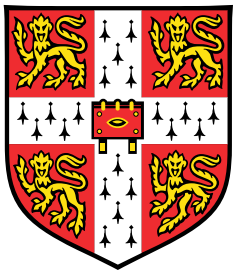
In search for the optimal phase hologram



Jinze Sha

Supervisor: Prof. Timothy D. Wilkinson

Department of Engineering
University of Cambridge

This dissertation is submitted for the degree of
Doctor of Philosophy

King's College

September 2024

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements. This dissertation contains fewer than 65,000 words including appendices, bibliography, footnotes, tables and equations and has fewer than 150 figures.

Jinze Sha
September 2024

Acknowledgements

And I would like to acknowledge ...

Abstract

This is where you write your abstract ...

List of Publications

[1] Jana Skirnewskaja, Yunuen Montelongo, Jinze Sha, and Timothy D. Wilkinson. Holographic lidar projections with brightness control. In *Imaging and Applied Optics Congress 2022 (3D, AOA, COSI, ISA, pcAOP)*, page 3F2A.6. Optica Publishing Group, 2022

[2] Jinze Sha, Andrew Kadis, Fan Yang, and Timothy D. Wilkinson. Limited-memory bfgs optimisation of phase-only computer-generated hologram for fraunhofer diffraction. In *Digital Holography and 3-D Imaging 2022*, page W3A.3. Optica Publishing Group, 2022

[3] Andrew Kadis, Benjamin Wetherfield, Jinze Sha, Fan Yang, Youchao Wang, and Timothy D. Wilkinson. Effect of bit-depth in stochastic gradient descent performance for phase-only computer-generated holography displays. *London Imaging Meeting*, 3:36–40, 7 2022

[4] Jinze Sha, Andrew Kadis, Fan Yang, Youchao Wang, and Timothy D. Wilkinson. Multi-depth phase-only hologram optimization using the l-bfgs algorithm with sequential slicing. *J. Opt. Soc. Am. A*, 40(4):B25–B32, Apr 2023

[5] Jinze Sha, Adam Goldney, Andrew Kadis, Jana Skirnewskaja, and Timothy D. Wilkinson. Digital pre-distorted one-step phase retrieval algorithm for real-time hologram generation for holographic displays. *Journal of Imaging Science and Technology*, 67(3):030405–1–030405–1, 2023

[6] Jana Skirnewskaja, Yunuen Montelongo, Jinze Sha, Phil Wilkes, and Timothy D. Wilkinson. Accelerated augmented reality holographic 4k video projections based on lidar point clouds for automotive head-up displays. *Advanced Optical Materials*, 12(12):2301772, 2024

[7] Roubing Meng, Jinze Sha, Zhongling Huang, and Timothy D. Wilkinson. Extending FOV of holographic display with alternating lasers. In Peter Schelkens and Tomasz Kozacki,

editors, *Optics, Photonics, and Digital Technologies for Imaging Applications VIII*, volume 12998, page 129981J. International Society for Optics and Photonics, SPIE, 2024

[8] Jinze Sha, Andrew Kadis, Benjamin Wetherfield, Roubing Meng, Zhongling Huang, Dilawer Singh, Antoni Wojcik, and Timothy D. Wilkinson. Information capacity of phase-only computer-generated holograms for holographic displays. In Peter Schelkens and Tomasz Kozacki, editors, *Optics, Photonics, and Digital Technologies for Imaging Applications VIII*, volume 12998, page 129980J. International Society for Optics and Photonics, SPIE, 2024

[9] Jinze Sha, Antoni Wojcik, Benjamin Wetherfield, Jianghan Yu, and Timothy D. Wilkinson. Multi frame holograms batched optimization for binary phase spatial light modulators. *Scientific Reports*, 14(1):19380, Aug 2024

Table of contents

List of figures **xv**

List of tables **xix**

1 Introduction **1**

2 Literature Review **3**

2.1 The Nature of Light . . . 3

2.1.1 Wave-Particle Duality . . . 3

2.1.2 Wave Equation . . . 5

2.2 Fundamentals of Holography . . . 7

2.2.1 Light Source . . . 7

2.2.2 Diffraction . . . 8

2.2.3 Spatial Light Modulator (SLM) . . . 13

2.3 Computer-Generated Holography (CGH) . . . 16

2.3.1 Naive Method . . . 17

2.3.2 Direct Binary Search (DBS) Algorithm . . . 20

2.3.3 Simulated Annealing (SA) Algorithm . . . 23

2.3.4 Gerchberg-Saxton (GS) Algorithm . . . 27

2.3.5 One-Step Phase Retrieval (OSPR) Algorithm . . . 30

2.3.6 Adaptive One-Step Phase Retrieval (AD-OSPR) Algorithm . . . 32

2.3.7 3D CGH . . . 32

2.4 Numerical Optimisation Methods . . . 34

2.4.1 Gradient Descent . . . 35

2.4.2 Newton's Method . . . 35

2.4.3 Quasi-Newton Method: Broyden-Fletcher-Goldfarb-Shanno (BFGS) 36

2.4.4 Large Scale Quasi-Newton Method: Limited Memory BFGS (L-BFGS) 37

3 Gamma Correction in Holographic Projection 39
3.1 Experimental Setup . . . 40
3.2 Determining the Gamma Correction Curve . . . 41
3.3 Applying the Gamma Correction Curve . . . 43
3.4 Summary . . . 46

4 Multi-Depth Phase-Only Hologram Optimization using L-BFGS Algorithm with Sequential Slicing 47
4.1 abstract . . . 47
4.2 Introduction . . . 47
4.3 Method . . . 49
4.3.1 L-BFGS Optimization Algorithm Background . . . 49
4.3.2 Hologram Optimization for Multi-Depth Targets . . . 51
4.4 Results . . . 54
4.4.1 CGH for 4-slice targets . . . 54
4.4.2 CGH for a 30-Slice Target . . . 59
4.5 Conclusion . . . 59

5 Multi Frame Holograms Batched Optimization for Binary Phase Spatial Light Modulators 63
5.1 abstract . . . 63
5.2 Introduction . . . 63
5.3 Method . . . 64
5.4 Results . . . 66
5.4.1 Simulation results . . . 66
5.4.2 Optical Experiment results . . . 68
5.5 Conclusion . . . 75

6 Information capacity of phase-only computer-generated holograms for holographic displays 77
6.1 abstract . . . 77
6.2 Introduction . . . 77
6.3 Methods . . . 78
6.3.1 Computation of Diffraction . . . 78

6.3.2 Computer-Generated Hologram (CGH) Algorithm 79
6.3.3 Measurement of Information 80
6.4 Results 81
6.4.1 Targets at far field (Fraunhofer region) 81
6.4.2 Targets at near field (Fresnel region) 83
6.5 Conclusion 87

References **89**

List of figures

1.1 A photo of the holographic portrait of Dennis Gabor [10] 2

2.1 An illustration of the Young's Double-slit experiment [11] 4
2.2 Coherent v.s. incoherent light 7
2.3 Structure of the first laser [12] 8
2.4 Diffraction geometry 9
2.5 Huygens-Fresnel wavelet principle [13] 10
2.6 Fresnel and Fraunhofer region [14] 11
2.7 Modulation loci in the complex plane [15] 13
2.8 Rotational symmetry in the projection result using the binary phase SLM . 15
2.9 Sample target image of a mandrill (T) [16] 16
2.10 Naive method output 17
2.11 Output of the improved Naive method 18
2.12 Output of the improved Naive method with binary-phase quantisation . . . 19
2.13 DBS algorithm running on the rotationally symmetrical mandrill target . . . 21
2.14 DBS algorithm running on the low resolution target 22
2.15 SA algorithm running on the low resolution target 25
2.16 SA algorithm running on the rotationally symmetrical mandrill target . . . 26
2.17 GS algorithm output on the mandrill target 28
2.18 GS algorithm running on the rotationally symmetrical mandrill target . . . 29
2.19 OSPR algorithm running on the rotationally symmetrical mandrill target . . 31
2.20 Schematic diagram of the intermediate plane method [17] 33

3.1 Optical setup [18] 40
3.2 Measurement of gamma response, which inverse is the correction 41
3.3 Application of the correction curve on the grey-scale ramp 42
3.4 Application of the correction curve on 10-step strips 43

3.5 Application of the correction curve on two sample real-word images 44
3.6 Projection output of the two sample images before and after gamma correction 45

4.1 Loss between the multi-depth targets ($\mathbf{T}_1$ to $\mathbf{T}_n$) and the reconstructions ($\mathbf{R}_1$ to $\mathbf{R}_n$) of hologram $\mathbf{H}$ 51
4.2 Optimization of CGH with sequential slicing (SS) flowchart 53
4.3 Layout of the 4-slice target ($z_1 = 1\ cm$, $z_2 = 2\ cm$, $z_3 = 3\ cm$, $z_4 = 4\ cm$) . . 54
4.4 Final NMSE and run time comparison across the three techniques 55
4.5 Comparison among SS techniques for the 4-slice target using (a) GD algorithm, (b) L-BFGS algorithm, (c) GS algorithm, (d) DCGS algorithm. (e) Average NMSE, and (f) difference between the maximum and minimum NMSE across all slices. 56
4.6 Comparison of final holograms and reconstructions 57
4.7 Layout of the non-binary 4-slice target 58
4.8 Comparison of final holograms and reconstructions for non-binary target . . 58
4.9 30-slice target sliced from a 3D Teapot mesh 59
4.10 Average NMSE v.s. time plot for the 30-slice target 60
4.11 Difference between the maximum and minimum NMSE across all slices v.s. time plot for the 30-slice target 61

5.1 MFHBO flowchart 65
5.2 An example iteration in the optimization process 67
5.3 Convergence of optimization 67
5.4 Holographic projection system components [18] 69
5.5 Simulation and optical reconstruction results for different number of frames 70
5.6 Sample target image - ‘holography’ ambigram 71
5.7 Optical results comparison of the proposed MFHBO method against the existing OSPR and AD-OSPR methods 72
5.8 4-slice target and according reconstruction results 73
5.9 Real-life captured image as target field and their reconstruction results . . . 74

6.1 Gerchberg-Saxton (GS) [19] algorithm flowchart 79
6.2 Quantization of phase holograms 80
6.3 Quantized Gerchberg-Saxton (GS) algorithm flowchart 80
6.4 Del operation on a sample image 81

6.5 Hologram generated at certain bit depths and their according reconstructions at far field 82
6.6 NMSE v.s. Hologram bit depth for target images set at far field 82
6.7 Scatter plot for target images set at far field. (a) NMSE v.s. Entropy. (b) NMSE v.s. Delentropy. 83
6.8 Hologram generated at certain bit depths and their according reconstructions at near field 84
6.9 NMSE v.s. Hologram bit depth for target images set at near field 84
6.10 Scatter plot for target images set at near field. (a) NMSE v.s. Entropy. (b) NMSE v.s. Delentropy. 85
6.11 Hologram entropy and NMSE v.s. iteration number for target images set at near field, with the initial phase ($\angle A$) being (a) zeros (b) random 86

List of tables

3.1 Gamma response results before and after gamma correction 43
3.2 Gamma correction results for sample images 46

5.1 MFHBO runtime (s) 68
5.2 Quantitative analysis of the optical results in Fig. 5.7 71
5.3 Quantitative analysis of the optical results in Fig. 5.8 73
5.4 Quantitative analysis of the optical results in Fig. 5.9 75

Chapter 1

Introduction

The pursuit of three-dimensional (3D) display has never stopped. Currently, most commercially available so-called '3D display' products such as 3D cinema, 3D TV, handheld 3D devices (e.g. Nintendo 3DS, HTC Evo 3D) and Virtual Reality (VR) and Augmented Reality (AR) head sets are in fact stereoscopic displays where two different two-dimensional (2D) images are displayed to the left and right eye respectively, creating a 3D illusion in the brain. Despite its high image quality, the major issue with stereoscopic displays is that they cannot provide real defocusing effect in depth. Modern 3D cinema are able to provide good comfort because the polarisation glasses are as light as regular glasses, and the variable defocusing issue can be avoided by the combination of good design of point of interest in each scene and the according defocusing effect as captured by the camera, so most audience won't experience much discomfort for around 2 to 3 hours. But the content, viewing angle and depth of focus are fixed at how they are captured. To provide an interactive and real-time rendered immersive experience, the VR/AR headset has frequently been advertised as the 'gateway to metaverse' in recent years. However, my personal experience with VR headset is far from comfortable, not only because of its heavy weight, but also because the display is physically at a very near distance, while my brain thinks the objects are at various distances and yet are still all in focus, which is very unnatural, because in real life, when the eye is focused on a near object, the far backgrounds would blur out. And also, the two displays in the VR headset needs to be rendered in real-time based on the location and angle of the user, which is nowhere near practical. Hence, the heavy weight, the lack of defocusing, the delay between the rendering and the change in my position are the three major factors causing my dizziness using VR headsets, either of which is quite impractical to solve, especially the

weight issue. Only if VR/AR headsets could be reduced to the weight of eyeglasses would I ever consider the possibility of those head-mounted devices leading us to the 'metaverse'.

In comparison, the holography technique can produce the full 3D light field, which does not rely on any head mounted device, has the true depth of focus, and does not need to re-render according to change in viewer positions and viewing angle.

Fig. 1.1 A photo of the holographic portrait of Dennis Gabor [10]

Holography, taking its name from the Greek word $o\lambda o\sigma$ (holos), meaning *whole*, was first introduced in 1948 by Dennis Gabor [20], originally named as *wavefront reconstruction* [21]. It is a cool technology which generates 3D images via the diffraction of light. Similar to 2D photography, the earliest holography uses a piece of film to record the diffraction pattern, which can then reconstruct the 3D field, as shown in Fig. 1.1 which is a holographic recording of Dennis Gabor himself. After the invention of digital cameras, digital holography emerged. The limitation of both methods is that they require a physical object as a priori to record the hologram. In order to generate hologram for objects that do not physically exist, computer-generated holography (CGH) emerged where a hologram can be calculated through various algorithmic approaches and then displayed on a spatial light modulator (SLM) in order to create an image in the replay field through diffraction [15, 22, 23, 19]. Although being a fancy technology of true 3D display, CGH still has some fundamental issues, mainly its image quality and the heavy computation required, the solutions of which are my ultimate goals.

Chapter 2

Literature Review

This chapter lays down the fundamental theories of optoelectronics and search algorithms, which are essential to the research outlined in the later chapters of this thesis.

2.1 The Nature of Light

2.1.1 Wave-Particle Duality

The problem of how light propagates has been troubling scientists for centuries. The journey began with Sir Isaac Newton, who advocated the particle theory of light in the 17th century. Newton proposed that light consists of particles, or 'corpuscles', which explained phenomena such as reflection and refraction [24]. In contrast, Christiaan Huygens, a contemporary of Newton, demonstrated that light behaves as a wave, as it is capable of diffraction and interference [25].

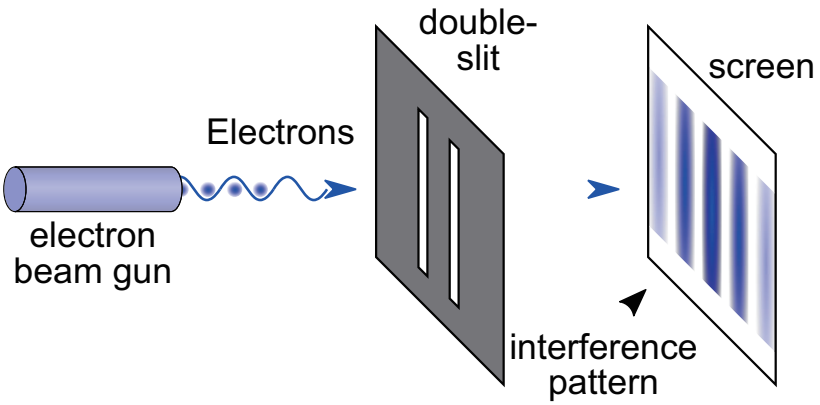


Fig. 2.1 An illustration of the Young's Double-slit experiment [11]

The wave theory gained significant support in the early 19th century through the experiments of Thomas Young. Young's double-slit experiment in 1801 provided clear evidence of the wave nature of light by showing that light passing through two slits creates an interference pattern on a screen [26], as illustrated in Fig. 2.1. Augustin-Jean Fresnel further advanced the wave theory by developing a comprehensive mathematical framework to describe light as a wave, explaining phenomena such as polarization and the diffraction of light [27].

This understanding was further reinforced by James Clerk Maxwell in the mid-19th century. In 1864, James Clerk Maxwell organised a set of four equations describing the space and time dependence of the electromagnetic field, which are:

$$\nabla \times \mathbb{E} = -\frac{\partial \mathbb{B}}{\partial t} \tag{2.1}$$

$$\nabla \times \mathbb{H} = \mathbb{J} + \frac{\partial \mathbb{D}}{\partial t} \tag{2.2}$$

$$\nabla \cdot \mathbb{D} = \rho \tag{2.3}$$

$$\nabla \cdot \mathbb{B} = 0 \tag{2.4}$$

where $\mathbb{D}$ is the electric flux density, $\mathbb{E}$ is the electric field intensity, $\mathbb{B}$ is the magnetic flux density, $\mathbb{H}$ is the magnetic field intensity, ρ is the volume charge density, and $\mathbb{J}$ is the current density [28].

And the relation between $\mathbb{D}$ and $\mathbb{E}$ and between $\mathbb{B}$ and $\mathbb{H}$ for linear materials are:

$$\mathbb{B} = \mu\mathbb{H} \tag{2.5}$$

$$\mathbb{D} = \varepsilon\mathbb{E} \tag{2.6}$$

where μ is the magnetic permeability and ε is the dielectric permittivity of the material [29].

Maxwell's equations unified electricity and magnetism into a single theory of electromagnetism, predicting that light is an electromagnetic wave that propagates through space [30].

Despite the success of the wave theory, it could not explain all light-related phenomena. The early 20th century brought a pivotal development with Albert Einstein's explanation of the photoelectric effect. In 1905, Einstein proposed that light also behaves as particles, or 'quanta' (later called photons), which could eject electrons from a metal surface when light is shone upon it [31]. This particle nature of light was critical in explaining observations that wave theory alone could not address and earned Einstein the Nobel Prize in Physics in 1921.

These discoveries collectively revealed that light exhibits both wave and particle properties, depending on the experimental context. This wave-particle duality became a cornerstone of quantum mechanics, fundamentally altering human's understanding of the nature of light. Although to date, it is still not yet known what light exactly is, it is now known how light behaves.

2.1.2 Wave Equation

To mathematically describe the propagation of light in free space (i.e. in absence of free charge), the Maxwell equations in Eq. (2.1) - Eq. (2.4) can be simplified as:

$$\nabla \times \mathbb{E} = -\mu \frac{\partial \mathbb{H}}{\partial t} \tag{2.7}$$

$$\nabla \times \mathbb{H} = \varepsilon \frac{\partial \mathbb{E}}{\partial t} \tag{2.8}$$

$$\nabla \cdot \varepsilon\mathbb{E} = 0 \tag{2.9}$$

$$\nabla \cdot \mu\mathbb{H} = 0 \tag{2.10}$$

Taking the curl on both the left and right hand sides of Eq. (2.7), and using the vector identity of $\nabla \times (\nabla \times \mathbf{u}) = \nabla(\nabla \cdot \mathbf{u}) - \nabla^2 \mathbf{u}$, we get:

$$\nabla \times (\nabla \times \mathbb{E}) = -\nabla \times (\mu \frac{\partial \mathbb{H}}{\partial t}) \tag{2.11}$$

$$\nabla(\nabla \cdot \mathbb{E}) - \nabla^2 \mathbb{E} = -\frac{\partial}{\partial t} \nabla \times (\mu \mathbb{H}) \tag{2.12}$$

Then, by substituting Eq. (2.8) and Eq. (2.9) in, Eq. (2.12) becomes:

$$-\nabla^2 \mathbb{E} = -\frac{\partial}{\partial t}(\mu \varepsilon \frac{\partial \mathbb{E}}{\partial t}) \tag{2.13}$$

Hence, we have a generic form of wave equation, relating the space and time domain relation of electromagnetic waves propagating in free space:

$$\nabla^2 \mathbb{E} = \mu \varepsilon \frac{\partial^2 \mathbb{E}}{\partial t^2} \tag{2.14}$$

A valid solution to Eq. (2.14) is:

$$\mathbb{E} = \mathbb{E}_0 e^{j(\omega t - kr)} \tag{2.15}$$

where ω is the angular velocity of the wave, t is time, r is the propagation distance and k is called the wave number ($k = \frac{2\pi}{\lambda}$, where λ is the wavelength). From Eq. (2.15) we can see that the propagation of light in free space is essentially a phase shift. This suggests that, if we have a coherent light source and a device to manipulate light (called SLM, further explained in Section 2.2.3), we can produce an interference pattern reconstructing the target field we desire, and such method is called holographic projection.

2.2 Fundamentals of Holography

Holography is a technology that can fully reconstruct the wavefront of 3D objects, which is usually achieved by modulating a coherent light source. This section explains what a coherent light source is and how it is modulated and diffracted.

2.2.1 Light Source

The mechanism of holographic projection is to control the propagation of light in a way that, after diffraction, reconstructs a wavefront that matches the target field. We usually prefer to start from a coherent light source rather than a random one which will be a lot more difficult or even impossible to analyse and predict the interference pattern.

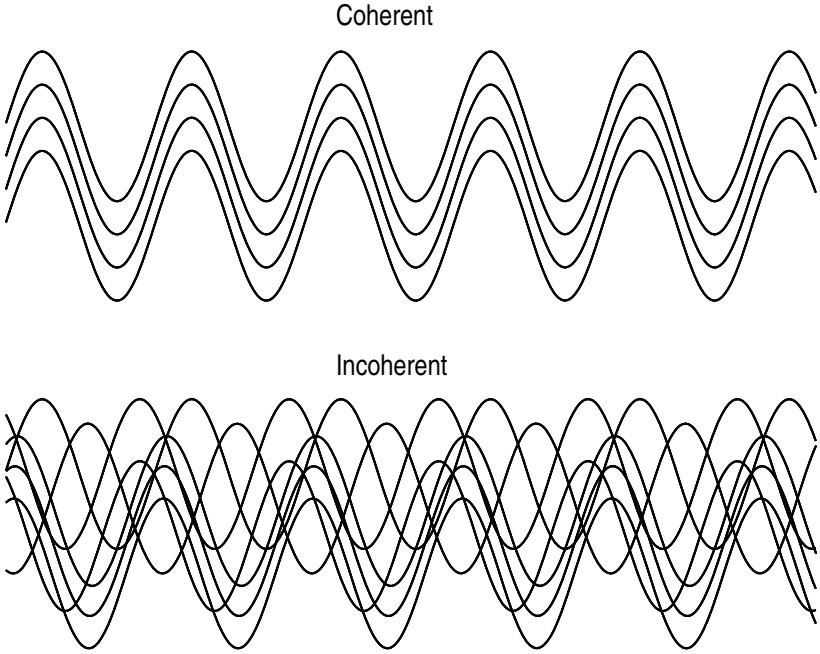


Fig. 2.2 Coherent v.s. incoherent light

The coherence of light refers to the property of light waves where the phase relationship between the waves is consistent over time and space, corresponding to temporal and spatial coherence:

- **Temporal coherence:** Temporal coherence describes the correlation between the phases of a light wave at different points along its propagation direction. It indicates how monochromatic (i.e. single-frequency) a light source is.

- **Spatial coherence:** Spatial coherence describes the correlation between the phases of a light wave at different points across the wavefront, perpendicular to the direction of propagation. It indicates the uniformity of the phase across the wavefront, as illustrated in Fig. 2.2. High spatial coherence means that the light waves across different points on the wavefront are in phase.

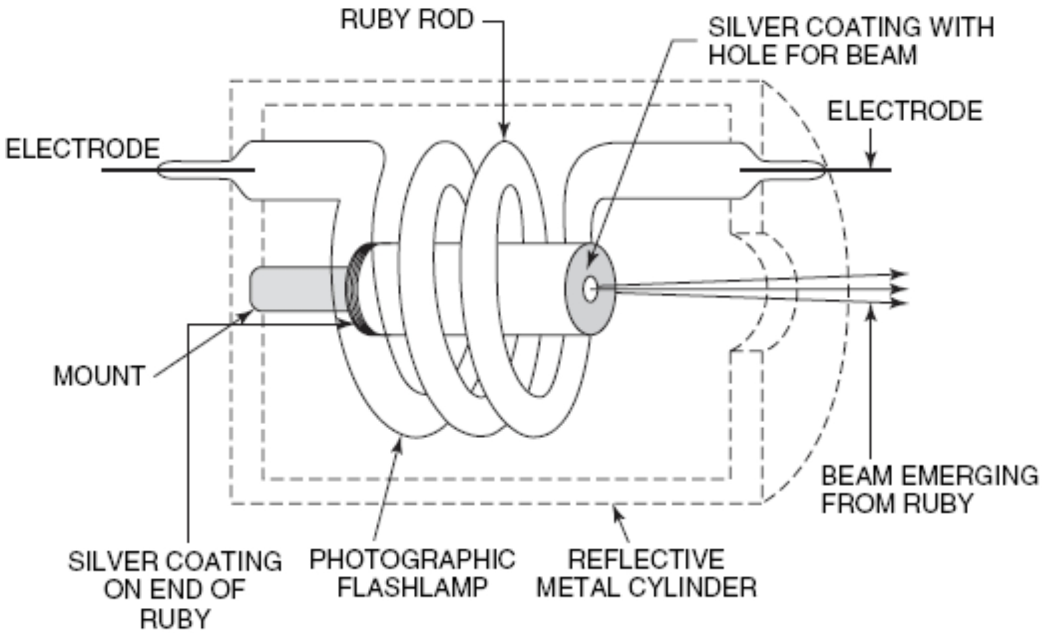


Fig. 2.3 Structure of the first laser [12]

The most common coherent light source is Laser, which stands for ***L**ight **A**mplification by the **S**timulated **E**mission of **R**adiation*. It was first invented by Theodore Maiman in 1959, with the structure shown in Fig. 2.3 [12, 32, 33]. It differs from other sources of light in that it emits coherent light, which is suitable for holographic projection. However, the coherent and monochromatic property of laser also has a side effect of speckle noise in the reconstructed image [34], which is one of the major problems affecting the image quality of holographic projections and has seen lots of efforts to cope with it in the literature [35–38].

2.2.2 Diffraction

This section delves into how light interacts with apertures, leading to diffractions. Understanding diffraction is essential for holography, as it explains how light can be manipulated to reconstruct three-dimensional light fields. The principles of diffraction and interference

underpin the essential process of holographic projection, making it possible to accurately recreate complex wavefronts and achieve true 3D visualization.

