30. colormap gray;

```

附录 1.2 随机相位编码全息算法

```

1. Uo=imread("C:\Users\De'l'l\Desktop\bnu.png"); %读入图像
2. %Uo=imresize(Uo,[1080,1920]); %将图像调整到 1080×1920 像素
3. Uo=double(Uo(:,:,1));
4. figure;
5. imshow(Uo/255)
6. title("Original Object")
7.
8. [r,c]=size(Uo); %返回矩阵大小
9. ef=0.5; %设置纯相位随机噪声系数
10. FUo=fftshift(fft2(Uo.*exp(1j.*rands(r,c).*pi.*ef))); %计算傅里叶变换
11. phi=angle(FUo); %计算相位分布
12. H=mod(phi,2*pi); %计算相位取模数  $2\pi$  的余数
13. H1=round(H/max(H(:))*255); %将余数量化为 255 个灰度级，绘制全息图
14. CGH=exp(1j.*H1/40.58); %漂白制作成纯相位全息图（理想情况下相位还原到  $0\sim 2\pi$ ，  
40.58=255/2π），导入空间光调制器
15. rU=ifft2(CGH); %进行逆傅里叶变换得到再现光场
16.
17. figure;
18. imshow(H1/255); %显示编码得到的全息图
19. title("全息图")
20. R = rU.*conj(rU);
21. figure;
22. imshow(R/max(max(R))*4); %显示再现像
23. title("Reconstructed Object")

```

附录 1.3 Gerchberg - Saxton 全息算法 Python 代码


```

1. import math
2. import numpy as np
3. import matplotlib.pyplot as plt
4.
5. img = plt.imread('C:/Users/De\l\l/Desktop/bnu.png')
6. img = 0.2126 * img[:, :, 0] + 0.7152 * img[:, :, 1] + 0.0722 * img[:, :, 2];
7. plt.subplot(131),plt.imshow(img,'gray'),plt.title('Original Picture')
8.
9. def Ger_Sax_algo(img, max_iter):
10.     h, w = img.shape
11.     pm_s = np.random.rand(h, w)
12.     pm_f = np.ones((h, w))
13.     am_s = np.sqrt(img)
14.     am_f = np.ones((h, w))
15.
16.     signal_s = am_s*np.exp(pm_s * 1j)
17.
18.     for iter in range(max_iter):
19.         signal_f = np.fft.fft2(signal_s)
20.         pm_f = np.angle(signal_f)
21.         signal_f = am_f*np.exp(pm_f * 1j)
22.         signal_s = np.fft.ifft2(signal_f)
23.         pm_s = np.angle(signal_s)
24.         signal_s = am_s*np.exp(pm_s * 1j)
25.
26.     pm =pm_f
27.     return pm
28.
29. pm = Ger_Sax_algo(img, 20);
30. plt.subplot(132),plt.imshow(pm,'gray');
31. plt.title("hologram")
32. U=np.fft.ifft2(np.exp(1j*pm))
33. #shift2center=np.fft.fftshift(U)
34. #对傅立叶变换的结果进行对数变换，并显示效果
35. log_U = np.log(1 + np.abs(U*np.conjugate(U))*1000)
36. plt.subplot(133),plt.imshow(log_U,'gray');
37. plt.title("Reconstructed Picture")

```