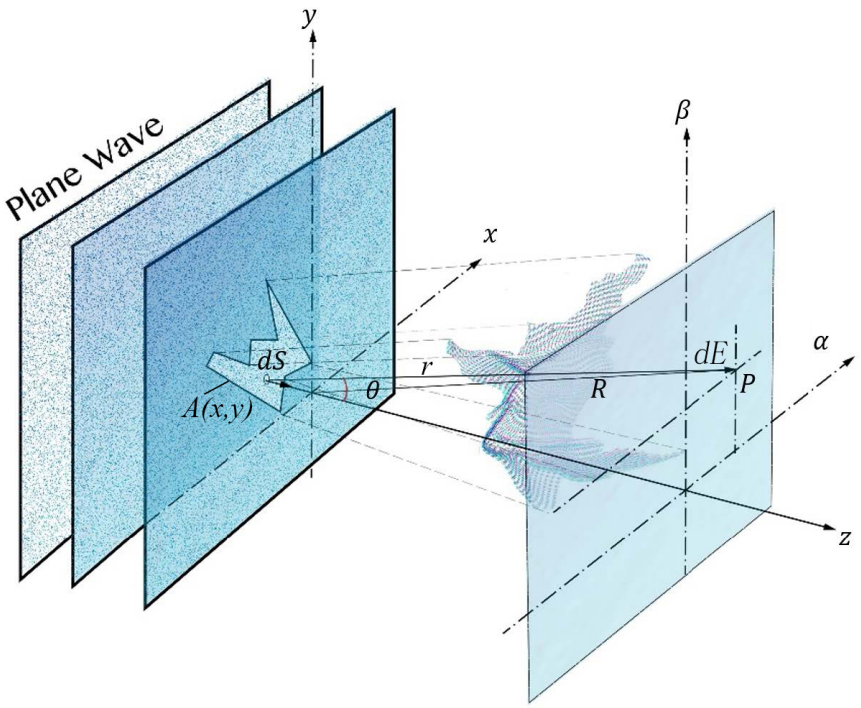


Fig. 2.4 Diffraction geometry

To model how light diffracts through a 2D aperture, we first set up a coordinate system as shown in Fig. 2.4, where the aperture is denoted by $A(x,y)$ and the diffracted field is denoted by $E(\alpha,\beta,z)$. R defines the distance between point P and the origin of the aperture ($(x,y) = (0,0)$), r defines the distance between point P and a point on the aperture, and θ defines the angle r from the z-axis. Then by trigonometry we can have the following identities:

$$cos(\theta) = \frac{z}{r} \tag{2.16}$$

$$R^2 = \alpha^2 + \beta^2 + z^2 \tag{2.17}$$

$$r^2 = (\alpha - x)^2 + (\beta - y)^2 + z^2 \tag{2.18}$$

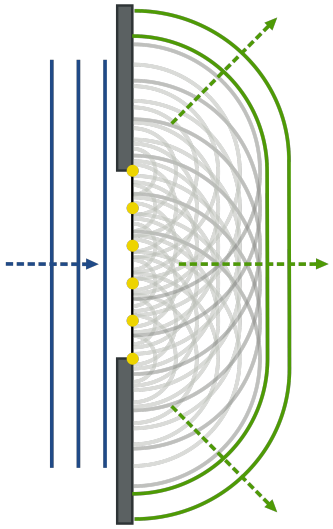


Fig. 2.5 Huygens-Fresnel wavelet principle [13]

The Huygens-Fresnel principle states that every point on a wavefront is itself the source of outgoing secondary spherical wavelets, which can be expressed mathematically as follows when $r \gg \lambda$ [39]:

$$E(\alpha,\beta,z) = \frac{1}{j\lambda}\iint A(x,y)\frac{e^{jkr}}{r}cos(\theta)dxdy \tag{2.19}$$

Applying the identities in Eq. (2.16) - Eq. (2.18), Eq. (2.19) becomes:

$$E(\alpha,\beta,z) = \frac{z}{j\lambda}\iint A(x,y)\frac{e^{jkr}}{r^2}dxdy \tag{2.20}$$

$$= \frac{z}{j\lambda}\iint A(x,y)\frac{e^{jk\sqrt{(\alpha-x)^2+(\beta-y)^2+z^2}}}{(\alpha-x)^2+(\beta-y)^2+z^2}dxdy \tag{2.21}$$

Unfortunately, Eq. (2.21) cannot be solved analytically except for few specific aperture functions $A(x,y)$, so we have to make some approximations in order to solve for arbitrary $A(x,y)$, the common methods are *Fresnel* and *Fraunhofer* approximations for regions depicted in Fig. 2.6.

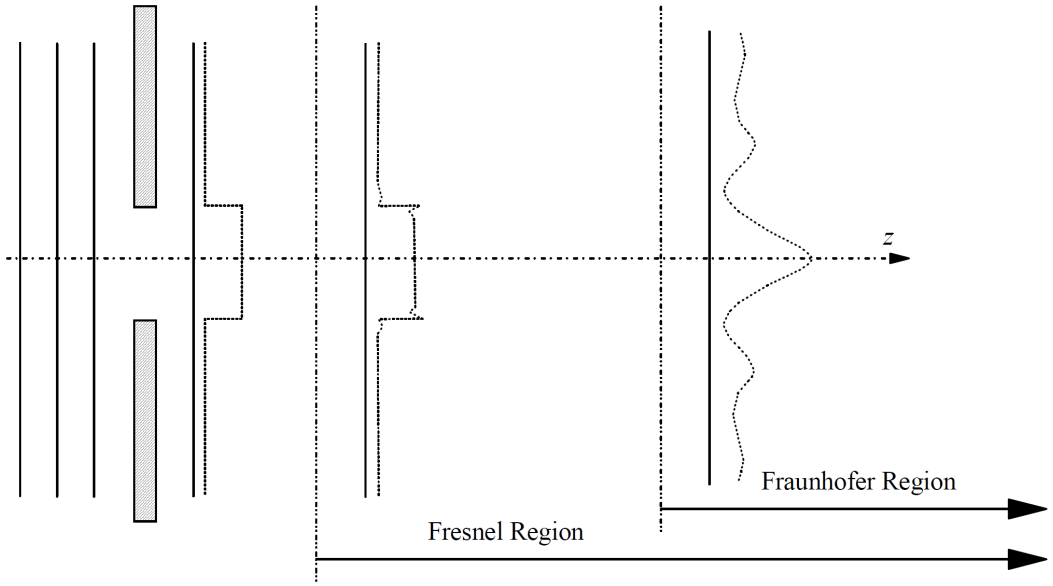


Fig. 2.6 Fresnel and Fraunhofer region [14]

Fresnel Approximation

$$\sqrt{1+d} = 1 + \frac{1}{2}d - \frac{1}{8}d^2 + \cdots \tag{2.22}$$

Fresnel approximation replaces expressions for spherical waves by quadratic-phase exponentials, using the binomial expansion of the square root (given in Eq. (2.22)) to approximate r in Eq. (2.20) [39].

Retaining only the first two terms of the expansion gives:

$$r = \sqrt{(\alpha - x)^2 + (\beta - y)^2 + z^2} \tag{2.23}$$

$$= z\sqrt{1 + \left(\frac{\alpha - x}{z}\right)^2 + \left(\frac{\beta - y}{z}\right)^2} \tag{2.24}$$

$$\approx z\left[1 + \frac{1}{2}\left(\frac{\alpha - x}{z}\right)^2 + \frac{1}{2}\left(\frac{\beta - y}{z}\right)^2\right] \tag{2.25}$$

For the r^2 in the denominator of Eq. (2.20), the error introduced by dropping all terms but z is generally acceptably small (i.e. $r^2 \approx z^2$), and for the r appearing in the exponent in the numerator of Eq. (2.20), errors are much more critical [39]. So, by substituting Eq. (2.25) for

the r in the numerator of Eq. (2.20) and substituting z for the r in the denominator, we have:

$$E(\alpha,\beta,z) \approx \frac{z}{j\lambda}\iint A(x,y)\frac{e^{jkz\left[1+\frac{1}{2}\left(\frac{\alpha-x}{z}\right)^2+\frac{1}{2}\left(\frac{\beta-y}{z}\right)^2\right]}}{z^2}dxdy \tag{2.26}$$

$$= \frac{e^{jkz}}{j\lambda z}e^{j\frac{k}{2z}(\alpha^2+\beta^2)}\iint\left\{A(x,y)e^{j\frac{k}{2z}(x^2+y^2)}\right\}e^{-j\frac{2\pi}{\lambda z}(\alpha x+\beta y)}dxdy \tag{2.27}$$

$$= \frac{e^{jkz}}{j\lambda z}e^{j\frac{k}{2z}(\alpha^2+\beta^2)}\mathscr{F}\left\{A(x,y)e^{j\frac{k}{2z}(x^2+y^2)}\right\} \tag{2.28}$$

where $\mathscr{F}$ is the Fourier Transform. Such method of including Fourier Transform (FT) in the study of optics is also named 'Fourier Optics'.

Now we have a more simple and solvable expression than Eq. (2.21). And also, as we are only interested in the scaling of relative points at P with respect to each other, so it is safe to normalise the multiplier term before the Fourier Transform to 1 [14]. So we can express the diffraction pattern in Fresnel region as:

$$E_{Fresnel\ region}(\alpha,\beta,z) = \mathscr{F}\left\{A(x,y)e^{j\frac{k}{2z}(x^2+y^2)}\right\} \tag{2.29}$$

Fraunhofer Approximation

Fraunhofer diffraction is a form of diffraction in which the distance between the light source and the receiving screen are in effect at infinite, so that the wave fronts can be treated as planar rather than spherical [28]. Fraunhofer approximation is very stringent, it assumes that the distance between the light source and the receiving screen are in effect at infinite:

$$z \gg \frac{k(x^2+y^2)_{max}}{2} \tag{2.30}$$

so that the wave fronts can be treated as planar rather than spherical [28], then the $e^{j\frac{k}{2z}(x^2+y^2)}$ term tends to 1, and Eq. (6.1) becomes:

$$E_{Fraunhofer\ region}(\alpha,\beta) = \mathscr{F}\{A(x,y)\} \tag{2.31}$$

which suggests that the far field pattern is simply the Fourier Transform of the aperture function.

2.2.3 Spatial Light Modulator (SLM)

SLMs are critical components in computer-generated holography (CGH). SLM is a device used to control the amplitude or phase of light waves in a spatially varying manner. SLMs typically consist of a two-dimensional(2D) array of pixels, each of which can modulate the light either passing through or reflected from it. These pixels are usually addressed by electronic signals, allowing precise manipulation of the light wavefront.

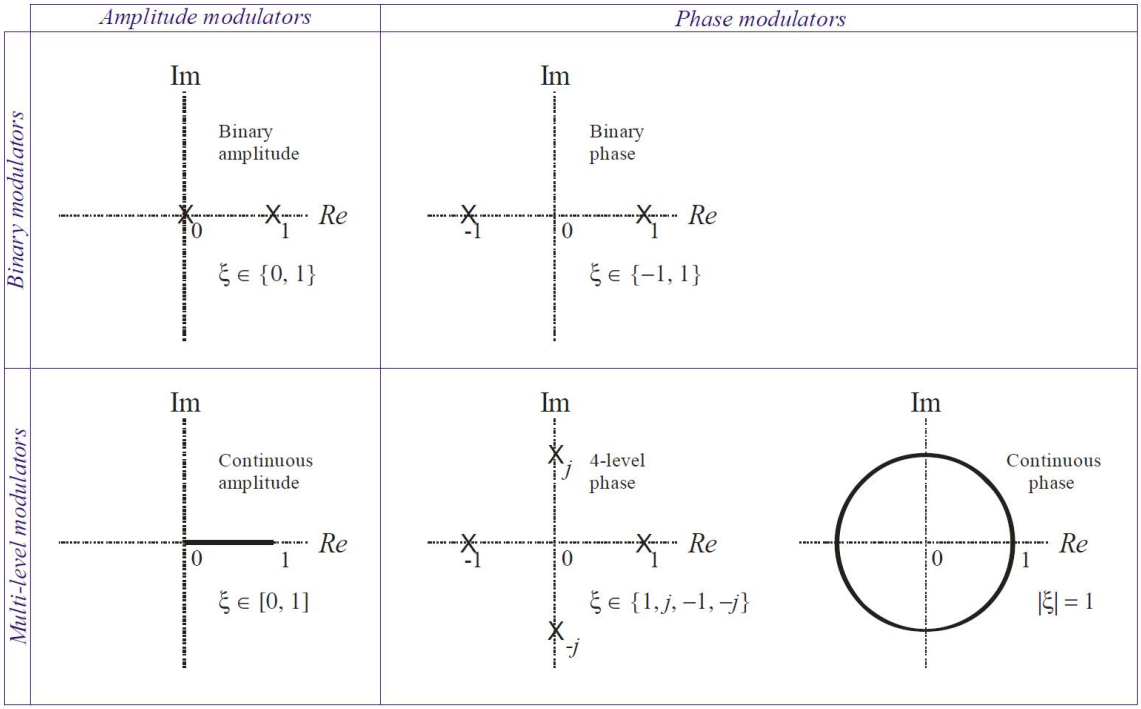


Fig. 2.7 Modulation loci in the complex plane [15]

The modulation can be achieved through various mechanisms, such as liquid crystal SLMs(LC-SLM), magneto-optic SLMs, deformable mirror SLMs, multiple-quantum-well SLMs, or acousto-optic Bragg cells [39], which all fall into the four modulation categories, as illustrated in Fig. 2.7 [15]. The four modulation schemes are:

- **Multi-level Amplitude** modulators can modulate each pixel from zero transmission (0) to full transmission (1), either continuously or in discrete steps. (e.g. nematic liquid crystal display [40], found for example in laptops and many conventional video projectors)

- **Binary Amplitude** modulators can switch each pixel to zero transmission (0) or full transmission (1), but nothing in between. (e.g. deformable mirror device [41], ferroelectric liquid crystal display [42], both used in high-end video projectors)
- **Multi-level Phase** modulators can modulate the phase shift imparted by each pixel from 0 to 2π radians, either continuously or in discrete steps. (e.g. Nematic liquid crystal devices [43])
- **Binary Phase** modulators can switch each pixel for a phase shift of either 0 or π radians. (e.g. Ferroelectric liquid crystal displays [44])

Among the four modulation schemes, phase modulations are of higher interests for the purpose of holography, because amplitude modulators, either multi-level or binary, blocks light at the SLM, causing waste of energy, leading to poorer energy efficiency. And also, amplitude modulations always have a zero-order (forming a central bright spot), because the average amplitude is always between 0 and 1; on contrary, phase modulation can suppress the zero-order by designing the hologram to have zero average.

As there is no complex modulator available yet, we need algorithms to generate phase-only holograms, such process is called phase retrieval. There are currently many algorithms for such purpose, which will be discussed in Section 2.3.

Rotational symmetries in the binary phase modulation

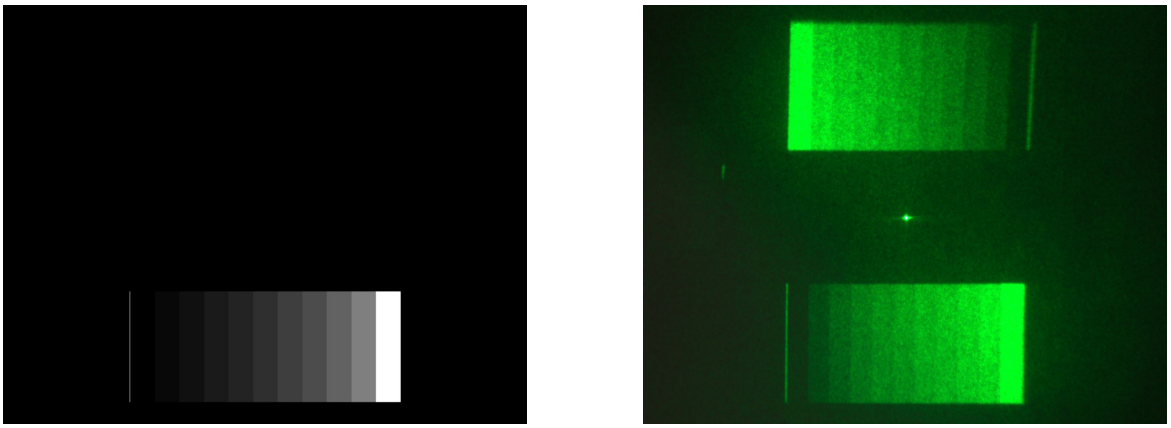
The spatial light modulator (SLM) used in this thesis is a binary phase modulator. As the binary phase modulation is purely real (i.e. it's only switching between 0° and 180°, corresponding to 1 and -1 values of $A^*(x,y)$), the complex conjugate $A^*(x,y)$ is the same as $A(x,y)$:

$$A^*(x,y) = A(x,y) \tag{2.32}$$

because the Fourier transform of $A^*(x,y)$ is the same as the Fourier transform of $A(x,y)$

$$E(-\alpha,-\beta) = \mathscr{F}[A^*(x,y)] = \mathscr{F}[A(x,y)] = E(\alpha,\beta) \tag{2.33}$$

So, at the Fraunhofer region, there is no distinction between the desired image and its 180° rotation in the replay field, causing a symmetrical conjugate image rotated 180° from the target image.



(a) Target image (b) Projection result

Fig. 2.8 Rotational symmetry in the projection result using the binary phase SLM

To demonstrate such phenomena, the example target image shown in Fig. 2.8a ran through the binary-phase hologram generation algorithm called one-step phase-retrieval (OSPR), which will be explained in detail in Section 2.3.5, and the projection output is shown in Fig. 2.8b. The simplest workaround for this issue is to use only half of the reconstruction field, like the target image in Fig. 2.8a. However, the side effect of this workaround is that half of the energy will be wasted, leading to higher power consumption and heat dissipation.

2.3 Computer-Generated Holography (CGH)

Different from the traditional analogue optical holography and digital holography which both need a physically existing object to record the hologram, CGH is a method to use computer algorithms to generate holograms without the need for the physical target object. Ideally, holograms can be simply computed using the inverse Fourier Transform, using the inverse functions of Eq. (6.1) and Eq. (6.2) derived in Section 2.2.2. The inverse Fourier Transforms on most targets will end up with results in complex numbers; however, as mentioned in Section 2.2.3, currently available SLMs cannot achieve complex modulation yet. Therefore, computer algorithms are needed to compute either phase-only or amplitude-only holograms, among which the former is preferred, as explained in Section 2.2.3.

This section reviews the existing methods in the literature for calculating phase-only holograms. The phase-only hologram is labelled as H, so that it differs from the previous notation of A for complex-valued hologram apertures. And the propagation function is unified as $\mathscr{P}$, which can be either the Fraunhofer propagation equation in Eq. (6.1) or the Fresnel propagation equation in Eq. (6.2) depending on the distance of the target field. Lastly, R denotes the reconstruction from the hologram, which is the amplitude of the result from $\mathscr{P}$.

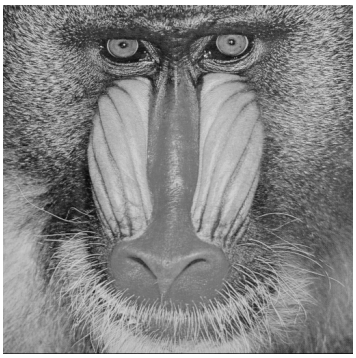


Fig. 2.9 Sample target image of a mandrill (T) [16]

A sample image of a mandrill (shown in Fig. 2.9) is chosen from the University of Southern California Signal and Image Processing Institute (USC-SIPI) Image Database [16] to test and compare the classical phase retrieval algorithms in the literature. As the square of the amplitude, also known as the 'intensity' of light, is the only visible component with human eyes [45], the diffracted electric field's amplitude ($|E|$) will be targeted to match the square root of T.

2.3.1 Naive Method

The simplest method to get a phase-only hologram H is by directly extracting the phase of the reverse propagation from the target field, discarding the amplitude component (e.g. for Fraunhofer propagation, H will simply be the phase of the inverse Fourier transform $\mathscr{F}^{-1}$ of the target field). This method is named as 'Naive' method in this thesis for its simplicity. The pseudocode of the Naive method is shown in Algorithm 1 below:

Algorithm 1 Naive method

Input: Target image T, Propagation function $\mathscr{P}$ (e.g. Fresnel or Fraunhofer propagation)

Output: Phase hologram H and its reconstruction intensity R

$E \leftarrow \sqrt{T}$

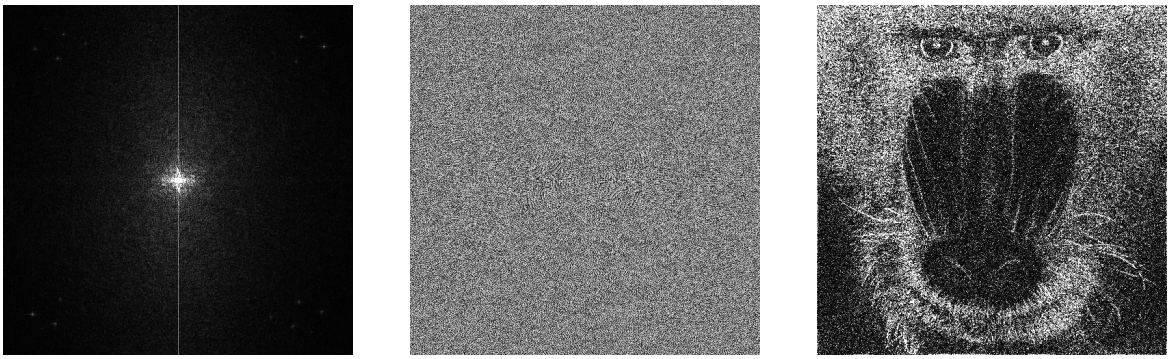
$A \leftarrow \mathscr{P}^{-1}[E]$

$H \leftarrow \angle A$

$R \leftarrow |\mathscr{P}[e^{jH}]|^2$

where $j = \sqrt{-1}$, the $\angle$ sign means phase extraction (i.e. arguments of complex numbers element-wise in the matrix), and e^{jH} converts the angles back to complex numbers. All exponentials, modulus and square-root operators are carried out in an element-wise manner, so that the dimensions of T, A, H, R and E are all the same.

The Naive method (described in Algorithm 1) was then implemented in MATLAB and ran on the sample target image shown in Fig. 2.9, with the distance set to infinity (i.e. in the Fraunhofer region using the propagation formula Eq. (6.2)). The results are shown in Fig. 2.10 below:



(a) Discarded amplitude ($|A|$) (b) Phase Hologram (H) (c) Reconstruction (R)

Fig. 2.10 Naive method output

After taking the inverse Fourier Transform, the amplitude and the phase of A are shown in Fig. 2.10a and Fig. 2.10b respectively. The next step then discards the amplitude component (Fig. 2.10a) and uses the phase component (in Fig. 2.10b) as the phase hologram H. Then the phase hologram went through a forward propagation and the resulting reconstruction intensity is shown in Fig. 2.10.

The reconstruction intensity is very far from the desired target image in Fig. 2.9. It shows that, discarding amplitude has introduced a significant loss of information. From a signal processing point of view, the peak around the centre in Fig. 2.10a corresponds to low spatial frequency signals, and discarding them causes the reconstruction in Fig. 2.10c to lose low frequency components and effectively becomes an edge detector. Another explanation of the poor reconstruction quality is that, this method is assuming a uniform phase profile for the target image, which is physically difficult to achieve. A simple improvement can be made by adding a random phase to the target, as shown in the pseudocode below:

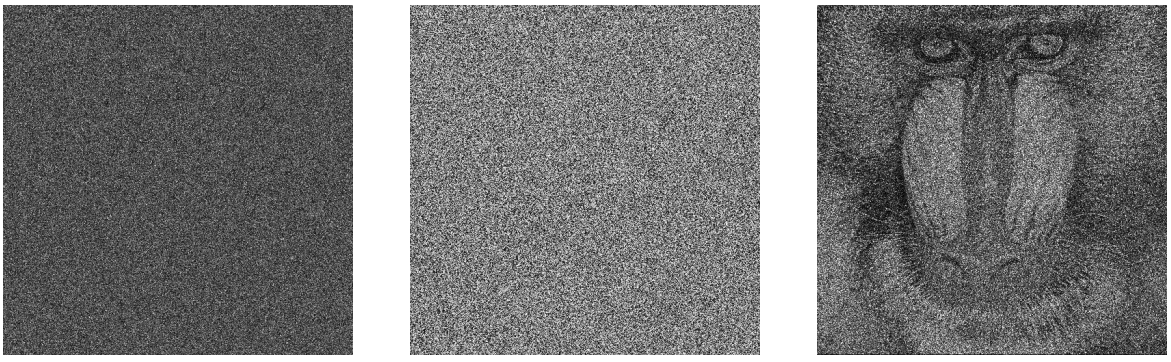
Algorithm 2 Improved Naive method with random phase added to the target field

Input: Target image T, Propagation function $\mathscr{P}$ (e.g. Fresnel or Fraunhofer propagation)

Output: Phase hologram H and its reconstruction intensity R

$E \leftarrow \sqrt{T} * RandomPhase()$
$A \leftarrow \mathscr{P}^{-1}[E]$
$H \leftarrow \angle A$
$R \leftarrow |\mathscr{P}[e^{jH}]|^2$

The improved Naive method was implemented in MATLAB and produced the results in Fig. 2.11 below:

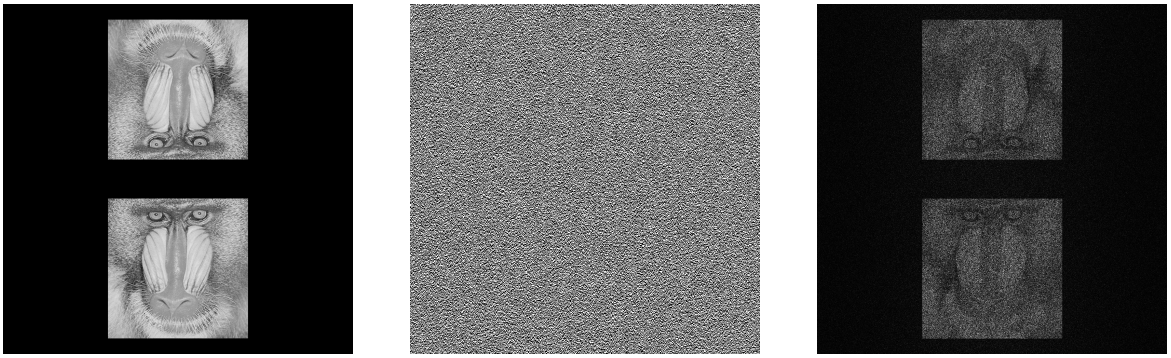


(a) Discarded amplitude ($|A|$) (b) Phase Hologram (H) (c) Reconstruction (R)

Fig. 2.11 Output of the improved Naive method

As shown in Fig. 2.11c, the reconstruction quality has been greatly improved, although still quite noisy. The amplitude of the hologram being discarded (shown in Fig. 2.11a) is a lot more uniformly distributed than the one in Fig. 2.10a, so that the loss of information evenly spread across all spatial frequencies, leading to the much better reconstruction quality. However, the reconstruction quality is still quite noisy, with a normalised mean squared error (NMSE) of 1.0228×10^{-6} and a structural similarity index (SSIM [46]) of 0.1603.

Moreover, additional error will be introduced during the quantisation step, which is necessary for the phase hologram to be displayed on SLMs with limited bit depth. As the SLM used in this thesis is a binary phase SLM, which has a rotational symmetry property as explained in Section 2.2.3, a target image is specifically designed as shown in Fig. 2.12a which is rotational symmetrical by itself.



(a) Target image for the binary-phase SLM

(b) Binary-phase Hologram

(c) Reconstruction

Fig. 2.12 Output of the improved Naive method with binary-phase quantisation

The binary phase hologram in Fig. 2.12b is generated by adding an additional binary quantisation step ($\mathscr{Q}$) on the phase hologram computed using Algorithm 2, which is simply rounding all phases to 0 rad and π rad. The reconstruction of the binary-phase hologram is shown in Fig. 2.12c, which is of poor quality with NMSE being 4.5452×10^{-7} and SSIM being 0.0603. The only advantage of this method is its speed, as it only requires one inverse Fourier Transform calculation. To improve the reconstruction quality, better algorithms are needed. The following sections explores predecessors' efforts in quality improvement.

2.3.2 Direct Binary Search (DBS) Algorithm

Direct Binary Search (DBS) algorithm [22] is an algorithm that generates the hologram by randomly flipping each pixel in the SLM between binary states (0 and π), one by one for many times in order to minimise the difference between its reconstruction intensity R and the target image T. The detailed algorithm is described in Algorithm 3 below:

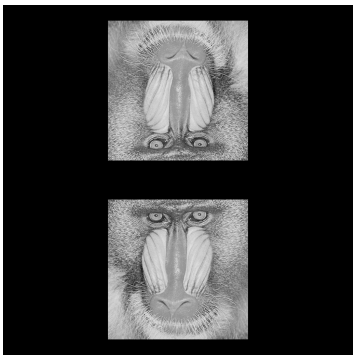
Algorithm 3 Direct Binary Search (DBS) algorithm

Input: Target image T, Propagation function $\mathscr{P}$, Loss function $\mathscr{L}$ (e.g. mean-squared error), Number of iterations N

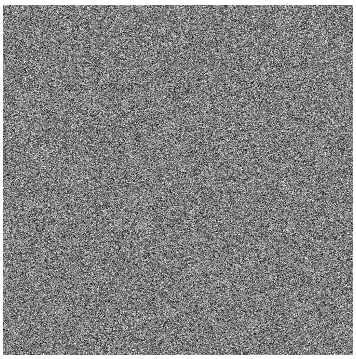
Output: Phase hologram H and its reconstruction intensity R

```
// Start with a random hologram with a size matching T
H ← Rand(Size(T))
R ← |𝒫[e^{jH}]|^2
L ← ℒ[R, T]
for n = 1 to N do
    // Flip a random pixel in the hologram
    H_n ← FlipRandomPixel(H)

    // Calculate the loss function for the new hologram
    R_n ← |𝒫[e^{jH_n}]|^2
    L_n ← ℒ[R_n, T]

    // Compare the new loss with the old one
    if L_n < L then
        // Accept the new hologram if loss is lower
        H ← H_n
        R ← R_n
        L ← L_n
    end if
end for
```

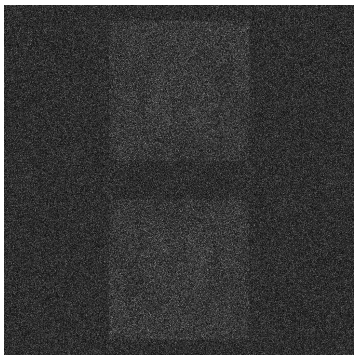
Although the DBS algorithm is specifically suited for generating binary phase holograms, it can also be adapted for generating multi-level phase holograms, by representing each level as binary numbers, at the cost of more computation.



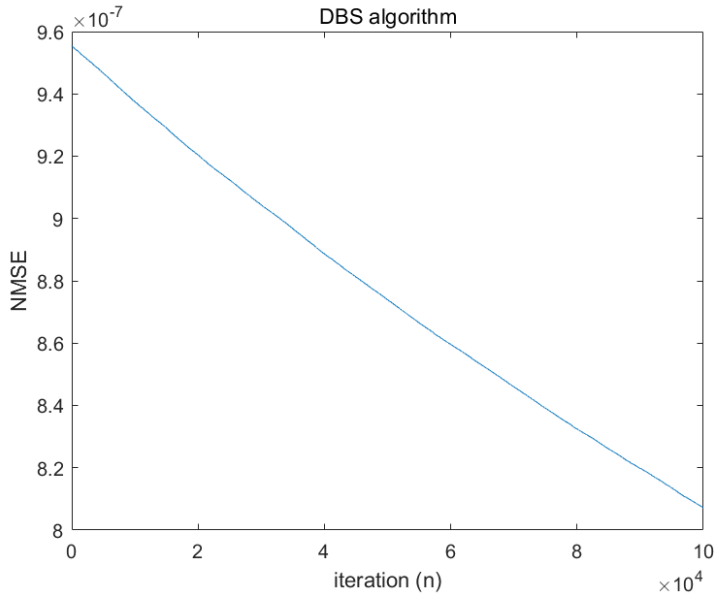
(a) Target image ($1024px \times 1024px$)



(b) Binary-phase Hologram



(c) Reconstruction



(d) NMSE v.s. iteration plot

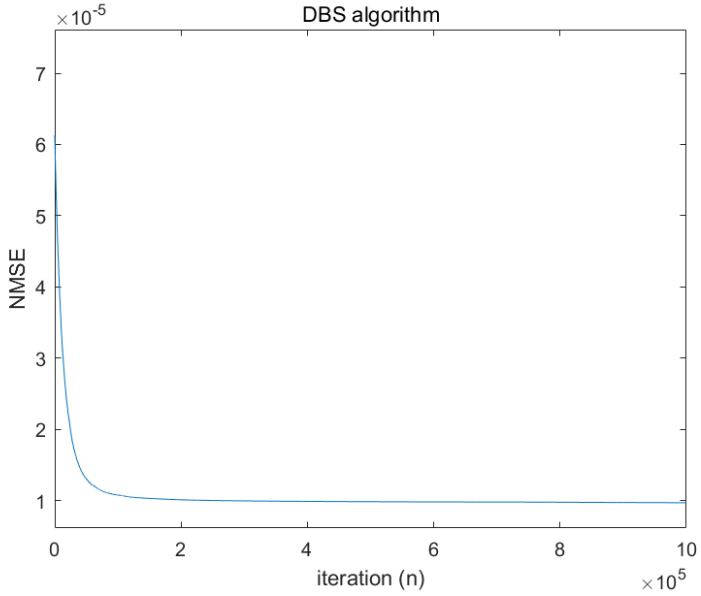
Fig. 2.13 DBS algorithm running on the rotationally symmetrical mandrill target

DBS algorithm can sometimes find very accurate hologram if the run is lucky; however, it is extremely slow, because it takes numerous iterations (as shown in Fig. 2.13d, even 10^5 iterations has not reached a good convergence) and each iteration requires a Fourier Transform which is computationally expensive, the example run on the target image of resolution $1024px \times 1024px$ in Fig. 2.13a took more than one hour to run the 10^5 iterations, which is still nowhere near convergence. The binary-phase hologram produced is shown in Fig. 2.13b and its corresponding reconstruction in Fig. 2.13c is of very poor quality.

(a) Target image ($128px \times 128px$)

(b) Binary-phase Hologram

(c) Reconstruction



(d) NMSE v.s. iteration plot

Fig. 2.14 DBS algorithm running on the low resolution target

As the DBS algorithm only flips one pixel per iteration, it naturally takes significantly longer to generate holograms with higher resolution. To test the programme on a smaller image for much quicker convergence, another target image has been designed as shown in Fig. 2.14a, which is also rotationally symmetrical as it is used for the binary-phase hologram generation. After 10^6 iterations, which took 10 minutes, the hologram generated is shown in Fig. 2.14b and its resulting reconstruction is shown in Fig. 2.14c, which has an NMSE of 9.6881×10^{-6} and an SSIM of 0.2871. The NMSE v.s. iteration plot in Fig. 2.14d shows that it reaches a good convergence at around 2×10^5 iterations, corresponding to around 2 minutes. And the

curve of NMSE is monotonically decreasing with iteration number, as only holograms with better results are accepted during the iterations.

In summary, the DBS algorithm is a slow but working algorithm for binary phase hologram generation. The programme running time scales up significantly when the target image's resolution gets higher. And also, as it only cares about local optimality at each iteration, it is a greedy algorithm that only follow the steepest descent route, which could easily get trapped in a local minimum where flipping any bit is not getting better reconstruction. Another consequence of the random nature is that the generated hologram will be different at each run, so the quality of the resulting reconstruction (R) will depend on how 'lucky' each run is. The Simulated Annealing (SA) algorithm [47] in the next session aims to resolve this issue.

2.3.3 Simulated Annealing (SA) Algorithm

Simulated Annealing (SA) algorithm [47] is a variant of the DBS algorithm. It adopts a probabilistic approach to avoid the steepest gradient descent. Its name derives from the fact that it approximates the recrystallisation process during metal annealing and is particularly well-suited to avoiding the trap of local minima [23]. To implement this idea, we then need a function ($\mathscr{L}$) to calculate the probability of the hologram (H), and a threshold p_t to decide whether the probability is high enough for the according hologram to be accepted. In this thesis, the probability is selected to be a random function and the threshold is chosen to be 0.9. The pseudocode for this algorithm is listed in Algorithm 4.

Algorithm 4 Simulated Annealing (SA) algorithm

Input: Target image T, Propagation function $\mathscr{P}$, Loss function $\mathscr{L}$, Number of iterations N, Probability function $\mathscr{Z}$, Probability threshold p_t

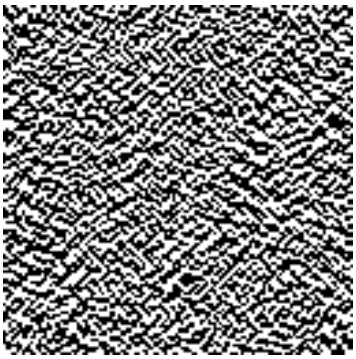
Output: Phase hologram H and its reconstruction intensity R

// Start with a random hologram with a size matching T
$H \leftarrow \text{Rand}(\text{Size}(T))$
$R \leftarrow |\mathscr{P}[e^{jH}]|^2$
$L \leftarrow \mathscr{L}[R, T]$
for $n = 1$ to N **do**
    // Flip a random pixel in the hologram
    $H_n \leftarrow \text{FlipRandomPixel}(H)$

    // Calculate the loss function for the new hologram
    $R_n \leftarrow |\mathscr{P}[e^{jH_n}]|^2$
    $L_n \leftarrow \mathscr{L}[R_n, T]$

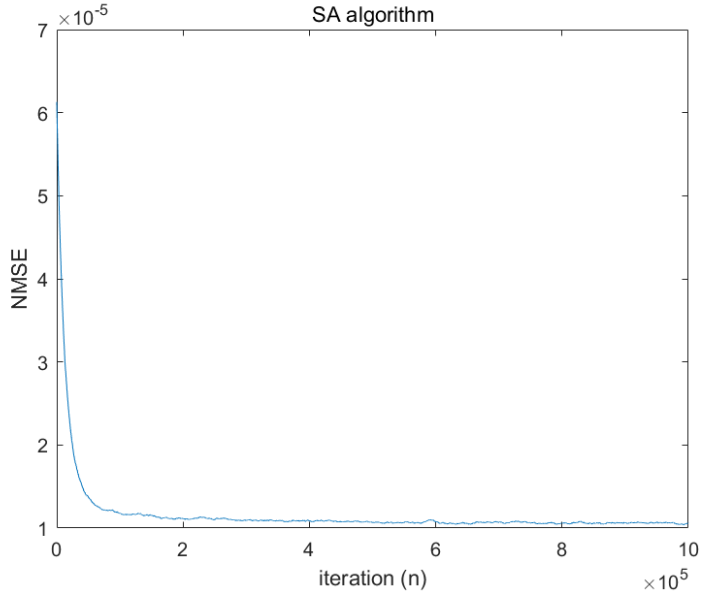
    // Compare the new loss with the old one
    if $L_n < L$ **then**
        // Accept the new hologram if loss is lower
        $H \leftarrow H_n$
        $R \leftarrow R_n$
        $L \leftarrow L_n$
    else
        // Calculate the probability of the hologram
        $p_n \leftarrow \mathscr{Z}[H_n]$
        if $p_n > p_t$ **then**
            // Accept the new hologram if the probability exceeds the threshold
            $H \leftarrow H_n$
            $R \leftarrow R_n$
            $L \leftarrow L_n$
        end if
    end if
end for

(a) Target image ($128px \times 128px$)



(b) Binary-phase Hologram

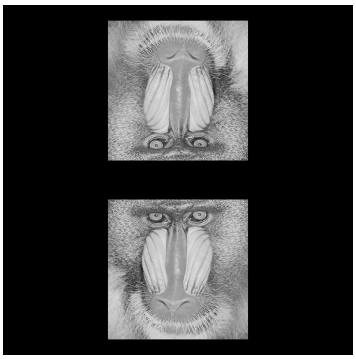
(c) Reconstruction



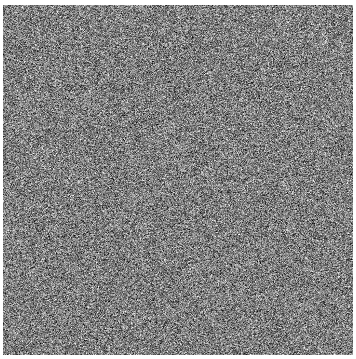
(d) NMSE v.s. iteration plot

Fig. 2.15 SA algorithm running on the low resolution target

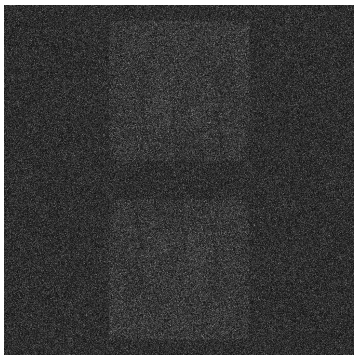
An implementation of SA algorithm with $p_t = 0.9$ was carried out on the low resolution target in Fig. 2.15a, and the resulting binary-phase hologram and its reconstruction are shown in shown in Fig. 2.15b and Fig. 2.15c respectively. From the NMSE v.s. iteration plot in Fig. 2.15d, it can be seen that, instead of the monotonic decrease observed in Fig. 2.14d for DBS algorithm, the SA algorithm has occasional rises in NMSE, which happens when the probability p_n exceeds the threshold p_t. The final NMSE was recorded to be 1.0542×10^{-5} and the final SSIM was 0.2750, which are both slightly worse than the DBS algorithm in this case. Due to the probabilistic nature of the SA algorithm, although it can avoid being trapped in local optimal points, the 'jump backs' can also cause delays in convergence.



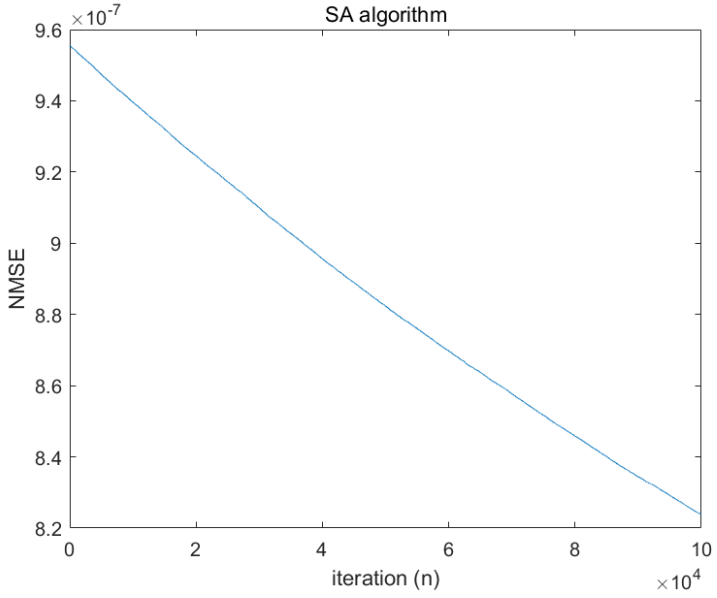
(a) Target image ($1024px \times 1024px$)



(b) Binary-phase Hologram



(c) Reconstruction



(d) NMSE v.s. iteration plot

Fig. 2.16 SA algorithm running on the rotationally symmetrical mandrill target

Then the SA algorithm was run for the mandrill target in Fig. 2.16a. The convergence plot in Fig. 2.16d shows that it did not converge within 10^5 iterations, which took around one hour. The binary phase hologram generated is shown in Fig. 2.16b and its corresponding reconstruction intensity is shown in Fig. 2.16c, which is of very poor quality. Both the DBS and the SA algorithms rely on flipping only a single pixel per iteration, which is very inefficient. A better algorithm should change the value of multiple pixels at every iteration for better efficiency.

2.3.4 Gerchberg-Saxton (GS) Algorithm

The Gerchberg-Saxton (GS) algorithm [19] is a revolutionary algorithm and is much better and more robust than the algorithms introduced in the previous sections. Although being more than 50 years old, the GS algorithm is still frequently used and has lots of variants [48–50]. It functions by iteratively determining the phase profile of the hologram required to reconstruct a target image, looping between the hologram and the reconstruction plane, and applying constraints to each plane accordingly during each iteration. GS algorithm is very easy to implement, its pseudocode is shown in Algorithm 5.

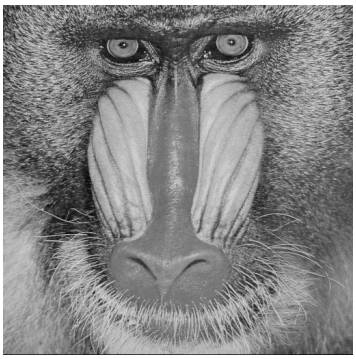
Algorithm 5 Gerchberg-Saxton (GS) Algorithm

Input: Target image T, Propagation function $\mathscr{P}$, Number of iterations N, Initial phase Φ (e.g. random, zeros, or other patterns)

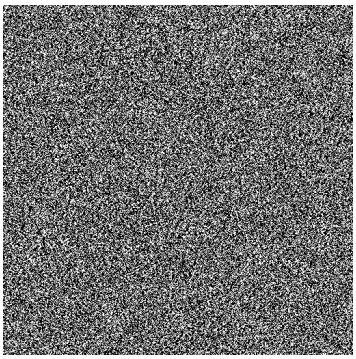
Output: Phase hologram H and its reconstruction intensity R

```
// Initiate E with amplitude √T and initial phase Φ
E ← √T * e^{jΦ}
for n = 1 to N do
    // Compute the hologram plane
    A ← 𝒫^{-1}[E]
    // Apply the phase-only constraint at the hologram plane
    A ← e^{j∠A}

    // Compute the propagation for the new hologram
    E ← 𝒫[A]
    // Apply the target field amplitude constraint at the reconstruction plane
    E ← √T * e^{j∠E}
end for
H ← ∠A
R ← |𝒫[A]|^2
```

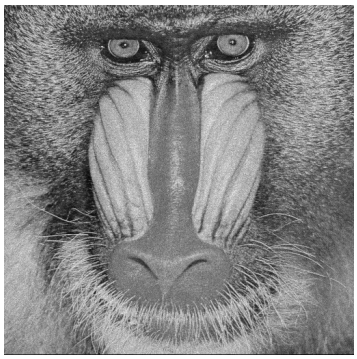
The GS algorithm described in Algorithm 5 was implemented in MATLAB and was first run on the mandrill target image in Fig. 2.9 the output results are shown in **??**. It can be seen from Fig. 2.17c that, the reconstruction after 30 iterations of GS algorithm reached a very good result, having an NMSE of 2.6612×10^{-8} and an SSIM of 0.7940, which are both much better than the single-iteration Naive method's result in Fig. 2.11c.



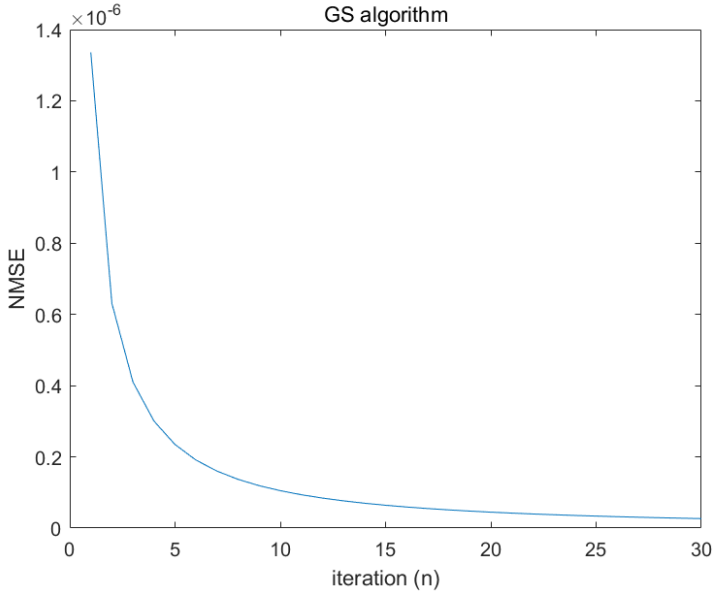
(a) Target image ($512px \times 512px$)



(b) Hologram



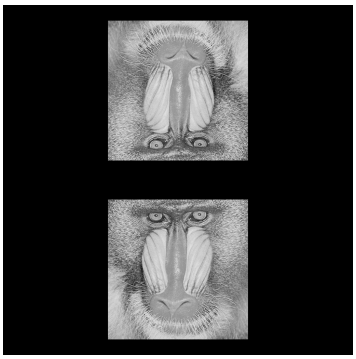
(c) Reconstruction



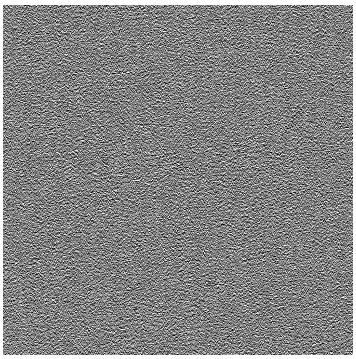
(d) NMSE v.s. iteration plot

Fig. 2.17 GS algorithm output on the mandrill target

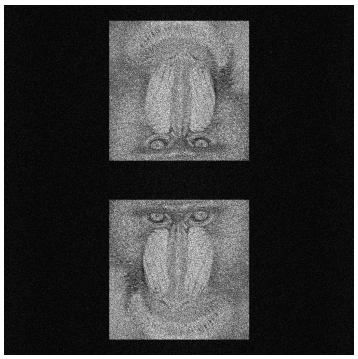
The NMSE v.s. iteration plot in Fig. 2.17d shows that the GS algorithm converged quickly, providing very good result in tens of iterations, which is much fewer than the DBS and SA algorithms. Although the GS algorithm is more computationally expensive at each iteration, as it needs to compute both a forward and an backward propagation, leading to two Fourier transforms every iteration, the GS algorithm is still much faster and provides much better reconstruction quality than the DBS and SA algorithms.



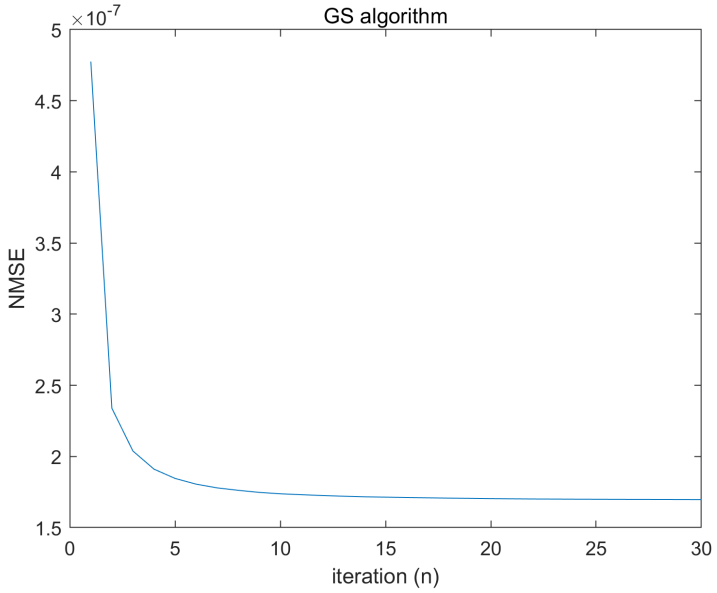
(a) Target image ($1024px \times 1024px$)



(b) Binary-phase Hologram



(c) Reconstruction



(d) NMSE v.s. iteration plot

Fig. 2.18 GS algorithm running on the rotationally symmetrical mandrill target

Then the GS algorithm was adapted to generate binary-phase holograms, for use on the binary-phase SLM in this thesis. The change was simply implemented by adding a quantisation function ($\mathscr{Q}$) when applying the phase-only constraint at the hologram plane (i.e. the line '$A \leftarrow e^{j\angle A}$' in Algorithm 5 is changed to '$A \leftarrow e^{j\mathscr{Q}[\angle A]}$'). The results are shown in Fig. 2.18. Fig. 2.18d shows good convergence within 20 iterations. The resulting binary-phase hologram is shown in Fig. 2.18b and its corresponding reconstruction in Fig. 2.18c has an NMSE of 1.6968e-07 and an SSIM of 0.0619, which are both better than the Naive method's result in Fig. 2.12. In summary, the GS algorithm is quick and robust. On my laptop computer of model ASUS ROG Zephyrus M16, which has a CPU of model i7-11800H and a GPU of

model RTX3060, the 30 iterations took 1.5 seconds to complete. It reached convergence in tens of iterations. However, as it is still iterative, generating holograms in real-time is still a challenge, and the reconstruction still suffers from noise.

2.3.5 One-Step Phase Retrieval (OSPR) Algorithm

OSPR algorithm was first demonstrated by Buckley [51]. It is a solution to high-quality hologram reconstruction that relies on time multiplexing of holograms, exploiting the response time of eye in order to reduce noise in the replay field [15]. The random noises are averaged by the eye, while the target image stays, so that the average noise can be reduced. The perceived noise is lessened by the temporal average detected by the eye, rather than computational optimisation of the hologram [15]. The pseudocode for OSPR is shown in Algorithm 6 below.

Algorithm 6 One-Step Phase Retrieval (OSPR) algorithm

Input: Target image T, Propagation function $\mathscr{P}$, Number of sub-frames S, Quantisation function $\mathscr{Q}$

Output: List of phase holograms $H[1 \dots S]$

```
// Compute a list of hologram sub-frames based on different additive random phase
for s = 1 to S do
    E ← √T ∗ RandomPhase()
    A ← P^{-1}[E]
    H[s] ← Q[∠A]
end for

// Then display the sub-frames on the phase modulator sequentially
s ← 1
while True do
    Display(H[s])
    s ← s + 1
    if s > S then
        s ← 1
    end if
end while
```

When generating the list of holograms, it repetitively computes the inverse propagation of the target amplitude (which is the square-root of the target intensity T) multiplied by different random phases, for a total of S times to generate S hologram sub-frames ($H[1 \dots S]$). The computation of each hologram sub-frame is the same as the Naive method in Algorithm 2 discussed in Section 2.3.1. Then the S hologram sub-frames are displayed sequentially on a SLM having a refresh rate being so fast that the average reconstruction intensity is perceived by the human eyes. As currently available fast SLMs are binary-phase modulators, an example run on the rotationally symmetrical target previously used in Fig. 2.13a was carried out for 24 sub-frames ($S = 24$). The results are summarised in Fig. 2.19.

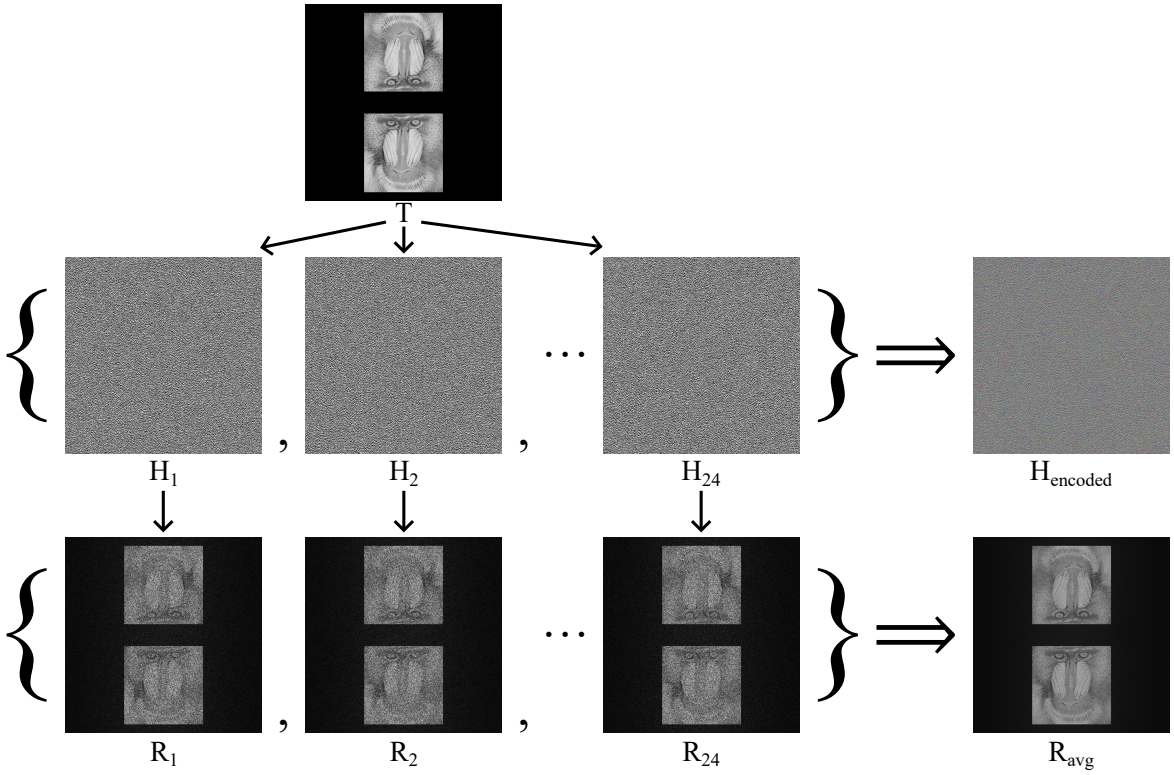


Fig. 2.19 OSPR algorithm running on the rotationally symmetrical mandrill target

In Fig. 2.19, a total of 24 binary-phase hologram sub-frames ($H_1, H_2, \dots, H_{24}$) were generated for the target image T. For easier data transfer and to fit with the common display signal formats, the 24 binary-phase hologram sub-frames are encoded into a single file with 8 bit depth and RGB (red-green-blue) channels, so that each of the $(8 \times 3 =)24$ bit-planes corresponds to a single binary-phase hologram sub-frame. For a quantitative analysis, the reconstruction intensities of the hologram sub-frames are computed in $R_1, R_2, \dots, R_{24}$ respectively, whose average is R_{avg} in Fig. 2.19. The average reconstruction intensity

R_{avg} has an NMSE of 9.8632×10^{-8} and an SSIM of 0.1321, which are both significantly better than the GS algorithm (NMSE=1.6968×10^{-7}, SSIM=0.0619) and the Naive method (NMSE=4.5452×10^{-7}, SSIM=0.0603).

The major advantage of the OSPR algorithm is that it is superfast. It is non-iterative and requires only one Fourier Transform per frame, taking less than a second to generate a 24-frame hologram set. Its non-iterative nature also allows it to be parallelised to further improve computation speed, which is crucial in how the Light Blue Optics made real-time holographic laser projector possible in 2010 [52], although the product was later discontinued for commercial reasons.

2.3.6 Adaptive One-Step Phase Retrieval (AD-OSPR) Algorithm

2.3.7 3D CGH

Section 2.3.1 - Section 2.3.5 described several algorithms to generate a phase hologram for a single slice target field. Then the problem arises as how to generate a hologram for 3D target, and hence making full use of the major benefit of holography, which is true 3D reconstruction. There are several ways to achieve this.

Multi-Layer Slicing

The simplest method is to slice the 3D target into a set of layers, and then generate a set of phase holograms for each slice at its according distance (z) using Fresnel propagation model. Then the set of phase holograms are added up to form the final phase hologram, based on the principle of superposition.

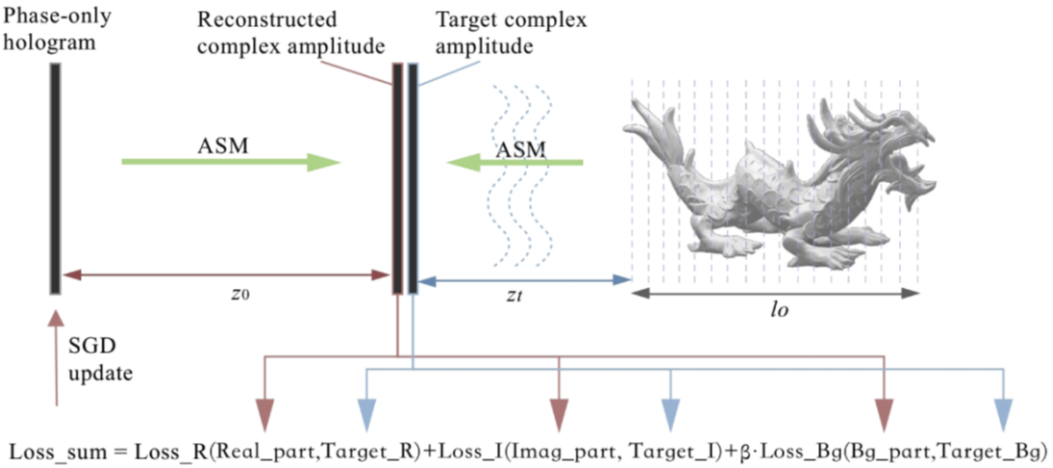


Fig. 2.20 Schematic diagram of the intermediate plane method [17]

There is also an alternative solution, that is to propagate each slice of the 3D target into an intermediate plane, and then run the phase retrieval algorithms on the complex target field (T) with a loss function ($\mathscr{L}$), where needed, that accounts for both amplitude and phase components. The schematic of this method is shown in Fig. 2.20 [17].

Point Cloud Method

Point cloud method, as its name infers, divides a 3D target into a collection of points, each emitting a spherical wave, and then summed under the principle of superposition. The point cloud method is extremely computationally heavy and is very slow.

2.4 Numerical Optimisation Methods

In addition to the conventional CGH algorithms described in Section 2.3, literature review has also found some recent work that compute CGH using numerical optimisation methods [53–55, 17, 3]. This section is a review on what numerical optimisation is and how it works. And the implementation of optimisation of CGH is further discussed in Chapter 4.

Numerical optimisation methods aim to find an optimal solution which minimise an objective function numerically. They begin with an initial guess of the optimal solution ($\mathbf{x}_0$) and then, after iterations, generate a sequence of gradually improved estimates until they reach a solution [56]. If we have $\mathbf{x}$ as the vector of variables, and denote $f(\mathbf{x})$ as the objective function, which is a function of x we want to minimise, any unconstrained optimisation problem can be written as

$$\underset{\mathbf{x}\in R^n}{\text{minimise}} \quad f(\mathbf{x}) \tag{2.34}$$

Numerical optimisation then calculate the optimal solution $\mathbf{x}^*$ iteratively, the iteration is given by

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k \tag{2.35}$$

where the positive scalar α_k is called step length, or sometimes may be referred as 'learning rate' in some context especially when related to machine learning, and the vector $\mathbf{p}_k$ is the search direction, which usually takes the form of

$$\mathbf{p}_k = -\mathbf{B}_k^{-1} \nabla f_k \tag{2.36}$$

where $\mathbf{B}_k$ is a nonsingular matrix that varies for different optimisation methods. The gradient ∇f_k, if unable to evaluate directly, can be approximated by

$$\begin{gathered} \nabla f_k \approx \frac{f_{k+1} - f_k}{\mathbf{x}_{k+1} - \mathbf{x}_k} \\ \text{where } f_k \text{ denotes } f(\mathbf{x}_k) \end{gathered} \tag{2.37}$$

The strategy used to determine $\mathbf{p}_k$ distinguishes one algorithm from another. Most methods make use of the values of f, ∇f and $\nabla^2 f$, and some methods even make use of the accumulated historical values of those derivatives, which are further discussed in Section 2.4.1 - Section 2.4.4.

2.4.1 Gradient Descent

Gradient descent (GD) is a first-order optimisation method, it finds a local minimum by following the negative of the gradient (i.e. the steepest descent direction). The $\mathbf{B}_k$ (in Eq. (2.36)) for gradient descent simply takes the value of $\mathbf{I}$, which is the identity matrix. And the search direction becomes:

$$\mathbf{p}_k = -\nabla f_k \tag{2.38}$$

The steepest descent method is very intuitive: among all possible directions to move away from $\mathbf{x}_k$, the steepest gradient direction is the one which f decreases most rapidly. The advantage of this method is that it requires few computation and memory resource, because it only requires a computation of the first derivative, and it does not require any accumulation of historical gradients. However, it is a greedy method that only considers the current iteration without any global consideration, so it can be extremely slow on complicated problems. [56]

To work around the disadvantage, a few variants have emerged, such as AdaGrad [57], RMSProp [58] and Adam [59] which combines the advantages of AdaGrad and RMSProp. It can be said to be an iconic variant of the gradient descent, often referred to as gradient descent with momentum. The name Adam is derived from adaptive moment estimation. Adam algorithm is based on adaptive estimates of lower-order moments [59]. Adam method computes individual adaptive learning rates for different parameters from estimates of first and second moments of the gradients [59]. Although some improvements are observed, it still does not fix entirely.

2.4.2 Newton's Method

Newton's method is a second-order optimisation method. Its search direction is derived from the second-order Taylor series approximation to $f(\mathbf{x}_k + \mathbf{p})$, which is

$$f(\mathbf{x}_k + \mathbf{p}) \approx f_k + \mathbf{p}^T \nabla f_k + \frac{1}{2}\mathbf{p}^T \nabla^2 f_k \mathbf{p} \stackrel{\text{def}}{=} m_k(\mathbf{p}) \tag{2.39}$$

The Newton direction can then be obtained by finding the vector $\mathbf{p}$ that minimises $m_k(\mathbf{p})$. By setting the derivative of $m_k(\mathbf{p})$ to zero, $\mathbf{p}$ can be obtained as:

$$\mathbf{p}_k = -\nabla^2 f_k^{-1} \nabla f_k \tag{2.40}$$

By comparing Eq. (4.5) to Eq. (2.36), it can be seen that the Newton's method has a $\mathbf{B}_k$ of $\nabla^2 f_k$. Unlike the gradient descent method, there is a "natural" step length of 1 associated

with the Newton direction, so $\alpha_k = 1$ by default and is only adjusted when it does not produce a satisfactory reduction in the value of f.

The Newton direction is reliable when the difference between the true function $f(\mathbf{x}_k + \mathbf{p})$ and its quadratic model $m_k(\mathbf{p})$ is not too large. Methods that use the Newton direction have a fast rate of local convergence, typically quadratic. After a neighbourhood of the solution is reached, convergence to high accuracy often occurs in just a few iterations. The main drawback of the Newton direction is the need for the Hessian $\nabla^2 f_k$. Explicit computation of this matrix of second derivatives can sometimes be a cumbersome, error-prone, and expensive process. [56]

2.4.3 Quasi-Newton Method: Broyden-Fletcher-Goldfarb-Shanno (BFGS)

Quasi-Newton method provides an attractive alternative to Newton's method, in that they do not require computation of the Hessian and yet still attain a super linear rate of convergence. In place of the true Hessian $\nabla^2 f_k$, they use an approximation $\mathbf{H}_k \overset{\text{def}}{=} \mathbf{B}_k^{-1}$, which is updated after each step to take account of the additional knowledge gained during the step. The updates make use of the fact that changes in the gradient provide information about the second derivative of f along the search direction. The most popular quasi-Newton algorithm is the BFGS method, named for its discoverers Broyden, Fletcher, Goldfarb, and Shanno. [56]

The process of the BFGS method is shown below:

$$denote \begin{cases} \mathbf{H}_k &= \mathbf{B}_k^{-1} \\ \mathbf{p}_k &= -\mathbf{H}_k \nabla f_k \end{cases} \tag{2.41}$$

$$Initiate\ \mathbf{H}_0 \leftarrow \frac{\mathbf{y}_k^T \mathbf{s}_k}{\mathbf{y}_k^T \mathbf{y}_k} \mathbf{I} \tag{2.42}$$

$$update\ \mathbf{H}_{k+1} = (\mathbf{I} - \rho_k \mathbf{s}_k \mathbf{y}_k^T)\mathbf{H}_k(\mathbf{I} - \rho_k \mathbf{y}_k \mathbf{s}_k^T) + \rho_k \mathbf{s}_k \mathbf{s}_k^T \tag{2.43}$$

$$where \begin{cases} \mathbf{s}_k &= \mathbf{x}_{k+1} - \mathbf{x}_k \\ \mathbf{y}_k &= \nabla f_{k+1} - \nabla f_k \\ \rho_k &= \frac{1}{\mathbf{y}_k^T \mathbf{s}_k} \end{cases} \tag{2.44}$$

The algorithm is robust, and its rate of convergence is super linear, which is fast enough for most practical purposes. Even though Newton's method converges more rapidly (that is, quadratically), its cost per iteration usually is higher, because of its need for second

derivatives and solution of a linear system. The drawback is that, it is not directly applicable to large optimisation problems because $\mathbf{H}_k$'s are usually dense, requiring large storage and computational requirements. [56]

2.4.4 Large Scale Quasi-Newton Method: Limited Memory BFGS (L-BFGS)

L-BFGS algorithm [60] modifies the technique described in Section 2.4.3 to obtain Hessian approximations that can be stored compactly in just a few vectors of length n, where n is the number of unknowns in the problem. The main idea of this method is to use curvature information from only the most recent iterations to construct the Hessian approximation. Curvature information from earlier iterations, which is less likely to be relevant to the actual behaviour of the Hessian at the current iteration, is discarded in the interest of saving storage. [56]

Denoting $\mathbf{V}_k = \mathbf{I} - \rho_k \mathbf{y}_k \mathbf{s}_k^T$, Eq. (2.43) can be written as:

$$\mathbf{H}_{k+1} = \mathbf{V}_k^T \mathbf{H}_k \mathbf{V}_k + \rho_k \mathbf{s}_k \mathbf{s}_k^T \tag{2.45}$$

The inverse Hessian approximation $\mathbf{H}_k$ will generally be dense, so that the cost of storing and manipulating it is prohibitive when the number of variables is large. To circumvent this problem, we store a modified version of $\mathbf{H}_k$ implicitly, by storing a certain number (say, m) of the vector pairs $\{\mathbf{s}_i, \mathbf{y}_i\}$ used in the Eq. (2.43) and Eq. (4.9). The product $\mathbf{H}_k \nabla f_k$ can be obtained by performing a sequence of inner products and vector summations involving ∇f_k and the pairs $\{\mathbf{s}_i, \mathbf{y}_i\}$. After the new iterate is computed, the oldest vector pair in the set of pairs $\{\mathbf{s}_i, \mathbf{y}_i\}$ is replaced by the new pair $\{\mathbf{s}_k, \mathbf{y}_k\}$ obtained from the current step (Eq. (4.9)). In this way, the set of vector pairs includes curvature information from the m most recent iterations. Practical experience has shown that modest values of m (between 3 and 20, say) often produce satisfactory results. We now describe the updating process in a little more detail. At iteration k, the current iterate is $\mathbf{x}_k$ and the set of vector pairs is given by $\{\mathbf{s}_i, \mathbf{y}_i\}$ for $i = k-m, \ldots, k-1$. We first choose some initial Hessian approximation $\mathbf{H}_k^0$ (in contrast to the standard BFGS iteration, this initial approximation is allowed to vary from iteration to iteration) and find by repeated application of Eq. (2.43) that the L-BFGS approximation $\mathbf{H}_k$ satisfies the following formula: [56]

$$\begin{aligned}\mathbf{H}_k = &(\mathbf{V}_{k-1}^T \cdots \mathbf{V}_{k-m}^T)\mathbf{H}_k^0(\mathbf{V}_{k-m} \cdots \mathbf{V}_{k-1}) \\ &+ \rho_{k-m}(\mathbf{V}_{k-1}^T \cdots \mathbf{V}_{k-m+1}^T)\mathbf{s}_{k-m}\mathbf{s}_{k-m}^T(\mathbf{V}_{k-m+1} \cdots \mathbf{V}_{k-1}) \\ &+ \rho_{k-m+1}(\mathbf{V}_{k-1}^T \cdots \mathbf{V}_{k-m+2}^T)\mathbf{s}_{k-m+1}\mathbf{s}_{k-m+1}^T(\mathbf{V}_{k-m+2} \cdots \mathbf{V}_{k-1}) \\ &+ \cdots \\ &+ \rho_{k-1}\mathbf{s}_{k-1}\mathbf{s}_{k-1}^T \end{aligned} \tag{2.46}$$

From this expression we can derive a recursive procedure (Algorithm 8) to compute the product $\mathbf{H}_k \nabla f_k$ efficiently.

Algorithm 7 L-BFGS two-loop recursion [56]

$\mathbf{q} \leftarrow \nabla f_k$
for $i = k-1, k-2, ..., k-m$ **do**
  $\alpha_i \leftarrow \rho_i \mathbf{s}_i^T \mathbf{q}$
  $\mathbf{q} \leftarrow \mathbf{q} - \alpha_i \mathbf{y}_i$
end for
$\mathbf{r} \leftarrow \mathbf{H}_k^0 \mathbf{q}$
for $i = k-m, k-m+1, ..., k-1$ **do**
  $\beta \leftarrow \rho_i \mathbf{y}_i^T \mathbf{r}$
  $\mathbf{r} \leftarrow \mathbf{r} + \mathbf{s}_i(\alpha_i - \beta)$
end for
Step with $\mathbf{p}_k \leftarrow -\mathbf{H}_k \nabla f_k = -\mathbf{r}$

Apart from being inexpensive, L-BFGS has the advantage that the multiplication by the initial matrix $\mathbf{H}_k^0$ is isolated from the rest of the computations, allowing this matrix to be chosen freely and to vary between iterations. A method for choosing $\mathbf{H}_k^0$ that has proved effective in practice is to use the same as BFGS as stated in Eq. (2.42). [56]

Chapter 3

Gamma Correction in Holographic Projection

Note: This Chapter is a continuation of my masters project in 2019-2020, which was unexpectedly terminated early by COVID-19. During my first year of PhD study, all measurements have been retaken for quantified analysis.

In order to improve image quality of holographic projection, the idea of the gamma correction method arose to improve the contrast of the replay field. Every display has an inherent property known as the gamma value γ, which essentially describes the transfer function between input pixel value and output pixel energy[61]. Gamma correction is normally done via a look-up table or correction curve which allows the relationship between the input and output to be adjusted. In a computer-generated holographic projection system, the image is generated via diffraction of light from spatial light modulators. In this process, several factors contribute to non-linearities between the replay field and the target image. This section evaluates the gamma response of the overall system experimentally, and then applies a gamma correction method, with the aim of increasing the image quality of a holographic projection system. Both a notable increase in replay field quality alongside a significant reduction in mean squared error were observed, demonstrating the effectiveness of gamma correction in holographic projection.

3.1 Experimental Setup

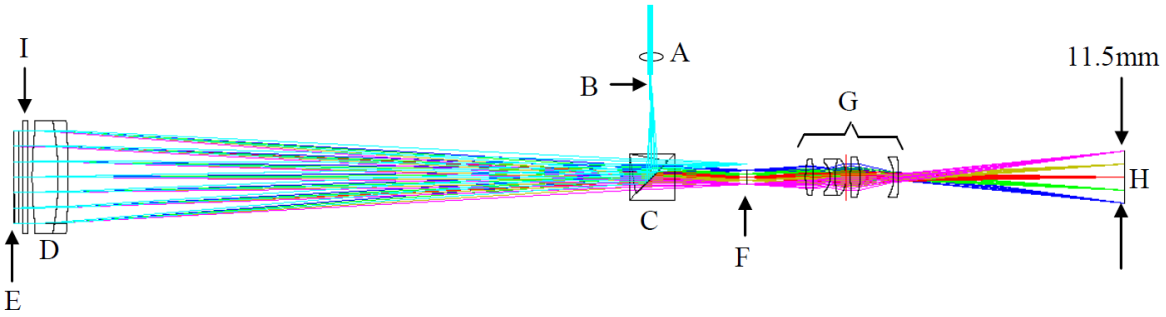
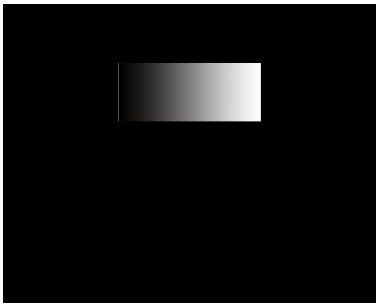


Fig. 3.1 Optical setup [18]

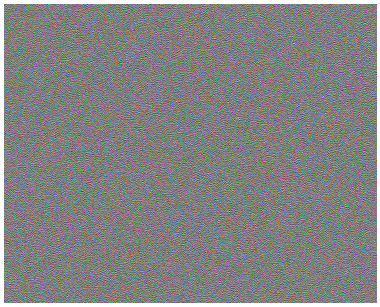
The holographic projector used in this experiment was a Fourier projection system developed by Freeman [18], as shown in Fig. 3.1. The design is consisted of a diode-pumped solid-state (DPSS) 532 nm 50mW laser source, focussed down by an aspheric singlet (A), the focus of which becomes the diffraction limited point source (B) for the projector. The beam then passes through a polarising beam splitter cube (C) to a collimating lens (D), which illuminates the SLM (E). The SLM is a binary phase SXGA-R2 ForthDD ferroelectric Liquid crystal on silicon (LCOS) micro-display with a refresh rate of 1440Hz, a pixel pitch of 13.6 µm and a resolution of 1280×1024. An aperture at point (F) spatially filters out the other orders, leaving only one first order, which is then magnified up by a finite conjugate lens group (G) to produce an image, of the required size, on the screen (H). [18]

The holograms displayed on the SLM are generated using the one-step phase retrieval (OSPR) algorithm [15] explained in Section 2.3.5, where each group of 24 individual, binary-phase holograms are encoded as the 8-bit reg, green, blue (RGB) channels of a 24-bit image to interface with the SLM driver electronics. The SLM displays each bit plane sequentially, with ones and zeros mapping to opposing phase modulations at each pixel. The images were captured using a Canon 550D camera with an EFS 18-55 mm lens. To ensure fair comparison, the camera was set to the same manual setting when comparing each pair of replay fields before and after gamma correction. The images captured were in 24-bit RGB colour, which were subsequently converted to grey-scale in 8-bit depth when calculating normalised mean squared error (NMSE).

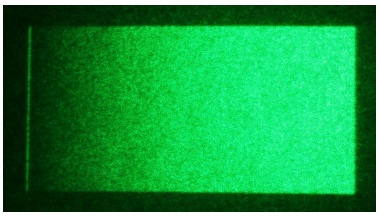
3.2 Determining the Gamma Correction Curve



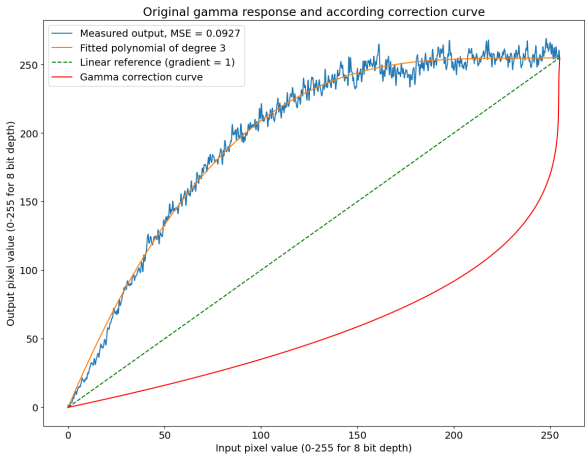
(a) Input linear grey-scale ramp



(b) CGH of Fig. 3.2a



(c) Holographic projection replay field of Fig. 3.2b



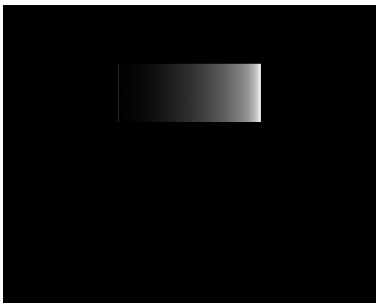
(d) Gamma response curve and according correction curve

Fig. 3.2 Measurement of gamma response, which inverse is the correction

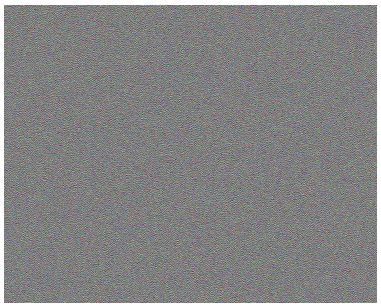
To determine the gamma correction curve of the holographic projection system, the gamma response needs to be measured first. A hologram was generated to form a linear grey-scale ramp of brightness from 0 to 255, as shown in Fig. 3.2a, along with a single pixel white (255) strip at the left end as a fiducial marker to demonstrate the beginning of the grey-scale region [15].

The projection output of the linear grey-scale ramp was captured as shown in Fig. 3.2c. From this the gamma response curve was determined, by averaging each column of pixels and normalising to a percentage scale, forming the blue line in Fig. 3.2d. A three-degree polynomial fit was applied, generating a smoothed gamma response curve (yellow line in Fig. 3.2d).

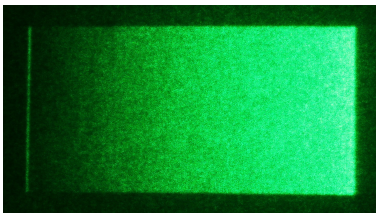
The resultant gamma response curve exhibits a high degree of non-linearity. By taking the mean of the square of the error between the measured output (blue line) and the linear reference (green dashed line), the NMSE of the measured output was calculated to be 0.0927. To correct the gamma response, the gamma correction curve (red line) was formed by inverting the gamma response curve.



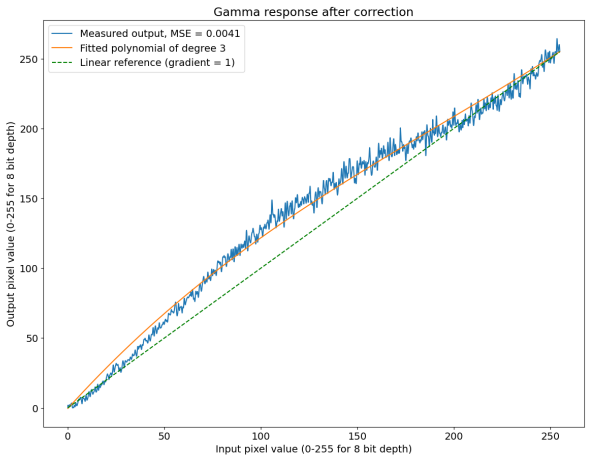
(a) Input gamma corrected ramp



(b) CGH of Fig. 3.3a



(c) Holographic projection replay field of Fig. 3.3b



(d) Gamma response curve after correction

Fig. 3.3 Application of the correction curve on the grey-scale ramp

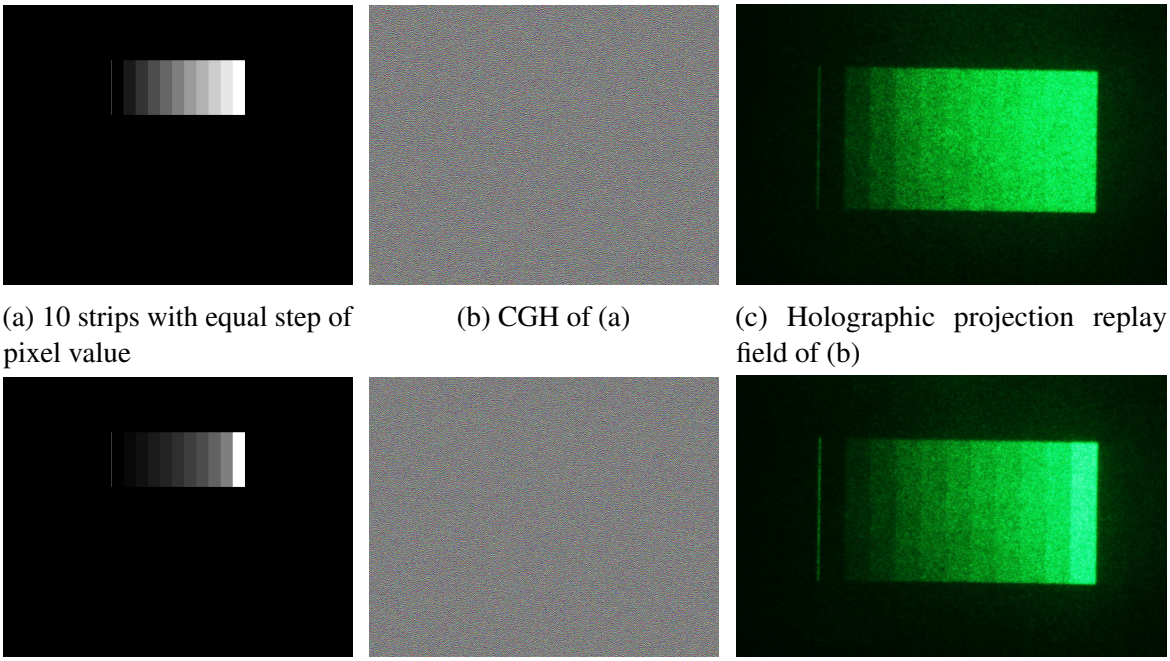
Subsequently, the gamma correction curve was implemented to adjust the grey-scale ramp, achieving the gamma corrected grey-scale ramp as shown in Fig. 3.3a. The gamma corrected projection output was captured as shown in Fig. 3.3c. By using the same method of averaging columns of pixels, the gamma corrected output was measured and plotted in Fig. 3.3d. It can be seen that the corrected gamma response was much closer to linear comparing to the original gamma response, and the NMSE was calculated to be 0.0041.

Table 3.1 Gamma response results before and after gamma correction

	NMSE	Percentage
Gamma response before correction	0.0927	100%
Gamma response after correction	0.0041	4.42%

Hence, as demonstrated in Table 3.1, gamma correction achieved a 95.58% reduction in MSE, which was a significant improvement, proving the effectiveness of gamma correction method on the grey-scale ramp.

3.3 Applying the Gamma Correction Curve



(a) 10 strips with equal step of pixel value

(b) CGH of (a)

(c) Holographic projection replay field of (b)

(d) Digital pre-distortion of (a)

(e) CGH of (d)

(f) Holographic projection replay field of (f)

Fig. 3.4 Application of the correction curve on 10-step strips

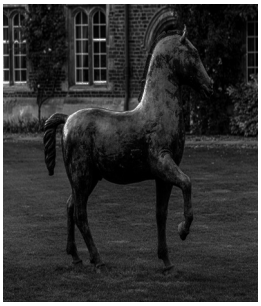
As shown in Fig. 3.4, when CGH is computed for the 10 strips with equal step of pixel value Fig. 3.4a, the right few strips in Fig. 3.4c are barely distinguishable. After applying the correction curve obtained in Section 3.2, it can be seen that each pair of adjacent strips in

Fig. 3.4f are much more distinguishable, validating the effectiveness of the gamma correction method.

(a) Sample image 1: City Scene [62]

(b) Sample image 2: Horse

(c) Sample image 1 after gamma correction



(d) Sample image 2 after gamma correction

Fig. 3.5 Application of the correction curve on two sample real-word images

Then the gamma correction curve was applied to the two sample images as shown in Fig. 3.5.

(a) Replay field of Sample image 1 before gamma correction (NMSE=0.06139)

(b) Replay field of Sample image 2 before gamma correction (NMSE=0.04309)

(c) Replay field of Sample image 1 after gamma correction (NMSE=0.04920)

(d) Replay field of Sample image 2 after gamma correction (NMSE=0.03635)

Fig. 3.6 Projection output of the two sample images before and after gamma correction

The replay fields of the holographic projection of uncorrected images are shown in Fig. 3.6a and Fig. 3.6b, and the replay fields of the holographic projection of images after gamma correction are shown in Fig. 3.6c and Fig. 3.6d respectively.

As shown in Fig. 3.6a, it can be seen that, before gamma correction, the edges between the buildings and the sky were quite ambiguous, with most detail of the sky being lost. In comparison, after gamma correction, the replay field in Fig. 3.6c provided not only sharper edges between buildings and the sky, but also more detail of clouds in the sky. The NMSE of the replay field for sample image 1 decreased from 0.06139 to 0.04920, which was a 19.86% reduction.

In Fig. 3.6b, before gamma correction, the horse was difficult to distinguish from the background, especially around the horse's back area. But after gamma correction, as shown

in Fig. 3.6d, contrast has been significantly boosted and the fine detail around this part of the horse is more evident. The NMSE of the replay field for sample image 2 decreased from 0.04309 to 0.03635, which was a 15.64% reduction.

Table 3.2 Gamma correction results for sample images

Sample image 1	NMSE	Percentage
Before gamma correction	0.06139	100%
After gamma correction	0.04920	80.15%
Sample image 2	**NMSE**	**Percentage**
Before gamma correction	0.04309	100%
After gamma correction	0.03635	84.36%

Hence, as summarised in Table 3.2, gamma correction achieved a 19.86% reduction in NMSE for sample image 1 and a 15.64% reduction in NMSE for sample image 2, proving the effectiveness of gamma correction method on real-world test images.

3.4 Summary

The gamma response of holographic projection can exhibit a high degree of non-linearity. By projecting a linear grey-scale ramp, the gamma response of the holographic projection system was measured. The gamma correction curve, which was simply the inverse of gamma response, was applied to the grey-scale ramp and successfully reduced the NMSE by 95.58%. And then the gamma correction method was applied on two sample images, it was observed that more details were shown in the replay field after gamma correction, and the NMSE's of the two example images were reduced by 19.86% and 15.64%. Hence, we have demonstrated the effectiveness of gamma correction method to boost image quality for a holographic projection system.

Chapter 4

Multi-Depth Phase-Only Hologram Optimization using L-BFGS Algorithm with Sequential Slicing

4.1 abstract

We implement limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) optimization of phase-only computer-generated hologram (CGH) for multi-depth three-dimensional (3D) target. Instead of computing the full 3D reconstruction of the hologram, we use a novel method using L-BFGS with sequential slicing (SS) for partial evaluation of the hologram during optimization, that only computes loss for a single slice of the reconstruction at every iteration. We demonstrate that its ability to record curvature information enables L-BFGS to have good quality imbalance suppression under SS technique.

4.2 Introduction

Holography, taking its name from the Greek word $o\lambda o\sigma$ (holos), meaning *whole*, was first introduced in 1948 by Dennis Gabor [20], originally named as *wavefront reconstruction* [21]. It is a technology that can record and reconstruct the wavefront of three-dimensional (3D) objects. Similar to two-dimensional (2D) photography, the earliest holography uses a piece of film to record the diffraction pattern of an object, which can then reconstruct the wavefront showing that object. In order to generate hologram for objects that do not

physically exist, computer-generated holography (CGH) emerged, where a hologram can be calculated through various algorithmic approaches and then displayed on a spatial light modulator (SLM) modulating a coherent light source, in order to reconstruct target images [15, 22, 23, 19], either 2D or 3D, which can provide full depth cues at arbitrary angles instead of stereoscopic displays which need to re-compute the left-eye and right-eye images every time the position changes. So the multi-depth reconstruction ability is a major advantage of the holography technology.

Currently available spatial light modulators can only modulate either phase or amplitude, so algorithms are needed to compute amplitude-only or phase-only holograms, among which the phase-only holograms are usually preferred due to its higher energy efficiency. The classic algorithms include direct binary search [22], simulated annealing [47] and Gerchberg-Saxton [19], which is still widely used despite it being nearly 50 years old. With the developments in modern numerical optimization methods and computation power, advances in CGH algorithms can be made. Literature review has found some recent work that compute CGH using numerical optimization methods [53–55, 17, 3], but speed and quality are still the major challenges in multi-depth hologram generation. The most common multi-depth CGH optimization methods either evaluate their loss against the entire multi-depth 3D target, which is time-consuming, or evaluate the hologram for each plane and then sum the holograms, which introduces quality degradation.

This paper therefore extends on the previous research, which proved the ability of L-BFGS algorithm to generate phase-only hologram for a 2D image [2], and proposes the sequential slicing (SS) technique for the optimization of CGH for multi-depth 3D target, which evaluates the loss for a single slice at each iteration, aiming for quicker hologram generation with proper overall quality and low quality imbalance across the multiple depths enabled by the second-order nature of the L-BFGS optimization algorithm. The following sections start from the background knowledge of numerical optimization including L-BFGS algorithm, then introduces and carries out the optimization of multi-depth CGH with sequential slicing (SS) technique, with results analysed.

4.3 Method

4.3.1 L-BFGS Optimization Algorithm Background

Numerical optimization is a very useful tool to find an optimal solution $\mathbf{x}^*$ which minimize an objective function $f(\mathbf{x})$. Any unconstrained optimization problem can be written as:

$$\min_{\mathbf{x}} f(\mathbf{x}) \tag{4.1}$$

and the iteration is given by Eq. (4.2):

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k \tag{4.2}$$

where the positive scalar α_k is the step length and the vector $\mathbf{p}_k$ is the search direction [56]. There are several optimization algorithms to determine the search direction $\mathbf{p}_k$. The widely used Gradient Descent (GD) algorithm, as its name suggests, directly uses the negative of the gradient as the search direction [63]:

$$\mathbf{p}_k = -\nabla f(\mathbf{x}_k) \tag{4.3}$$

In case if the first derivative cannot be evaluated directly, it can be approximated using Eq. (4.4) [56]:

$$\nabla f(\mathbf{x}_k) \approx \frac{f(\mathbf{x}_{k+1}) - f(\mathbf{x}_k)}{\mathbf{x}_{k+1} - \mathbf{x}_k} \tag{4.4}$$

However, GD is a greedy first-order optimization method that only considers the current iteration without any global consideration, so it can be extremely slow on complicated problems [56]. Therefore, Newton's method emerged, as a second-order optimization method, which has a search direction as shown in Eq. (4.5) [64]:

$$\mathbf{p}_k = -\nabla^2 f(\mathbf{x}_k)^{-1} \nabla f(\mathbf{x}_k) \tag{4.5}$$

Methods that use the Newton direction have a fast rate of local convergence, typically quadratic [56]. However, in many applications, including the optimization of CGH, the Hessian matrix $\nabla^2 f(\mathbf{x}_k)$ in Eq. (4.5) cannot be evaluated, and even if evaluable, its inverse would be computationally heavy to calculate. Therefore, a Quasi-Newton method of BFGS

emerged [64], which approximates the inverse of Hessian iteratively, denoted as $\boldsymbol{H}$, following the process described from Eq. (4.6) to Eq. (4.9) [56].

$$denote\ \mathbf{p}_k = -\boldsymbol{H}_k \nabla f(\mathbf{x}_k) \tag{4.6}$$

$$Initiate\ \boldsymbol{H}_0 \leftarrow \frac{\mathbf{y}_k^T \mathbf{s}_k}{\mathbf{y}_k^T \mathbf{y}_k}\mathbf{I} \tag{4.7}$$

$$update\ \boldsymbol{H}_{k+1} = (\mathbf{I} - \rho_k \mathbf{s}_k \mathbf{y}_k^T)\boldsymbol{H}_k(\mathbf{I} - \rho_k \mathbf{y}_k \mathbf{s}_k^T) + \rho_k \mathbf{s}_k \mathbf{s}_k^T \tag{4.8}$$

$$where \begin{cases} \mathbf{s}_k &= \mathbf{x}_{k+1} - \mathbf{x}_k \\ \mathbf{y}_k &= \nabla f(\mathbf{x}_{k+1}) - \nabla f(\mathbf{x}_k) \\ \rho_k &= \frac{1}{\mathbf{y}_k^T \mathbf{s}_k} \end{cases} \tag{4.9}$$

Then, aiming to save memory usage for large scale problems, L-BFGS algorithm was invented [60], which is listed in Algorithm 8. L-BFGS algorithm stores the Hessian matrix implicitly and compute the search direction from the m most recent iterations. L-BFGS algorithm is a quasi-Newtonian method which determines the gradient with curvature information [60], from the gradient history.

Algorithm 8 L-BFGS Algorithm [60]

$\mathbf{q} \leftarrow \nabla f_k$
for $i = k-1, k-2, ..., k-m$ **do**
  $\alpha_i \leftarrow \rho_i \mathbf{s}_i^T \mathbf{q}$
  $\mathbf{q} \leftarrow \mathbf{q} - \alpha_i \mathbf{y}_i$
end for
$\mathbf{r} \leftarrow \boldsymbol{H}_0 \mathbf{q}$
for $i = k-m, k-m+1, ..., k-1$ **do**
  $\beta \leftarrow \rho_i \mathbf{y}_i^T \mathbf{r}$
  $\mathbf{r} \leftarrow \mathbf{r} + \mathbf{s}_i(\alpha_i - \beta)$
end for
Step with $\mathbf{p}_k \leftarrow -\boldsymbol{H}_k \nabla f(\mathbf{x}_{k+1}) = -\mathbf{r}$

The optimization algorithms used for CGH in this paper are GD and L-BFGS. The phase-only constraint of CGH can be easily applied by fixing a constant amplitude of the hologram, while keeping its phase varying and being the argument of optimization ($\mathbf{x}$).

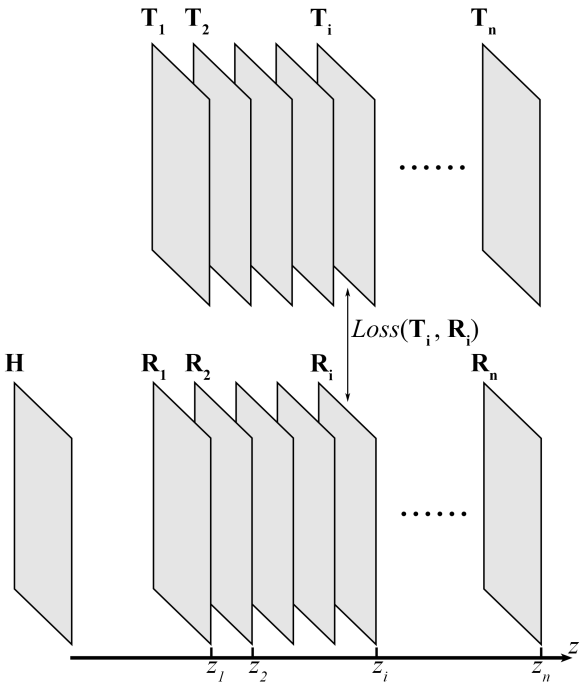


Fig. 4.1 Loss between the multi-depth targets ($\mathbf{T}_1$ to $\mathbf{T}_n$) and the reconstructions ($\mathbf{R}_1$ to $\mathbf{R}_n$) of hologram $\mathbf{H}$

4.3.2 Hologram Optimization for Multi-Depth Targets

As shown in Fig. 4.1, the multi-depth target is set up as a collection of n slices ($\mathbf{T}_1$ to $\mathbf{T}_n$), each slice $\mathbf{T}_i$ is at a distance z_i to the hologram plane. And for the hologram $\mathbf{H}$, its reconstruction at each distance z_i is computed using Fresnel diffraction formula in Eq. (6.1), which is labelled as the propagation function $\mathscr{P}(\mathbf{H}, z_i)$.

Then, to create an objective function ($f(\mathbf{x})$ in Eq. (4.1)) to optimize, we need to quantify the difference between each target slice ($\mathbf{T}_i$) and the respective reconstruction ($\mathbf{R}_i$) numerically, and minimize such difference. Among the lots of options available, the loss functions selected are mean squared error (MSE) [65], cross entropy (CE) [66] and relative entropy (RE) [67]. To adapt the loss functions for two-dimensional (2D) target image $\mathbf{T}_i$ and reconstructed image $\mathbf{R}_i$ of dimension $X \times Y$, the loss functions are adapted as shown in Eq. (4.10) to Eq. (5.2).

$$MSE(\mathbf{T}_i, \mathbf{R}_i) = \frac{1}{X \times Y} \sum_{x=1}^{X} \sum_{y=1}^{Y} (\mathbf{T}_{i;x,y} - \mathbf{R}_{i;x,y})^2 \tag{4.10}$$

$$CE(\mathbf{T}_i, \mathbf{R}_i) = -\sum_{x=1}^{X} \sum_{y=1}^{Y} \mathbf{T}_{i;x,y} \log(\mathbf{R}_{i;x,y}) \tag{4.11}$$

$$RE(\mathbf{T}_i, \mathbf{R}_i) = -\sum_{x=1}^{X} \sum_{y=1}^{Y} \mathbf{T}_{i;x,y} \log\left(\frac{\mathbf{R}_{i;x,y}}{\mathbf{T}_{i;x,y}}\right) \tag{4.12}$$

Mean squared error (MSE) is adapted as shown in Eq. (4.10). MSE is a traditional metric averaging the squared error between the target and observed values. Cross entropy (CE) is adapted as shown in Eq. (4.11). CE is often used in classification problems, such as language modelling [68]. Relative entropy (RE), also called Kullback-Leibler divergence (usually denoted as $D_{KL}(P||Q)$), is adapted as shown in Eq. (5.2). For the uniformity of symbols in this paper, relative entropy is denoted as RE. It is a measure of how much a probability distribution P is different from another probability distribution Q. Both CE and RE are usually computed between the true probabilistic distribution and the predicted probabilistic distribution, while the images are not probability distributions, the pixel values are normalized to decimal numbers in the range of 0 to 1 so that CE and RE can be applied.

The effectiveness of L-BFGS optimization of phase-only CGH for a single slice target image ($n = 1$) has been demonstrated in the previous research [2]. However, for a 3D target consisted of multiple slices at different depths, the optimization of CGH becomes challenging.

The typical technique is to sum the losses computed for each slice for each iteration during optimization, which is called the Sum-of-Loss (SoL) method in this paper. At every iteration, it computes the full 3D reconstructions ($\mathbf{R}_1, \cdots, \mathbf{R}_n$) of the hologram $\mathbf{H}$ at every distance z_i, requiring a total of n Fourier Transforms, and then computes the total loss between all the target slices and the reconstructed slices, therefore fully evaluating the hologram at each step. Although the total number of iterations does not scale with n, the number of Fourier Transforms performed at each iteration scales up with n, as it needs to compute the full multi-depth reconstructions at every iteration.

$$\text{SoL:} \quad \arg\min_{\mathbf{H}} \sum_{i=1}^{n} Loss(\mathbf{T}_i, \mathbf{R}_i) \tag{4.13}$$

Another universal technique for 3D CGH is to sum the sub-holograms $\mathbf{H}_i$ generated for each target slice $\mathbf{T}_i$ to form a total hologram based on the principle of superposition, which is called the Sum-of-Hologram (SoH) method in this paper. For a fixed number of iterations for each sub-hologram, the total computation scales up linearly with the number of slices n. SoH method's advantage is its ease of implementation, that it can compute multi-depth 3D CGH based on any existing single slice CGH algorithm. Its major disadvantage for phase-only hologram generation is that, the final summing of sub-holograms will result in a non-uniform amplitude hologram, and taking the phase of which will result in discarding the amplitude information of the summed hologram, leading to worse reconstructions quality. And also, SoH method suffers from the defocusing effect from one slice to another, causing additional noise.

$$\text{SoH:} \quad \sum_{i=1}^{n} \underset{\mathbf{H}_i}{\arg\min}\, Loss(\mathbf{T}_i, \mathbf{R}_i) \tag{4.14}$$

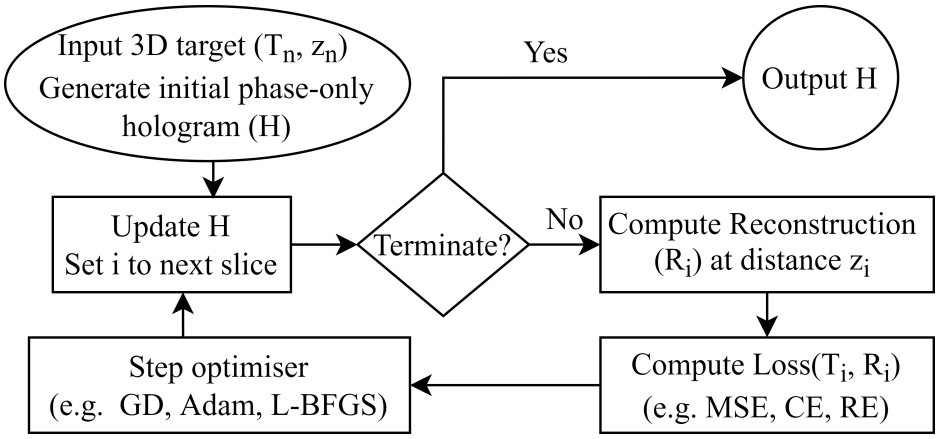


Fig. 4.2 Optimization of CGH with sequential slicing (SS) flowchart

This paper proposes a novel CGH optimization with sequential slicing (SS) technique, as shown in the flowchart in Fig. 4.2, that only computes the loss for a single slice at each iteration (between a reconstruction $\mathbf{R}_i$ at a single distance z_i and its according target slice $\mathbf{T}_i$), where i sweeps through the multi-layer 3D target sequentially when the algorithm iterates. When the final slice is reached ($i = n$), it goes back to the first slice ($i \leftarrow 1$). The proposed method only needs to carry out one Fourier Transform at each iteration, and the number of iterations does not need to scale up with n. So it is expected to be much quicker than SoL and SoH techniques while producing proper resulting hologram.

4.4 Results

4.4.1 CGH for 4-slice targets

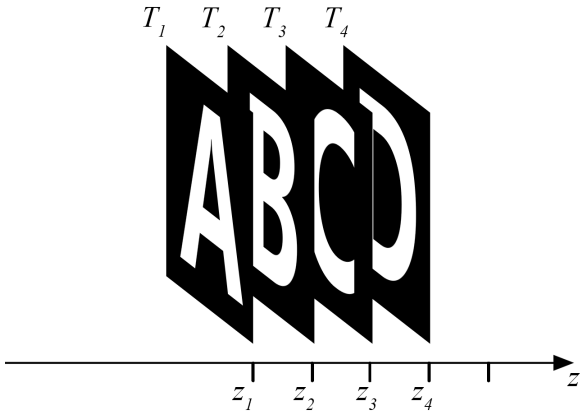


Fig. 4.3 Layout of the 4-slice target ($z_1 = 1$ *cm*, $z_2 = 2$ *cm*, $z_3 = 3$ *cm*, $z_4 = 4$ *cm*)

The first example 3D target used is consisted of 4 slices made from alphabets 'A', 'B', 'C', 'D', each has 512×512 pixels. The positions of the four slices range from 1 *cm* to 4 *cm* with 1 *cm* gap between each other (i.e. $z_i = i$ *cm*). The overall layout is shown in Fig. 4.3.

As there are two optimization algorithms (GD and L-BFGS), three techniques (SoL, SoH and SS) and three loss functions (MSE, CE, RE) in consideration, they can form a total of 18 combinations. In order to control the number of variables, all 18 combinations are set to start from the same initial random hologram and run for the same amount of 100 iterations for the optimization of each hologram, on the same laptop of model ASUS ROG Zephyrus M16, which has a CPU of model i7-11800H and a GPU of model RTX3060. For the L-BFGS algorithm, the gradient history of size 10 ($m = 10$ in Algorithm 8) is used for all techniques and loss functions. And to ensure a sensible comparison, although three different loss functions are used for optimization of CGH, the metric used to assess the quality of multi-depth reconstructions of the final hologram is the normalized mean squared error (NMSE). As there are a total of 4 slices in this example, the final NMSE of each are computed separately and the total optimization run time is recorded. The final results are gathered in Fig. 4.4. As each slice has a different final error, their mean and standard deviation (SD) are computed for investigations. Three columns are colour coded where green indicates better result while red indicates worse result.

Comparing the mean of final NMSE and the run time of the proposed sequential slicing (SS) technique to those of the sum-of-loss (SoL) and sum-of-hologram (SoH) techniques

		Loss	Final NMSE (x10^-6) Slice 1	Slice 2	Slice 3	Slice 4	Mean	SD	Time (s)
L-BFGS	SoL	MSE	1.62	1.23	1.43	1.19	1.37	0.17	1.04
		CE	1.76	1.29	1.52	1.24	1.45	0.21	2.16
		RE	1.62	1.23	1.43	1.19	1.37	0.17	1.02
	SoH	MSE	3.09	2.87	3.31	2.91	3.05	0.17	1.76
		CE	3.05	2.83	3.39	2.87	3.03	0.22	2.25
		RE	3.09	2.87	3.31	2.91	3.05	0.17	1.81
	SS	MSE	2.57	2.52	2.50	2.48	2.52	0.03	0.50
		CE	2.76	2.63	2.63	2.64	2.67	0.05	0.74
		RE	2.57	2.52	2.50	2.48	2.52	0.03	0.47
GD	SoL	MSE	1.94	1.65	1.96	1.66	1.80	0.14	0.86
		CE	2.20	1.94	2.30	1.97	2.10	0.15	1.92
		RE	2.23	1.98	2.35	2.01	2.14	0.15	0.85
	SoH	MSE	3.33	3.06	3.67	3.09	3.29	0.25	0.63
		CE	3.58	3.29	3.96	3.30	3.53	0.27	1.14
		RE	3.59	3.30	3.97	3.31	3.54	0.27	0.59
	SS	MSE	2.53	2.06	2.99	2.15	2.43	0.37	0.28
		CE	2.99	2.57	3.46	2.64	2.92	0.35	0.54
		RE	2.99	2.58	3.46	2.64	2.92	0.35	0.28

Fig. 4.4 Final NMSE and run time comparison across the three techniques

in Fig. 4.4, it can be concluded that, for all combinations of optimizers and loss functions attempted, the proposed SS technique runs much quicker than the existing SoL and SoH techniques, while still providing a proper result, sitting between the SoL and SoH techniques. So the SS technique has not demonstrated absolute advantage to the SoL technique yet on this 4-slice example, therefore further investigation is done on a 30-slice example in Section 4.4.2. However, the SS technique is both quicker and has better reconstructions quality than the SoH technique, demonstrating an absolute advantage. Meanwhile, the combinations with CE as loss function are much slower and has not demonstrated advantage in average NMSE, demonstrating an absolute disadvantage, so the results with CE loss or SoH method will not be shown in the per-iteration plots.

For comparison among the sequential slicing (SS) techniques, the NMSE for each slice and their average and maximum difference values are plotted for GD and LBFGS algorithms with MSE and RE loss functions in Fig. 4.5, and sequential GS and DCGS [50] are also implemented and plotted for reference purpose. Apart from the L-BFGS algorithm, all the other algorithms are showing a staircase-like plot, where a decrease in error on one slice results in an increase in error on all other slices, and so the final NMSE of each slice distinguishes a lot from each other. The sequential GS algorithm suffers the most from the quality imbalance between each slice, and the sequential GD algorithm follows. The DCGS algorithm benefits from its modification of the inclusion of a weighting factor consisting historical amplitude, therefore managed to converge. From the average NMSE plot (Fig. 4.5 (e)), the proposed sequential L-BFGS method does not appear to have the lowest average NMSE, but it has the lowest quality imbalance across the slices as shown in the maximum

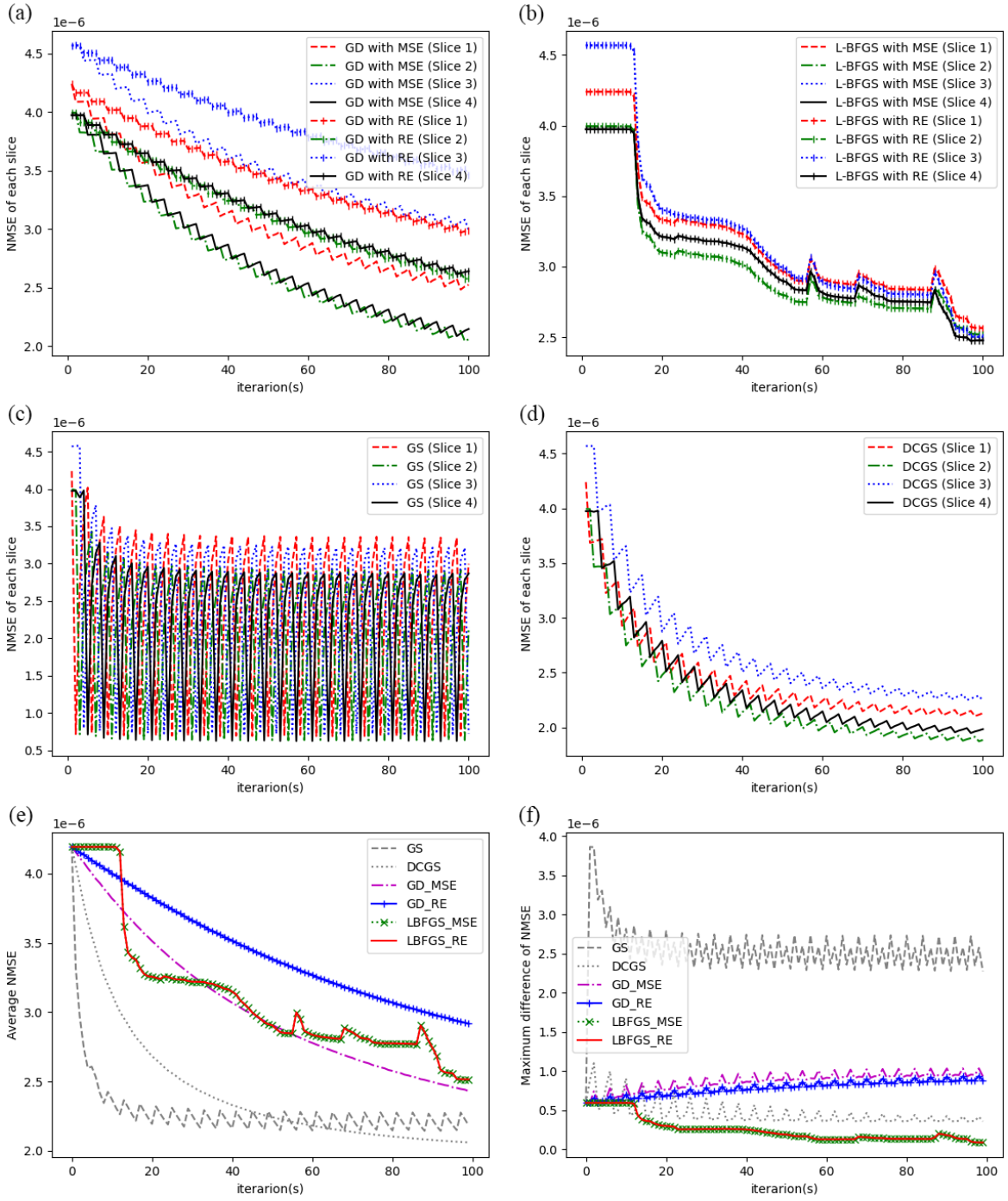


Fig. 4.5 Comparison among SS techniques for the 4-slice target using (a) GD algorithm, (b) L-BFGS algorithm, (c) GS algorithm, (d) DCGS algorithm. (e) Average NMSE, and (f) difference between the maximum and minimum NMSE across all slices.

difference plot (Fig. 4.5 (f)). The L-BFGS algorithm mainly benefits from its inclusion of curvature information during optimization, so that the update of hologram $\mathbf{H}$ at each iteration

takes into account not only the loss for current slice, but also all historical iterations up to the set history size (*m* in Algorithm 8). So the NMSE of each slice stays close for L-BFGS and at each iteration, the NMSE of all slices behave in the same way, ensuring each slice to have the similar quality.

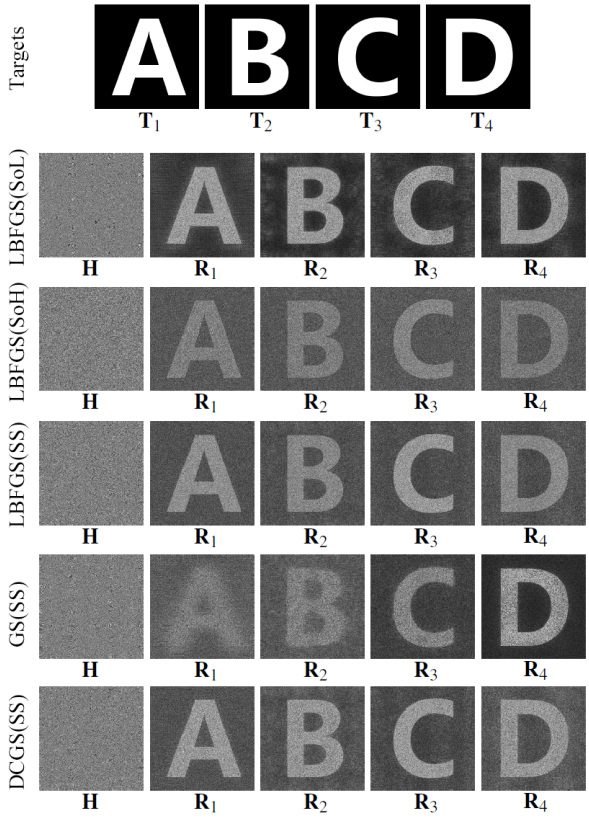


Fig. 4.6 Comparison of final holograms and reconstructions

The final holograms and their reconstructions for L-BFGS algorithm with SoL, SoH and SS techniques are shown in Fig. 4.6, with GS with SS technique and DCGS also shown as reference. The reconstructed images confirm the SS technique having a quality between SoH and SoL method (for the same amount of iterations), and has a much better quality imbalance than sequential GS, which has a very clear reconstruction at the fourth slice (letter 'D') because the iteration stopped at the fourth slice but much worse reconstructions at other slices. Admittedly, the proposed L-BFGS with SS method cannot surpass the GS-based DCGS algorithm yet.

To prove that the proposed method also works for non-binary targets, another example of a 4-slice 3D target was attempted, as shown in Fig. 4.7, where two of the slices are replaced by an image of the mandrill [16] and an image of the city scene [53] respectively.

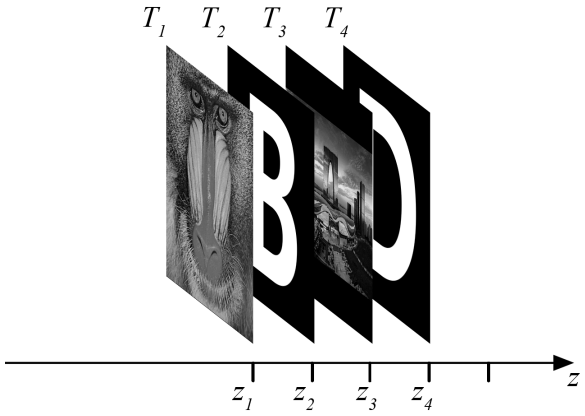


Fig. 4.7 Layout of the non-binary 4-slice target

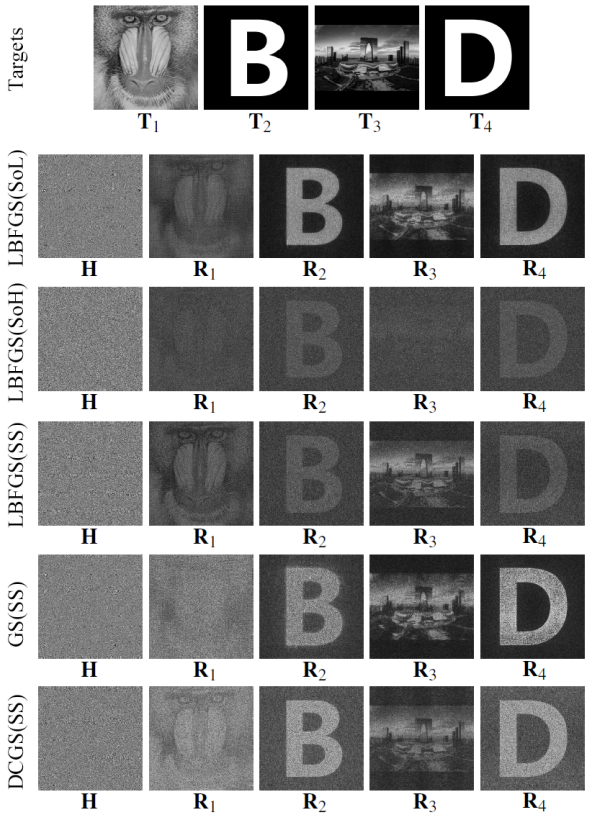


Fig. 4.8 Comparison of final holograms and reconstructions for non-binary target

As shown in the final holograms and reconstructions in Fig. 4.8, the proposed LBFGS with SS technique still managed to converge, with final reconstructions quality sitting between the SoL and SoH method, and also having a good quality balance across all slices.

4.4.2 CGH for a 30-Slice Target

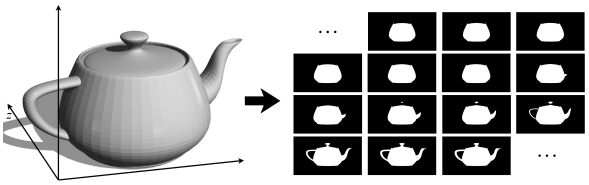


Fig. 4.9 30-slice target sliced from a 3D Teapot mesh

Another example of a 30-slice target sliced from a 3D teapot is attempted for speed analysis when the number of slices go higher. As shown in Fig. 4.9, the Utah teapot [69] is sliced into 30 planes, each of 720p resolution (1280×720). Each combination in Fig. 4.4 were run on a laptop of model ASUS ROG Zephyrus M16 with a CPU of model i7-11800H and a GPU of model RTX3060. The average and maximum difference of NMSE across all slices plotted against time (in Fig. 4.10 and Fig. 4.11 respectively) for all combinations except those with SoH technique or with CE as loss function as they have absolute disadvantage, for clearer plots.

Fig. 4.10 shows that, for both GD and L-BFGS optimization algorithms, the SS technique is faster than the SoL technique. Among the SS techniques, GD, GS and DCGS algorithms achieved less average NMSE than the proposed L-BFGS with SS method, but from the maximum difference of NMSE plot in Fig. 4.11, it can be shown that the L-BFGS algorithm has less quality imbalance than the GD algorithm and the GS algorithm, although admittedly it is not as good as the DCGS algorithm. Nevertheless, the proposed method has achieved an improvement of phase retrieval using optimization algorithms, in speed and quality imbalance suppression.

4.5 Conclusion

This paper has proposed the method of using L-BFGS optimization algorithm to generate phase-only hologram for multi-depth target, and discussed its suitability with the sequential slicing (SS) technique. The L-BFGS with SS method has demonstrated a good suppression on the quality imbalance across the multi-depth slices, benefiting from the nature of being a second order optimization algorithm, which implicitly records the historical gradients by other slices for the determination of the descent direction. For both GD and L-BFGS optimization algorithms, the SS technique ran faster and produced better reconstruction quality than the simple SoH technique, and it is quicker than the SoH technique when the

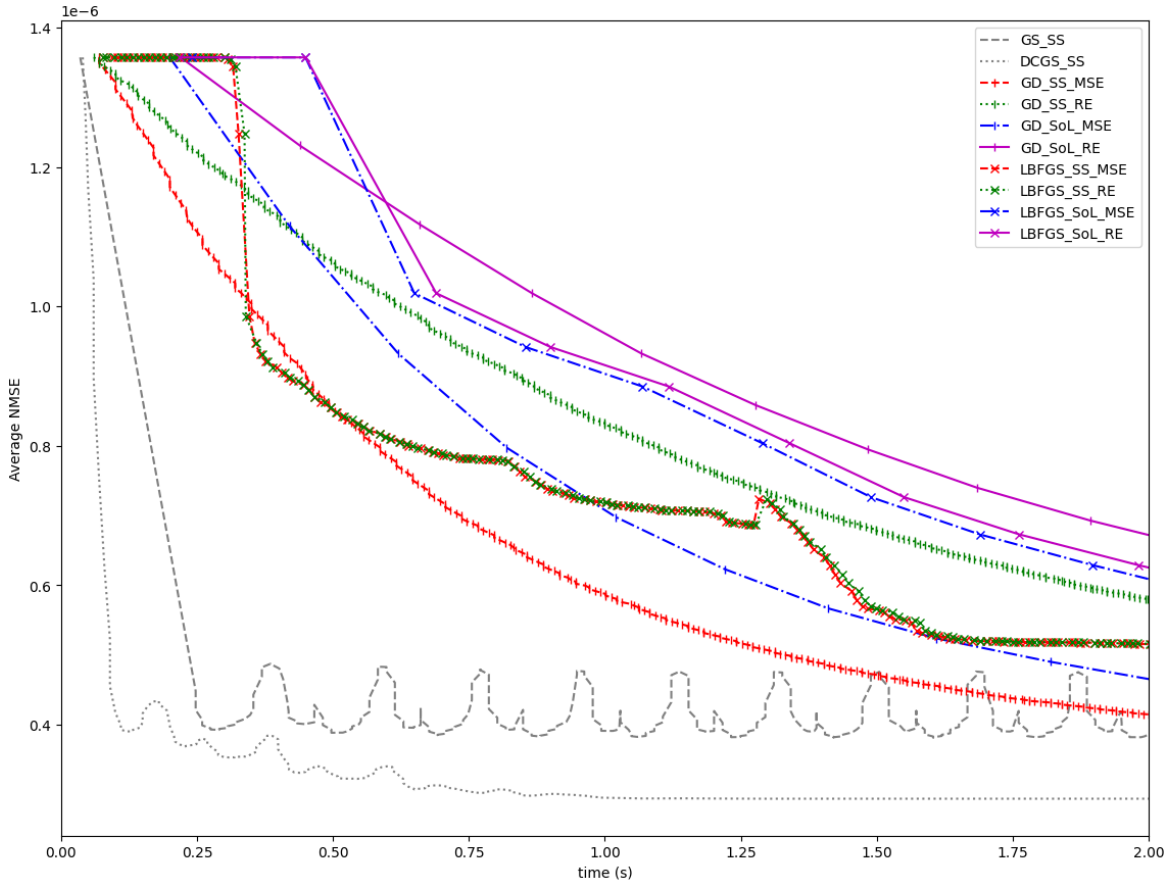


Fig. 4.10 Average NMSE v.s. time plot for the 30-slice target

number of depths get large. The proposed method has demonstrated great potential of time constrained optimization of multi-depth CGH.

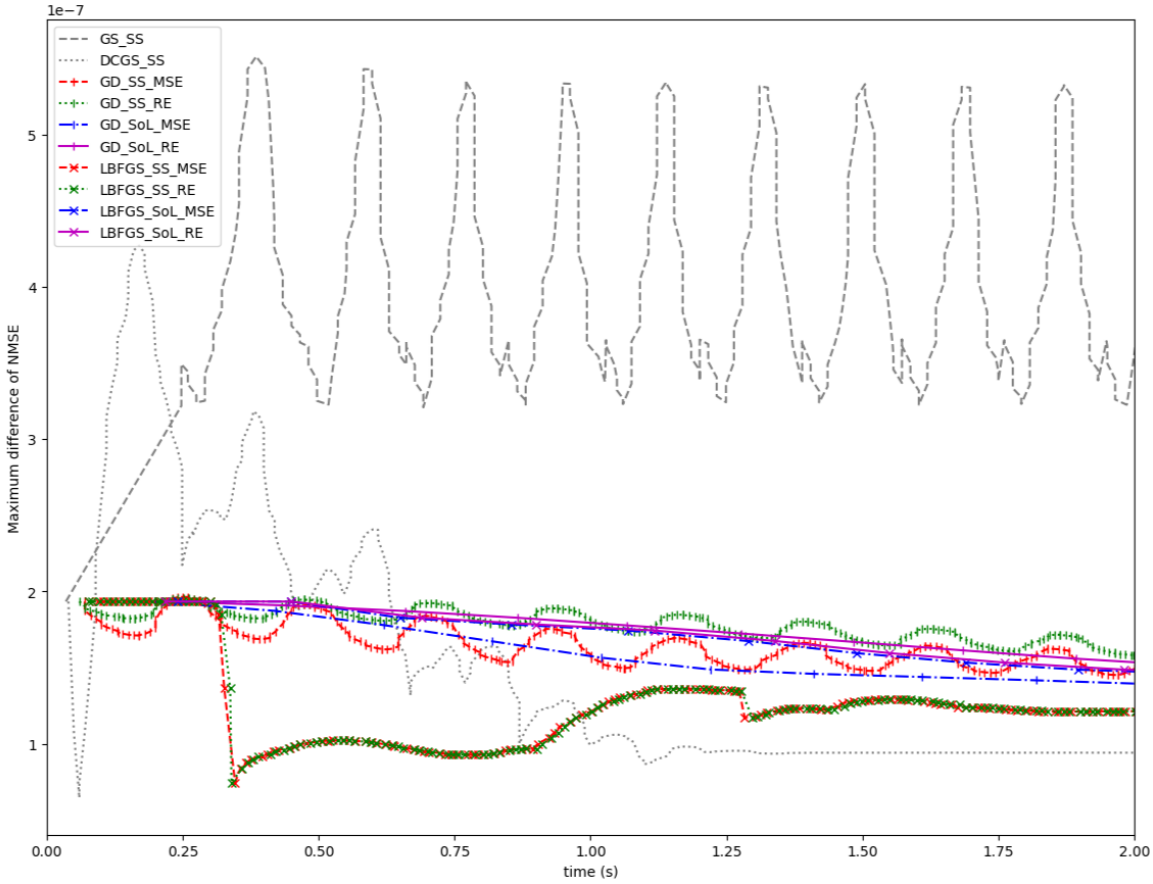


Fig. 4.11 Difference between the maximum and minimum NMSE across all slices v.s. time plot for the 30-slice target

Chapter 5

Multi Frame Holograms Batched Optimization for Binary Phase Spatial Light Modulators

5.1 abstract

Phase retrieval methods used in computer generated holograms such as Gerchberg-Saxton and gradient descent give results which are prone to noise and other defects. This work builds up on the idea of time-averaging multiple hologram frames, first introduced in methods like One-Step Phase-Retrieval and Adaptive One-Step Phase-Retrieval. The proposed technique called Multi-Frame Holograms Batched Optimization uses the L-BFGS optimization algorithm to simultaneously generate a batch of binary phase holograms which result in an average reconstructed image of improved fidelity and fast algorithmic convergence, both in the Fraunhoffer and the Fresnel regimes. The results are compared to One-Step Phase-Retrieval and Adaptive One-Step Phase-Retrieval in simulation and experimentally, proving the superiority of the proposed approach. This technique can be easily extended to other spatial modulation methods.

5.2 Introduction

Computer-generated hologram (CGH) enables producing three-dimensional (3D) multi-depth image reconstruction via modulating the wavefront of a coherent light source. Currently

available spatial light modulators (SLM) can only modulate either phase or amplitude, so algorithms are needed to compute amplitude-only or phase-only holograms. The classic phase-retrieval algorithms include direct binary search [22], simulated annealing [47] and Gerchberg-Saxton [19]. With the developments in modern numerical optimization methods and increase in computational power, phase retrieval with new numerical optimization methods has also been found in the literature such as: gradient descent [53, 54], its stochastic variations [17, 55, 3], and its derivative L-BFGS [2, 4]. However, all of these are single-frame hologram generation methods. In contrast, time multiplexing multi-frame holograms seeks to improve a time-averaged response by displaying different hologram sub-frames at a high refresh rate [70]. Such approach can exploit the finite response time of human vision, where human eyes average out the unwanted noise while the wanted signal remains. Similarly, such method could be used in holographic systems which require high precision without any hard restrictions on the projection refresh rate, such as holographic photo-lithography [71]. A few time-multiplexed multi-frame holograms generation methods have been explored in the literature, including the One-Step Phase-Retrieval (OSPR) algorithm [35] and the Adaptive One-Step Phase-Retrieval (AD-OSPR) algorithm [72]; however, both OSPR and AD-OSPR are still subject to defects in reconstruction quality.

This paper therefore extends on the previous research using the L-BFGS optimization algorithm for single-frame phase-only hologram generation [2, 4], and proposes a novel time-multiplexed multi-frame holograms generation method using L-BFGS optimization, called Multi-Frame Holograms Batched Optimization (MFHBO), to produce better reconstruction quality than the existing OSPR and AD-OSPR methods.

5.3 Method

This paper proposes a novel method of using numerical optimization algorithm L-BFGS [60] to generate multi-frame binary-phase holograms. The L-BFGS algorithm had previously been used for single-frame hologram optimization [2, 4]. To implement it onto multi-frame holograms generation, the argument to vary becomes the set of holograms with n sub-frames ($\{\mathbf{H}_1, \mathbf{H}_2, ..., \mathbf{H}_n\}$), each having a resolution of $X \times Y$ pixels matching the resolution of the target image, and the objective function to minimise is therefore the difference between the average reconstruction amplitude ($\mathbf{R}_{avg} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{R}_i$) and the target image ($\mathbf{T}$), which is denoted as $Loss(\mathbf{T}, \mathbf{R}_{avg})$, where n is the total number of frames, $\mathbf{R}_i$'s are reconstructions

from individual hologram sub-frames $\mathbf{H}_i$'s for $i \in [1,n]$. To compute each $\mathbf{R}_i$ from the corresponding $\mathbf{H}_i$, we start from the Fresnel diffraction formula given in Eq. (6.1) [39]

$$\mathbf{E}_{Fresnel\ region}(\alpha,\beta,z) = \mathscr{F}\left\{\mathbf{A}(x,y)\cdot e^{j\frac{k}{2z}(x^2+y^2)}\right\} \tag{5.1}$$

where $\mathbf{E}$ is the reconstructed electric field, in complex form, $\mathbf{A}$ is the hologram aperture, also in complex form, and $\mathscr{F}$ denotes the Fourier Transform, implemented on computers using the Fast Fourier Transform (FFT) function. As eyes cannot perceive phase, the reconstruction amplitude is therefore the absolute value, giving $\mathbf{R} = |\mathbf{E}|$. And as we are generating holograms for phase-only SLM's, $\mathbf{A}$ is then comprised of a uniform amplitude with phase $\mathbf{H}$, giving $\mathbf{A} = e^{j\mathbf{H}}$, where the exponential is taken element-wise.

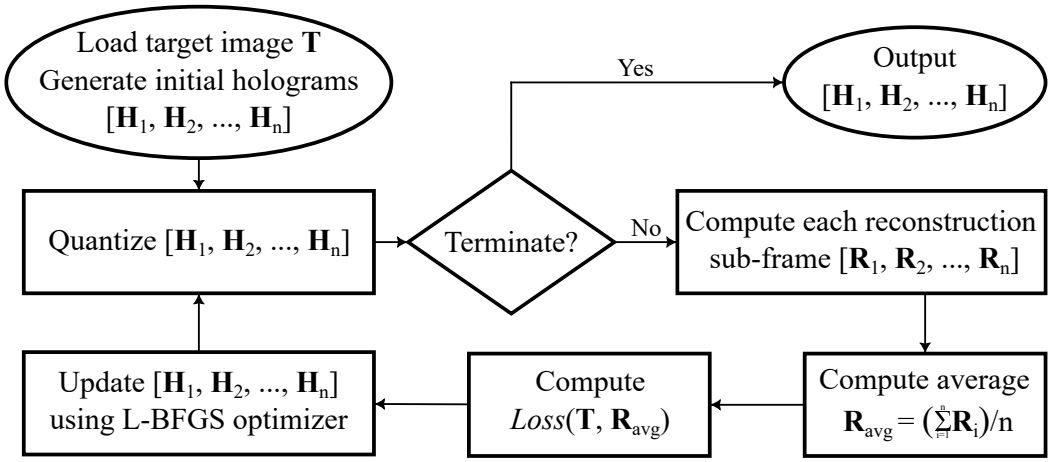


Fig. 5.1 MFHBO flowchart

To help explain the optimization process, a flow chart is drawn in Fig. 5.1. As shown in the flowchart, the target image $\mathbf{T}$ is first loaded, with a set of n hologram sub-frames ($\{\mathbf{H}_1,\mathbf{H}_2,...,\mathbf{H}_n\}$) generated randomly. Then at every iteration, each hologram sub-frame $\mathbf{H}_i$ is quantized to the bit-depth constraint of the SLM, and propagated to the reconstruction plane $\mathbf{R}_i$, and the average of the amplitudes of all reconstructions $\mathbf{R}_{avg}$ is computed and compared against the target image $\mathbf{T}$ using a loss function $Loss(\mathbf{T},\mathbf{R}_{avg})$, after which the search direction is computed using the L-BFGS optimizer and the hologram sub-frames are updated accordingly. Here the loss function selected is the relative entropy[67] given in Eq. (5.2).

$$Loss(\mathbf{T}, \mathbf{R}_{avg}) = -\sum_{x=1}^{X}\sum_{y=1}^{Y} \mathbf{T}_{(x,y)} \log\left(\frac{\mathbf{R}_{avg(x,y)}}{\mathbf{T}_{(x,y)}}\right) \tag{5.2}$$

Since fast SLM's available in the lab are binary-phase devices, the quantization step in the flowchart in Fig. 5.1 is carried out with bit-depth limit of 1, hence producing binary-phase holograms. However, the optimization algorithm does not converge with a straight binary quantization as integers are discrete, therefore a Sigmoid function [73] is used for a smoother and differentiable quantization, as defined in Eq. (5.3). The output of the Sigmoid function is then scaled by π so that the binary phase levels are 0 and π.

$$\text{sigmoid}(x) = \frac{1}{1+e^{-x}} \tag{5.3}$$

And finally, when displaying the multi-frame holograms, each of the n frames generated are then rounded to binary phase values and displayed on the binary phase SLM sequentially. And when the first round finished, the second round starts with the first frame again (i.e. after frame n, the next frame displayed is frame 1), and such infinite loop doesn't stop until another set of holograms are uploaded.

5.4 Results

5.4.1 Simulation results

To test the proposed MFHBO method, a target image $\mathbf{T}$ as shown in Fig. 5.2 was used. It was designed from the widely used mandrill image [16]. A rotational symmetry was introduced to match the rotational symmetric property of the far field projections from binary phase holograms. It was then zero padded to a resolution of $1024px \times 1024px$ and subsequently interpolated to a resolution of $1280px \times 1024px$ to match the resolution of the SLM in our lab. Note that the target image was zero padded to a square aspect ratio and then stretched to the non-square aspect ratio because more pixels in the horizontal axis only means higher sampling rate as part of the features of the FFT, the replay field is continuous and is not pixelated and the simulated reconstruction of $1280px \times 1024px$ resolution is the sampled results, which will be illustrated visually in Fig. 5.5 later.

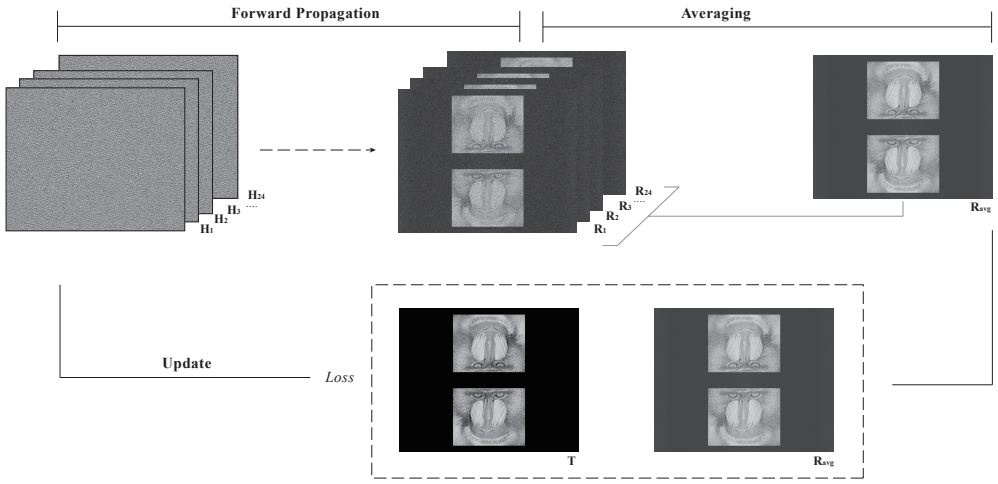
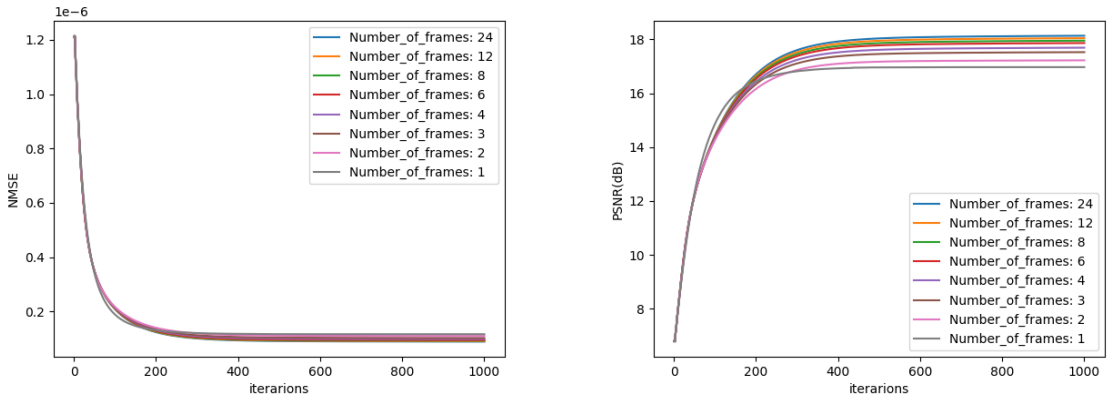


Fig. 5.2 An example iteration in the optimization process

To further explain the optimization process described in Fig. 5.1, an example iteration with $n = 24$ is shown in Fig. 5.2. At each iteration, every hologram is quantized and propagated to the reconstruction plane, forming $\{\mathbf{R}_1, \mathbf{R}_2, ..., \mathbf{R}_{24}\}$. The average reconstruction amplitude $\mathbf{R}_{avg}$ is then compared against the target image $\mathbf{T}$, using the loss function in Eq. (5.2). The holograms $\{\mathbf{H}_1, \mathbf{H}_2, ..., \mathbf{H}_{24}\}$ are then updated according to the search direction calculated using the L-BFGS optimizer. After setting the optimization to terminate when the number of iterations reach 1000, the same algorithm was run on the same target for different number of frames (n), the normalised mean squared error (NMSE) and the peak signal-to-noise ratio (PSNR) between the average reconstructions $\mathbf{R}_{avg}$ and the target image $\mathbf{T}$ were calculated at every iteration and plotted in Fig. 5.3a and Fig. 5.3b respectively.



(a) NMSE v.s. Iteration Number

(b) PSNR v.s. Iteration Number

Fig. 5.3 Convergence of optimization

The plots in Fig. 5.3 show that the proposed MFHBO method has achieved good convergence within 400 iterations, for the various number of frame settings n in $\{1,2,3,4,6,8,12,24\}$. The final NMSE values in Fig. 5.3a are difficult to distinguish in the plot, therefore it will be further compared in the bar chart in Fig. 5.5. The number of frames are chosen to be integer factors of 24, which is determined by our experimental setup, further explained in the next subsection.

Number of frames	200 iterations	400 iterations	600 iterations	800 iterations	1000 iterations
1	1.79	3.59	5.31	7.00	8.72
2	2.82	5.59	8.34	11.11	13.88
3	3.84	7.67	11.45	15.21	19.00
4	4.93	9.83	14.70	19.58	24.47
6	6.95	13.87	20.76	27.58	34.50
8	8.87	17.67	26.54	35.60	44.56
12	12.95	25.79	38.63	51.47	64.30
24	51.81	101.43	151.09	201.08	251.15

Table 5.1 MFHBO runtime (s)

The programme runtime of the proposed MFHBO method has been measured on a laptop computer of model ASUS ROG Zephyrus M16 (GU603H) with a CPU of model i7-11800H and a GPU of model NVIDIA RTX3060 and the results for different combinations of number of frames and number of iterations are listed in Table 5.1. It can be concluded that the application of the proposed method is for pre-computed high-quality holograms, instead of real-time holographic projections.

5.4.2 Optical Experiment results

The holographic projection system used in this experiment is the same as the one used in previous research [5], which was originally developed by Freeman [18]. The optical setup is shown in Fig. 5.4. The design is consisted of a diode-pumped solid-state (DPSS) 532 nm 50mW laser source, focused down by an aspheric singlet, and passed through a polarising beam splitter cube to a collimating lens, which illuminates the SLM [18]. The SLM is a binary phase SXGA-R2 ForthDD ferroelectric Liquid crystal on silicon (LCOS) micro-display with a refresh rate of 1440Hz, a pixel pitch of 13.6 µm and a resolution of 1280×1024 [18]. Since the SLM has a refresh rate of 1440Hz and modern computer monitors have refresh rate of at

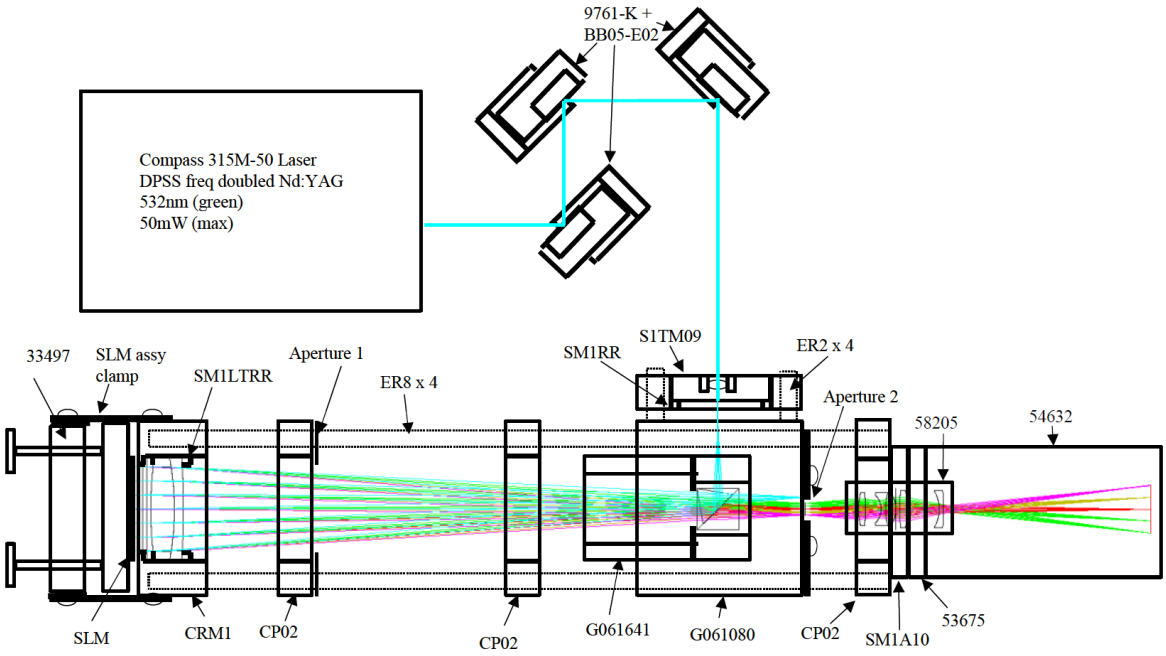


Fig. 5.4 Holographic projection system components [18]

least 60Hz, the maximum number of frames was chosen to be $1440/60 = 24$, so that each set of 24 frames will take a total of $1/60$ seconds to display, therefore giving an equivalent refresh rate of 60Hz. Then the integer factors of 24 were chosen so that the equivalent refresh rate becomes integer multiples of 60Hz. The number of frames starts from 1 to help illustrate how the increase in number of frames positively affect the reconstruction quality.

The results in Fig. 5.5 further compares the final results for different number of frames. The histogram in Fig. 5.5 shows that, as the number of frames increases, the NMSE between the average reconstructions $\mathbf{R}_{avg}$ and the target image $\mathbf{T}$ decreases and the structural similarity index (SSIM)[46] increases, showing a trend of better reconstruction quality with higher number of frames. Such trend is expected as more frames provide higher information capacity, which agrees with the previous research where holograms with higher bit depth were found to achieve better reconstruction quality [8]. The trend is also shown visually via the simulation results and their detail enlargements. The corresponding multi-frame holograms are then loaded onto the SLM, and the reconstructed field is captured using a camera of model Cannon EOS 1000D. Only the bottom halves of the reconstructed field were captured as the symmetrical conjugates were unwanted feature of far field projections from binary-phase SLM's. The raw data including multi-frame binary-phase holograms, simulated reconstructions and optical results captured are accessible in the database [74].

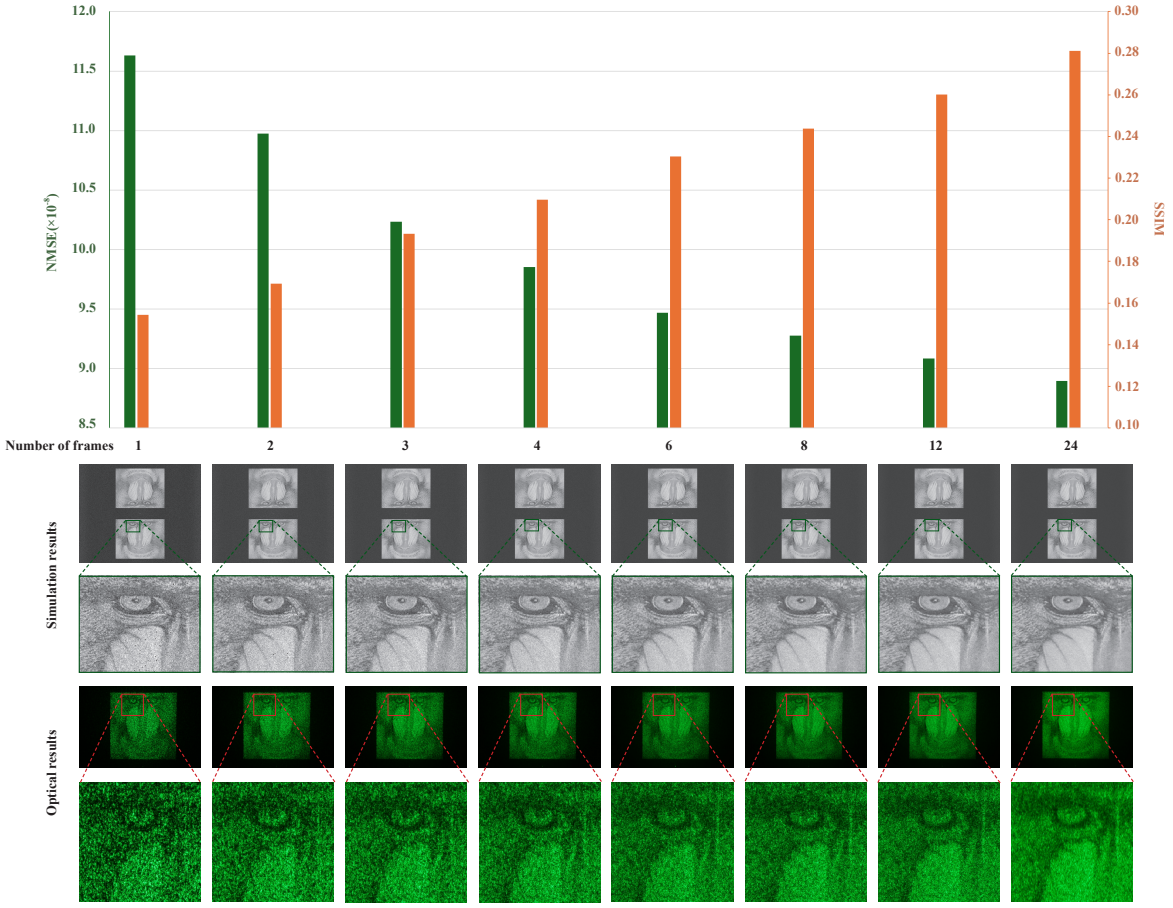


Fig. 5.5 Simulation and optical reconstruction results for different number of frames

Then another target image was tested, which is the holography ambigram shown as shown in Fig. 5.6.[1] The term ambigram is used to refer to (often typographical) designs that are invariant under a reflection, rotation or other symmetry. The 'holography' design contains 180-degree rotational symmetry, which makes it especially well suited to binary Fourier-holographic projection, where this symmetry is unavoidable. Multi-frame holograms were then generated using the proposed MFHBO method and the existing OSPR and AD-OSPR methods, for the same number of frames $n = 24$. And the optical results are shown in Fig. 5.7.

As shown in Fig. 5.7, for the Mandrill target image, it can be seen that the proposed MFHBO method achieved a much better optical reconstruction quality than the existing OSPR and AD-OSPR methods, with clearer details and better contrasts; for the 'holography' ambigram target image, the proposed MFHBO method is shown to have a much lower background noise around the centre, than the existing OSPR and AD-OSPR methods. The intended black

[1] Adapted, with colours reversed, from *holography* - Benjamin Wetherfield, 2022.

Fig. 5.6 Sample target image - ‘holography’ ambigram

regions are represented much more cleanly, with an elimination of speckle-like artefacts in the zero-valued space around the lettering, and an overall increase in discernible contrast.

Image	**Metric**	**MFHO**	**OSPR**	**AD-OSPR**
Mandrill	NMSE ($\times 10^{-4}$)	0.84	1.00	1.12
	SSIM	0.124	0.076	0.078
Ambigram	NMSE ($\times 10^{-5}$)	2.29	3.31	3.23
	SSIM	0.795	0.826	0.827

Table 5.2 Quantitative analysis of the optical results in Fig. 5.7

A quantitative analysis was then conducted on the optical results in Fig. 5.7, the NMSE and SSIM between the captured reconstructions and there corresponding targets are computed and listed in Table 5.2. The NMSE results of the proposed MFHBO method are lower than those of the existing OSPR and AD-OSPR methods, with a 25% reduction on average among both target images. On the other hand, the SSIM results have shown a 62% increase using MFHBO than OSPR and AD-OSPR for the mandrill target image, but a slight decrease of 3.7% for the ‘holography’ ambigram target image, which is negligible as it is less than 5% and the SSIM metric is not originally designed for binary-valued non-grayscale images.

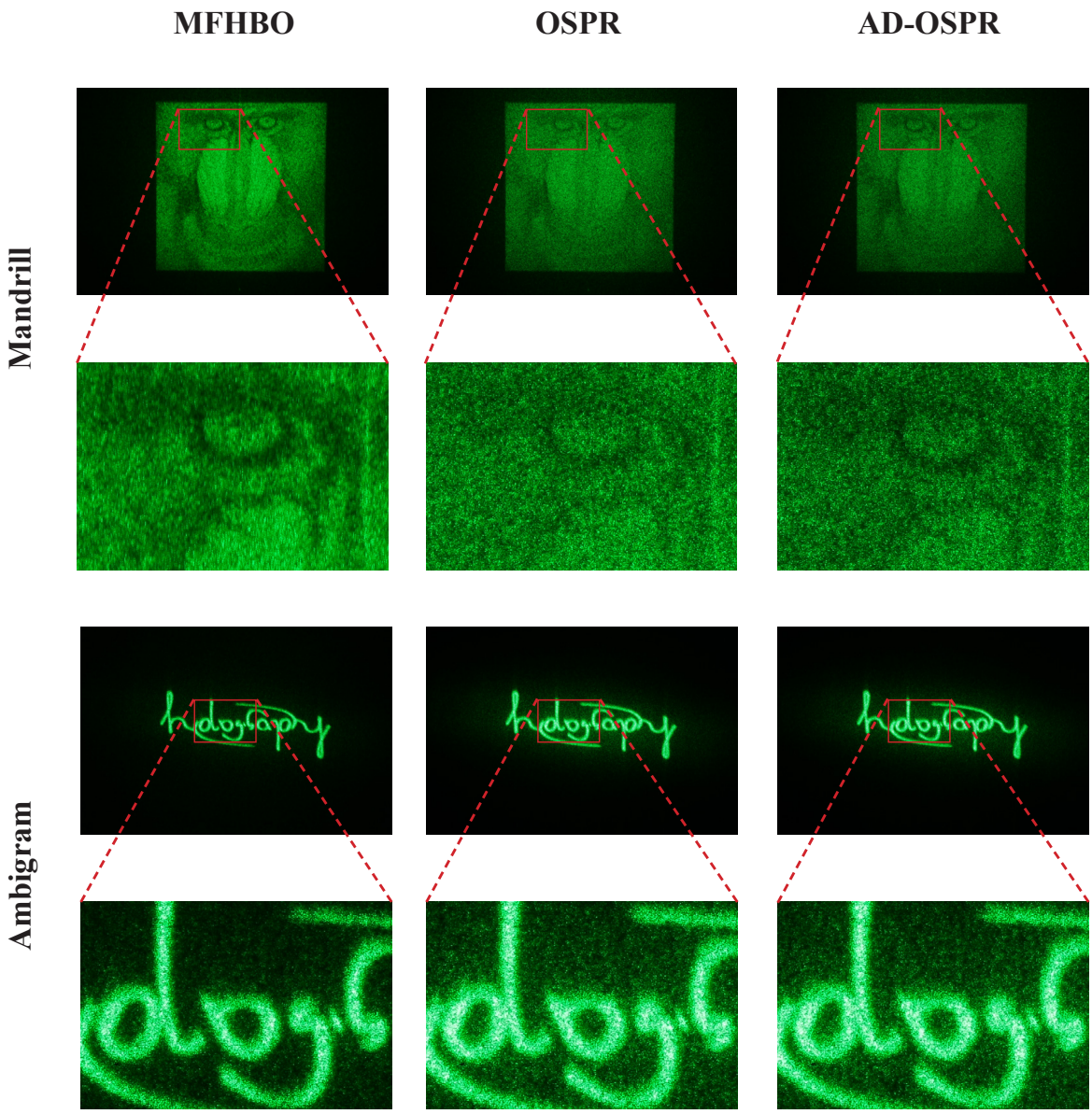


Fig. 5.7 Optical results comparison of the proposed MFHBO method against the existing OSPR and AD-OSPR methods

3D Holography

The proposed MFHBO method was extended to multi-slice targets, by computing the loss between all 4 slices of reconstructions and target images (the Sum-of-Loss method in [4]). An example 4-slice target made from alphabets 'A, B, C, D' is shown in Fig. 5.8. The z values, corresponded to the z variable in Eq. (6.1), were chosen to be $1.1, 1.9, 3.5, 7.7$ for the 4 slices respectively (as there's no correlation between each slice, larger separation was chosen for fewer cross-talks across different planes). It can be seen that the proposed

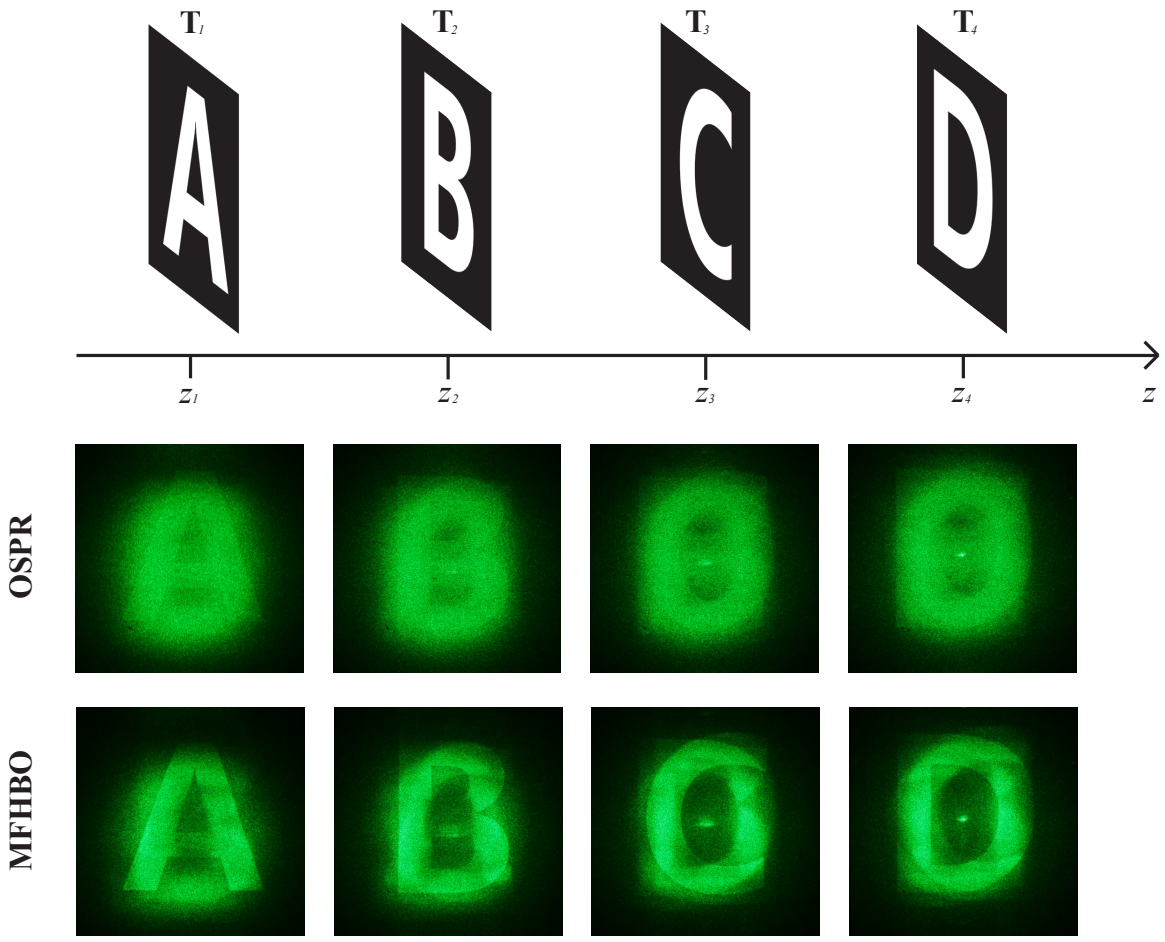


Fig. 5.8 4-slice target and according reconstruction results

MFHBO method has produced sharper edges in reconstructions than the existing OSPR method. (The AD-OSPR method was not attempted here as its application to multi-slice targets was not defined).

Method	Metric	Slice 1	Slice 2	Slice 3	Slice 4	Average
OSPR	NMSE($\times 10^{-4}$)	4.980	4.484	5.644	4.846	4.988
	SSIM	0.072	0.061	0.048	0.060	0.060
MFHO	NMSE($\times 10^{-4}$)	4.484	4.230	4.990	4.289	4.498
	SSIM	0.063	0.067	0.058	0.072	0.065

Table 5.3 Quantitative analysis of the optical results in Fig. 5.8

Then a quantitative analysis was carried out, with NMSE and SSIM values measured and shown in Table 5.3. The proposed MFHBO method has shown a 10% reduction in NMSE and a 8% improvement in SSIM on average than the existing OSPR method, demonstrating the effectiveness of the proposed method.

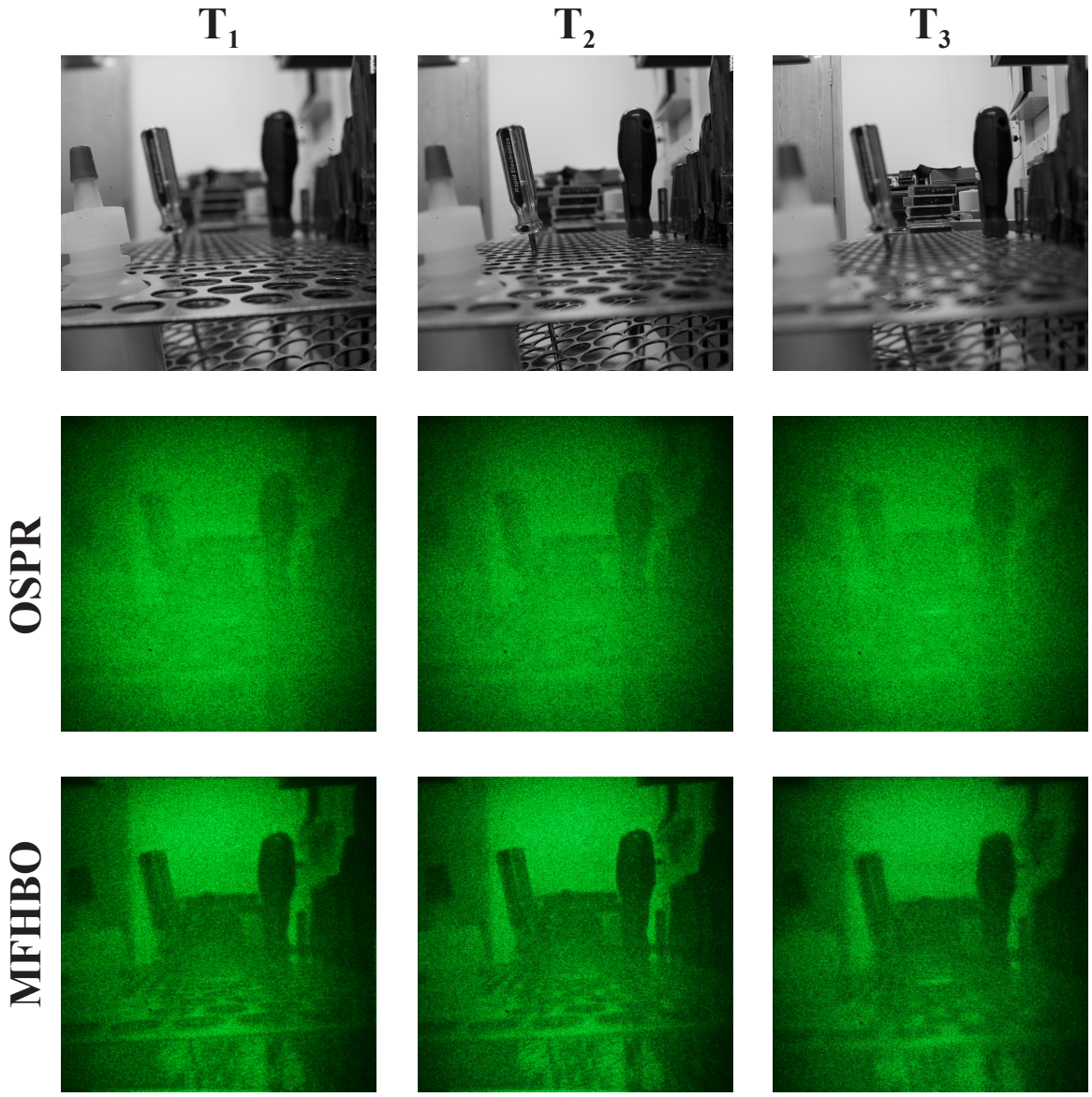


Fig. 5.9 Real-life captured image as target field and their reconstruction results

Lastly, a set of real-life scene was captured in the lab using near, middle and far focus, as shown in $\mathbf{T}_1, \mathbf{T}_2, \mathbf{T}_3$ in Fig. 5.9 respectively. The z values were set to $1.1, 1.2, 1.3$ for hologram generation, and the reconstruction results of the existing OSPR and the proposed

MFHBO methods are compared in Fig. 5.9. The proposed MFHBO method is shown to have achieved much better reconstruction quality than the existing OSPR method.

Method	Metric	Slice 1	Slice 2	Slice 3	Average
OSPR	NMSE($\times 10^{-6}$)	3.70	3.69	3.47	3.62
	SSIM	0.37	0.28	0.34	0.33
MFHO	NMSE($\times 10^{-6}$)	3.20	2.78	3.06	3.01
	SSIM	0.42	0.32	0.32	0.35

Table 5.4 Quantitative analysis of the optical results in Fig. 5.9

A quantitative analysis was conducted again, with NMSE and SSIM values measured and listed in Table 5.4. The proposed MFHBO method has shown a 17% reduction in NMSE and a 7% improvement in SSIM on average than the existing OSPR method, proving the effectiveness of the proposed method.

5.5 Conclusion

This paper proposed the MFHBO method to generate multi-frame binary-phase holograms to be displayed on high refresh rate binary-phase SLM. The proposed MFHBO method was shown to achieve much better reconstruction quality and higher contrast than the existing multi-frame binary-phase holograms generation methods OSPR [35] and AD-OSPR [72] on the holographic projector with binary-phase SLM, for all the single-slice far-field targets and the multi-slice near-field targets tested. Although the propose MFHBO method is slower than the existing OSPR and AD-OSPR methods, its much better reconstruction quality makes it suitable for pre-computed high-quality hologram applications. Its strong advantage for high contrast target such as the ‘holography’ ambigram, with much suppressed speckle noise in the background, makes it well-suited for photo-lithography applications. The proposed method can also be adapted for multi-level SLM’s by simply removing the quantization step (in Fig. 5.1). This could be the case for applications such as photo-lithography, where the time response of the system is much longer than it is for human vision, and the high refresh rates of the SLM are not necessary.

Chapter 6

Information capacity of phase-only computer-generated holograms for holographic displays

6.1 abstract

Despite many years of development in computer-generated holography, perfect phase-only holograms for most target images are still yet possible to compute. All computational phase retrieval algorithms end up with some error between the target image and the reconstruction of the computer-generated hologram (CGH), except for specific targets. This research focuses on the fundamental limits of phase-only CGH quantized to limited bit-depth levels, from the information theory point of view, revealing the information capacity of CGH and its effect on reconstruction quality, with an attempt to quantify how hard a target image is for phase-only hologram computation.

6.2 Introduction

Holography is a technology that can fully reconstruct the wavefront of three-dimensional (3D) objects. Computer-generated holography (CGH) is a technique for converting a 3D object scene into a two-dimensional (2D), complex-valued hologram [75], without the need for the 3D object to physically exist. However, despite many advancements in liquid crystals and micro mirrors technologies, complex-valued spatial light modulators (SLM) are still

not yet available, and the currently available SLM can only achieve either amplitude or phase modulation, among which the phase modulation is usually preferred in holographic projections for its lower zero order and higher energy efficiency due to lower blockage of light. There are many algorithms available to compute good quality phase-only holograms, such as direct binary search [22], simulated annealing [47], Gerchberg-Saxton [19] and other optimization based methods [53–55, 17, 3, 4]; however, none of them can guarantee to compute a perfect phase hologram for a 3D or even a 2D image, where they always end up with some error between the reconstruction of the hologram and the target scene, especially when the phase holograms are quantized to be able to display on SLM with limited bit depth. An intuition therefore arose that the bit depth of the phase hologram is limiting the reconstruction quality, and that the target scene's entropy seems to denote how difficult it is for phase hologram generation. It is obvious that the entropy of the target can certainly never exceed the bit depth limit of its corresponded perfect hologram, otherwise a lossless compressor breaking the Shannon's information theory [76] would be invented; however, such statement is not quite useful in practice as perfect holograms are generally not even possible to compute.

A literature review had found no related work on the information entropy of computer-generated holograms, with the closest match being the research on hologram compression using optical method to achieve lossy compression [77], which cannot be integrated in CGH processes. Therefore, this paper aims to investigate how much information content can a bit-depth-limited phase hologram contain, taking from an information theory point of view. Previously, work had been done to investigate the effect of Bit-depth in Stochastic Gradient Descent (SGD) performance for phase-only computer-generated holography displays [3]. This paper extends on previous research onto the Gerchberg-Saxton (GS) [19] algorithms for hologram generation, and investigates the correlation between the quality of the reconstructed image from the hologram and the information entropy of the target image, with an aim to reduce the hologram's entropy during the CGH process, for smaller size holograms.

6.3 Methods

6.3.1 Computation of Diffraction

To compute the spatial diffraction of light, the Fresnel diffraction formula is used, as shown in Equation (6.1) [39].

$$E_{Fresnel}(\alpha,\beta,z) = \mathscr{F}\left\{A(x,y)e^{j\frac{k}{2z}(x^2+y^2)}\right\} \tag{6.1}$$

where the hologram aperture is denoted by $A(x,y)$ and the diffracted field is denoted by $E(\alpha,\beta,z)$, with z being the distance from the hologram aperture $A(x,y)$, and j is $\sqrt{-1}$, k is the wave number ($k = \frac{2\pi}{\lambda}$, where λ is the wavelength), and $\mathscr{F}$ denotes the Fourier Transform. The amplitude of $E(\alpha,\beta,z)$, which is $|E(\alpha,\beta,z)|$, is referred to as the reconstruction of the hologram throughout this paper.

When $z \gg \frac{k(x^2+y^2)_{max}}{2}$ (i.e. in the far field), Equation (6.1) can be further approximated to give Equation (6.2), which is the Fraunhofer diffraction formula [39].

$$E_{Fraunhofer}(\alpha,\beta) = \mathscr{F}\{A(x,y)\} \tag{6.2}$$

To implement the formula on computers, both $A(x,y)$ and $E(\alpha,\beta)$ are discretized, just like digital images. The Fourier Transform $\mathscr{F}$ is discretized using the Fast Fourier Transform (FFT); similarly the inverse Fourier Transform $\mathscr{F}^{-1}$ is discretized using the Inverse Fast Fourier Transform (IFFT) [78].

6.3.2 Computer-Generated Hologram (CGH) Algorithm

To compute phase-only holograms, the Gerchberg-Saxton (GS) [19] algorithm, which is the classic and robust phase retrieval algorithm, is selected. Taking the Fraunhofer propagation in Equation (6.2) as an example, the operation flowchart of GS algorithm is illustrated in Fig. 6.1. GS algorithm functions that it iteratively determines the phase profile ($\angle A$) of the hologram (A) required to reconstruct a target image (T); it loops between the hologram (A) and the diffracted plane (E), and applying constraints to each plane accordingly during each iteration [19], with the total number of iterations denoted by N.

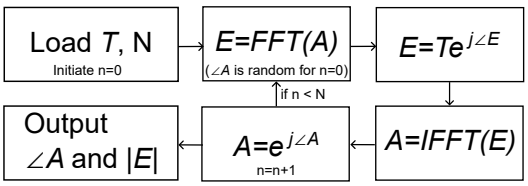
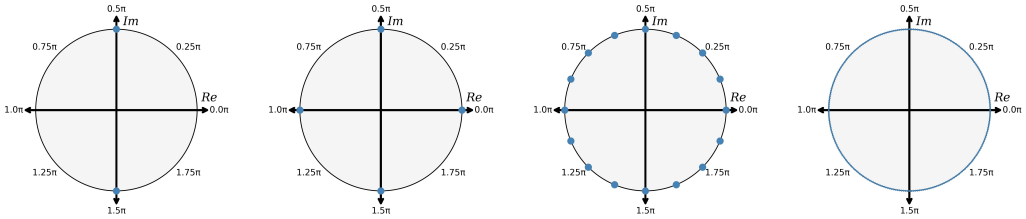


Fig. 6.1 Gerchberg-Saxton (GS) [19] algorithm flowchart

Then, as illustrated in Fig. 6.2, a quantization function (Q) can be defined by finding the closest point from one of the 2^d quantization levels, given the phase ($\angle A$) and the quantization bit depth d as input.



(a) bit depth = 1 (b) bit depth = 2 (c) bit depth = 4 (d) bit depth = 8

Fig. 6.2 Quantization of phase holograms

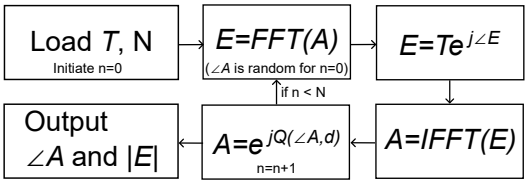


Fig. 6.3 Quantized Gerchberg-Saxton (GS) algorithm flowchart

To compute phase holograms quantized to certain bit depth (d), the GS algorithm is modified to include an additional quantization operation (Q) when applying the 'phase-only' constraint as shown in Fig. 6.3. Such method is better than applying the quantization at the end of the loop, as it includes the quantization constraint throughout the iterations, instead of introducing significant quantization noise in the end.

6.3.3 Measurement of Information

Shannon entropy

To quantify the information content, the classical one-dimensional (1D) Shannon entropy [76] with equation shown in Equation (6.3) is selected.

$$H(X) = -\sum_{x \in X} p(x) \log_2 p(x) \tag{6.3}$$

Although the Shannon entropy was designed for 1D data, it can also be implemented to two-dimensional (2D) data by ignoring the 2D spatial correlations and summing $p(x)\log_2 p(x)$ over the histogram of the 2D data. As only discrete data can have a meaningful Shannon

entropy, the entropy can only be calculated for quantized holograms and target images, which are usually quantized to less than 8 bit depth.

Delentropy

To account for 2D spatial correlation, the delentropy [79] is also used. Delentropy is an extension of the 1D Shannon entropy that it first computes the gradient (del) vector field image, whose entropy is then named as the delentropy, so that the spatial image structure and pixel co-occurrence can be captured [79].

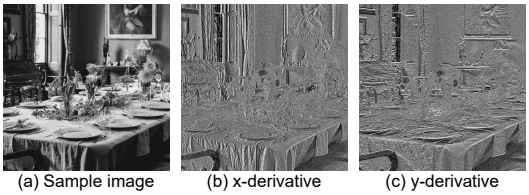


Fig. 6.4 Del operation on a sample image

As an example, the sample image in Fig. 6.4 (a) is the file '0500.png' under the 'DIV2K_train_HR' folder sourced from the DIV2K dataset [80]. The sample image is calculated to have a Shannon entropy of 7.502 bits/pixel. By taking the x-derivative (f_x) and y-derivative (f_y) as shown in Fig. 6.4 (b) and (c), and using the Papoulis Generalized Sampling (PGS) [81] theory, the delentropy is calculated using Equation (6.4)[79] to be 5.867 bits/pixel.

$$H_{PGS}(f_x, f_y) \leq \frac{H(f_x) + H(f_y)}{2} \tag{6.4}$$

6.4 Results

6.4.1 Targets at far field (Fraunhofer region)

The target images used in this paper are sourced from the DIV2K dataset [80], among which, all of the 800 images in the 'DIV2K_train_HR' folder are selected. The quantized GS algorithm in Fig. 6.3 was run on each of the 800 target images set at far field (using the Fraunhofer diffraction formula in Equation (6.2)), for hologram bit depth set to integers ranging from 1 to 8, with the total number of iterations (N) set to 100.

An example of quantized phase hologram generation for the sample image in Fig. 6.4 (a) is shown in Fig. 6.5, which demonstrates qualitatively how the reconstruction quality improves

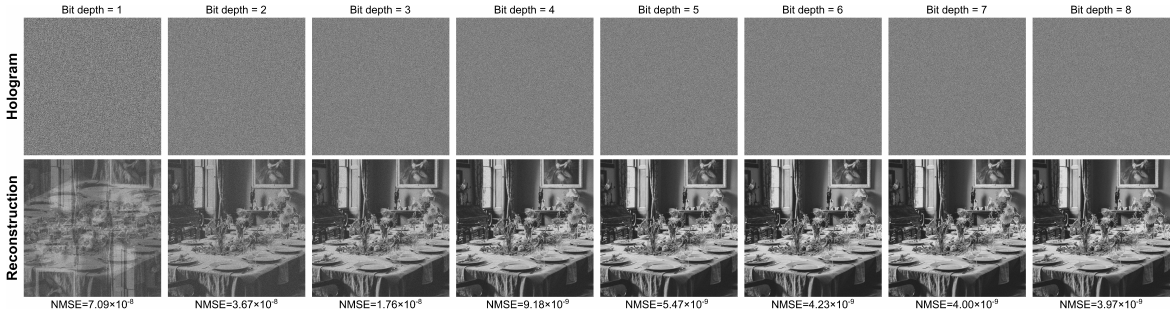


Fig. 6.5 Hologram generated at certain bit depths and their according reconstructions at far field

with the increase in the bit depth of the hologram, and also quantitatively, the normalized mean squared error (NMSE) between the reconstruction and target image has shown a decreasing trend as the bit depth of hologram increases. The rotational symmetry in the reconstruction of the hologram with bit depth 1 can be explained by the conjugate properties of Fourier Transforms, because the binary phase holograms have pixel values whose complex conjugates are the same as themselves.

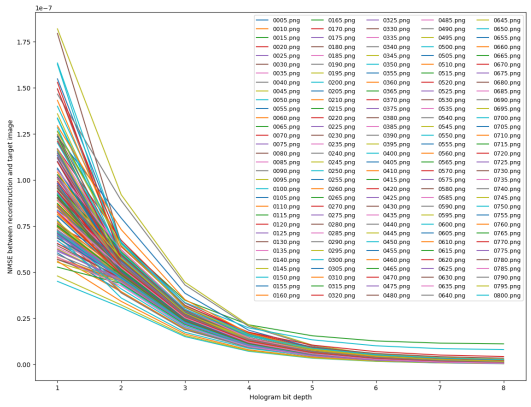
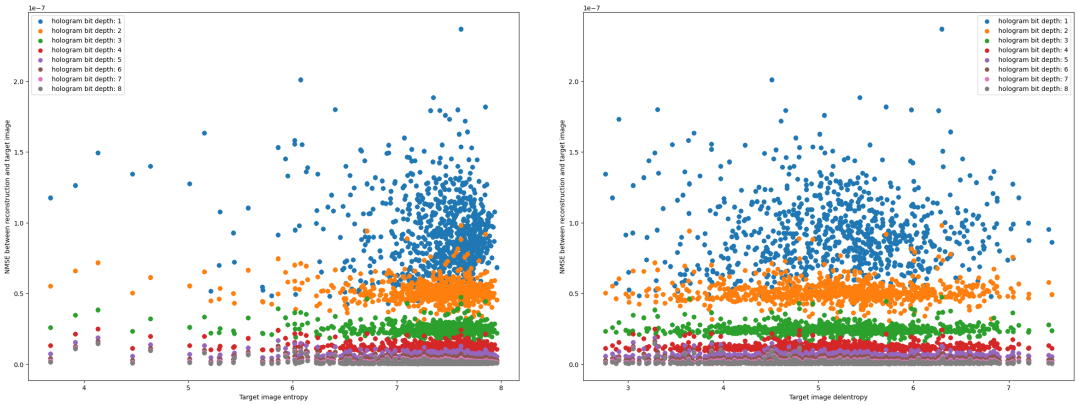


Fig. 6.6 NMSE v.s. Hologram bit depth for target images set at far field

Fig. 6.6 plots the NMSE between the reconstruction of the hologram and the target image against the hologram bit depth for every 5 images (as there are a total of 800 images, only every 5 images are shown in the plot to avoid overcrowding, where the full data can be accessed from the published research data [82]). It can be observed that, for each target image, the NMSE between the resulting reconstructions and their according target images decreases as the hologram bit depth increases. It infers that holograms with higher bit depth carries more sufficient information in order to better reconstruct the target images.



(a) (b)

Fig. 6.7 Scatter plot for target images set at far field. (a) NMSE v.s. Entropy. (b) NMSE v.s. Delentropy.

Then the entropy of each of the 800 target images are calculated using Equation (6.3), and a scatter plot of the NMSE between the reconstruction and target image against the target image entropy is plotted for all 800 target images as shown in Fig. 6.7 (a), with the difference in hologram bit depth distinguished by different colours. To avoid the effect of initial random phase on the final result, 5 different randomly generated initial phases are used for each run. However, the result [82] had rarely shown any difference between each run, leading to 5 dots overlapping at the same spot in the scatter plot. Unfortunately, no correlation has been found between the NMSE and the target image entropy, inferring that the Shannon entropy cannot be used to quantify the difficulty of CGH for a target image.

Lastly, the delentropy of each of the 800 target images are calculated following the method in Section 6.3.3, and a scatter plot of the NMSE between the reconstruction and target image against the target image delentropy is plotted in Fig. 6.7 (b) for the 800 target images. And again, no correlation has been found between the NMSE and the target image delentropy either, inferring that the 2D delentropy also fails to quantify the difficulty of CGH for a target image.

6.4.2 Targets at near field (Fresnel region)

The target images are now set to near field, where the Fresnel diffraction formula in Equation (6.1) applies; therefore the FFT and IFFT stages in Fig. 6.3 needs to be modified to include the phase term in Equation (6.1), and for experimental purpose, the distance (z) is set

at 10cm, the hologram's pixel pitch (sampling resolution of x and y) has a size of 13.62μm and the incident light's wavelength (λ) is 532nm.

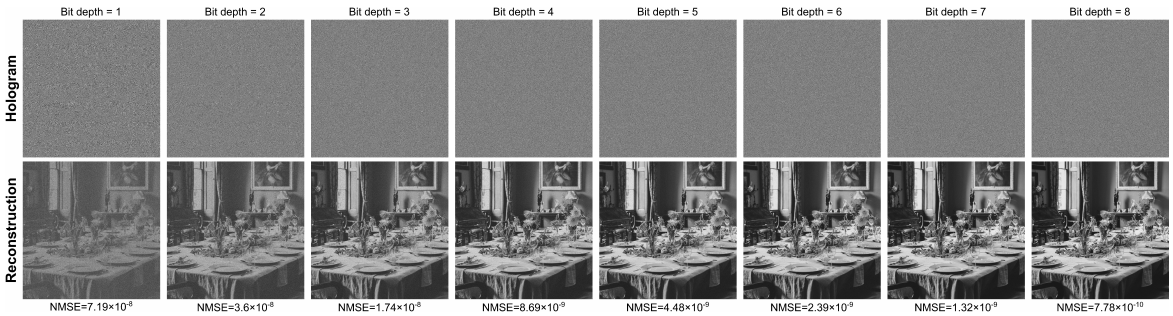


Fig. 6.8 Hologram generated at certain bit depths and their according reconstructions at near field

Fig. 6.8 shows both qualitatively and quantitatively how the reconstruction quality improves (i.e. NMSE decreases) with the increase in the bit depth of the hologram. Such trend is the same for target images places at near field as those placed at far field in Fig. 6.5. The rotational symmetry is gone for the binary phase hologram (bit depth = 1) due to the extra phase term in the Fresnel diffraction formula making the product of binary phase hologram and the phase term to be complex-valued whose complex conjugate does not equal to itself; however, the conjugate wouldn't disappear, but to appear at a different distance to where the target image is set at, leading to extra defocused noise onto the reconstruction plane. Nevertheless, the trend infers that holograms with higher bit depth produces better quality in the reconstruction plane.

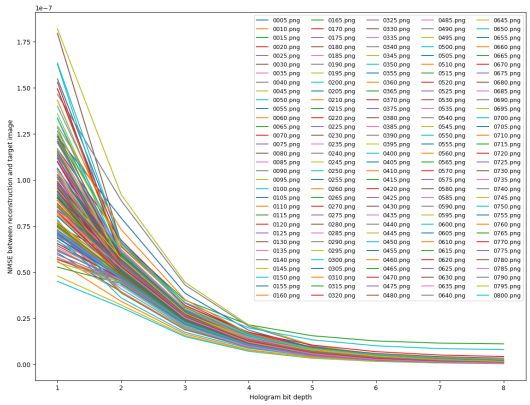
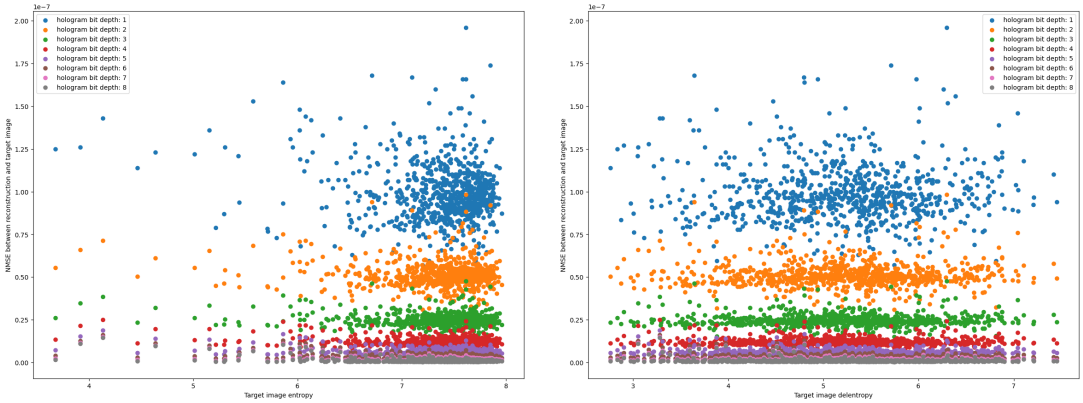


Fig. 6.9 NMSE v.s. Hologram bit depth for target images set at near field

Fig. 6.9 plots the trend of NMSE against increasing hologram bit depth, for every 5 target images to avoid overcrowding the plot, with all raw data accessible at the published research

data [82] (as the results for Fraunhofer has shown that the result from different initial random phases are the same, each run for Fresnel propagation only starts with one initial random phase as opposed to 5 different random phases for Fraunhofer propagation). In Fig. 6.9, the same trend as the one for far field in Fig. 6.6 can be observed, where NMSE decreases as hologram bit depth increases, for every single target image.

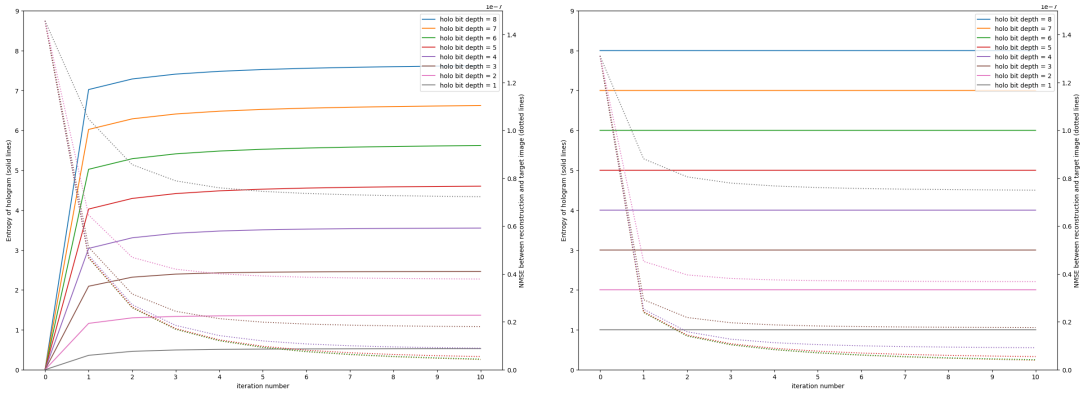


(a) (b)

Fig. 6.10 Scatter plot for target images set at near field. (a) NMSE v.s. Entropy. (b) NMSE v.s. Delentropy.

The scatter plot of NMSE against target image's entropy and delentropy are plotted, in Fig. 6.10 (a) and (b) respectively. No correlation has been observed between the NMSE and either the entropy or delentropy of target image. Such 'no correlation observed' result is the same as the results for the targets in far field in Section 6.4.1, confirming that neither entropy nor delentropy is suitable for quantifying how difficult a target image is for phase hologram computation.

The research moves on to investigate the entropy of holograms. In an example run for a quantized hologram generation in the near field (Fresnel propagation in Equation (6.1), both the hologram entropy and the NMSE between reconstruction and target image are recorded and plotted in Fig. 6.11 (a) and (b) for the first 20 iterations. As randomly generated holograms will by definition create holograms with high entropy, additional experiment has been carried out with initial phase set to zeros (i.e. $\angle A$ is set to zeros at $n = 0$ in Fig. 6.3). Therefore, two diagrams are plotted in Fig. 6.11, with Fig. 6.11 (a) having initial phase of zeros and Fig. 6.11 (b) having random initial phase. Both Fig. 6.11 (a) and (b) have the horizontal axis to be the iteration number n, and the vertical axis in the left is the entropy of hologram (corresponding to solid lines in the plot), while the vertical axis in the right is



(a) (b)

Fig. 6.11 Hologram entropy and NMSE v.s. iteration number for target images set at near field, with the initial phase ($\angle A$) being (a) zeros (b) random

the NMSE (corresponding to dotted lines in the plot). Colour coding is used to distinguish between the 8 different runs where hologram bit depth is set from 1 to 8.

The solid lines in Fig. 6.11 (a) show that the entropy of hologram keeps increasing towards a value lower than the bit depth, with their corresponding NMSE between reconstruction and target image (dotted lines) decreasing. Such trend can be explained qualitatively that, as the iteration goes on, the hologram is attempting to contain more information to sustain a better reconstruction, while the entropy cannot exceed or even reach the bit depth level. On the other hand, the random initial phase plotted in Fig. 6.11 (b) has a constant entropy approximately equal to the bit depth, which infers that the iterations are improving the reconstruction quality without reducing the information entropy of the hologram. In both cases, the entropy of the hologram does not decrease at any iteration, and the final NMSE does not have significant reduction when the hologram's bit depth exceeds 5.

The entropy of final hologram in (a) is lower than that in (b), although the final NMSE are similar. If the computer-generated holograms were to undergo a lossless compression, the hologram in (a) would be better compressed than the hologram in (b) as the entropy denotes the compression limit, assuming that the holograms are treated as 1D arrays when compressing. Therefore if hologram compression is of a concern, then starting with a low entropy initial hologram in the CGH process is recommended. In case if the reconstruction quality is not as high of a priority than making holograms to occupy less storage space, this paper also recommends quantized GS algorithm with 5 bit depth instead of 6-7 bit depth as the final reconstruction quality will not degrade significantly.

6.5 Conclusion

By carrying out the quantized GS algorithm on 800 sample target images placed at both far field (Fraunhofer diffraction) and near field (Fresnel diffraction), and computing the entropy and delentropy of the target images and holograms, this paper reaches the conclusion that, holograms with higher bit depth can sustain more information therefore producing better quality reconstructions. However, the quality of the reconstruction is not correlated to either the entropy or the delentropy of the target image, so neither entropy nor delentropy can quantify how difficult an image is for phase-only hologram generation. Additionally, the entropy of the hologram generated using quantized GS algorithm is not only bounded by the hologram bit depth, but also affected by the entropy of the initial phase. For applications where holograms file size is a high priority, if the quantized GS method is used, this paper advises a low entropy initial phase (e.g. all zeros) rather than a random initial phase and recommends to reduce the hologram bit depth limit, for lower entropy hologram generation.

References

[1] Jana Skirnewskaja, Yunuen Montelongo, Jinze Sha, and Timothy D. Wilkinson. Holographic lidar projections with brightness control. In *Imaging and Applied Optics Congress 2022 (3D, AOA, COSI, ISA, pcAOP)*, page 3F2A.6. Optica Publishing Group, 2022.

[2] Jinze Sha, Andrew Kadis, Fan Yang, and Timothy D. Wilkinson. Limited-memory bfgs optimisation of phase-only computer-generated hologram for fraunhofer diffraction. In *Digital Holography and 3-D Imaging 2022*, page W3A.3. Optica Publishing Group, 2022.

[3] Andrew Kadis, Benjamin Wetherfield, Jinze Sha, Fan Yang, Youchao Wang, and Timothy D. Wilkinson. Effect of bit-depth in stochastic gradient descent performance for phase-only computer-generated holography displays. *London Imaging Meeting*, 3:36–40, 7 2022.

[4] Jinze Sha, Andrew Kadis, Fan Yang, Youchao Wang, and Timothy D. Wilkinson. Multi-depth phase-only hologram optimization using the l-bfgs algorithm with sequential slicing. *J. Opt. Soc. Am. A*, 40(4):B25–B32, Apr 2023.

[5] Jinze Sha, Adam Goldney, Andrew Kadis, Jana Skirnewskaja, and Timothy D. Wilkinson. Digital pre-distorted one-step phase retrieval algorithm for real-time hologram generation for holographic displays. *Journal of Imaging Science and Technology*, 67(3):030405–1–030405–1, 2023.

[6] Jana Skirnewskaja, Yunuen Montelongo, Jinze Sha, Phil Wilkes, and Timothy D. Wilkinson. Accelerated augmented reality holographic 4k video projections based on lidar point clouds for automotive head-up displays. *Advanced Optical Materials*, 12(12):2301772, 2024.

[7] Roubing Meng, Jinze Sha, Zhongling Huang, and Timothy D. Wilkinson. Extending FOV of holographic display with alternating lasers. In Peter Schelkens and Tomasz Kozacki, editors, *Optics, Photonics, and Digital Technologies for Imaging Applications VIII*, volume 12998, page 129981J. International Society for Optics and Photonics, SPIE, 2024.

[8] Jinze Sha, Andrew Kadis, Benjamin Wetherfield, Roubing Meng, Zhongling Huang, Dilawer Singh, Antoni Wojcik, and Timothy D. Wilkinson. Information capacity of phase-only computer-generated holograms for holographic displays. In Peter Schelkens and Tomasz Kozacki, editors, *Optics, Photonics, and Digital Technologies for Imaging*

Applications VIII, volume 12998, page 129980J. International Society for Optics and Photonics, SPIE, 2024.

[9] Jinze Sha, Antoni Wojcik, Benjamin Wetherfield, Jianghan Yu, and Timothy D. Wilkinson. Multi frame holograms batched optimization for binary phase spatial light modulators. *Scientific Reports*, 14(1):19380, Aug 2024.

[10] Ivan Y. Lo. A photo of the holographic portrait of dennis gabor, 2018.

[11] Johannes Kalliauer. An illustration of the 'double-slit experiment' in physics, 2017.

[12] Jeff Hecht. *Solid-State and Fiber Lasers*, chapter 8, pages 223–263. John Wiley & Sons, Ltd, 2008.

[13] Arne Nordmann. Wave diffraction in the manner of huygens and fresnel, 2007.

[14] Timothy D. Wilkinson. Lecture notes of 4b11 photonics systems course, 2019. University of Cambridge.

[15] A. J. Cable. Real-time high-quality two and three-dimensional holographic video projection using the one-step phase retrieval (ospr) approach, 2006. PhD thesis, Department of Engineering, University of Cambridge, United Kingdom.

[16] Allan Weber. Sipi image database - misc, 2022. [retrieved 17 Nov 2022].

[17] Chun Chen, Byounghyo Lee, Nan-Nan Li, Minseok Chae, Di Wang, Qiong-Hua Wang, and Byoungho Lee. Multi-depth hologram generation using stochastic gradient descent algorithm with complex loss function. *Optics Express*, 29:15089, 5 2021.

[18] J. Freeman. Visor projected helmet mounted display for fast jet aviators using a fourier video projector, 2009. PhD thesis, Department of Engineering, University of Cambridge, United Kingdom.

[19] R W Gerchberg. A practical algorithm for the determination of phase from image and diffraction plane pictures. *Optik*, 35:237–246, 1972.

[20] D. Gabor. A new microscopic principle. *Nature*, 161:777–778, 1948.

[21] Eugene Hecht. *Optics*. Pearson Education Limited, 5 edition, 2017.

[22] Michael A. Seldowitz, Jan P. Allebach, and Donald W. Sweeney. Synthesis of digital holograms by direct binary search. *Applied Optics*, 26, 1987.

[23] Han Jin Yang, Jeong Sik Cho, and Yong Hyub Won. Reduction of reconstruction errors in kinoform cghs by modified simulated annealing algorithm. *Journal of the Optical Society of Korea*, 13, 2009.

[24] Isaac Newton. *Opticks*. Dover Press, 1704.

[25] C. Huygens. *Traite de la lumiere. Où sont expliquées les causes de ce qui luy arrive dans la reflexion, & dans la refraction. Et particulierment dans l'etrange refraction du cristal d'Islande, par C.H.D.Z. Avec un Discours de la cause de la pesanteur*. chez Pierre Vander Aa marchand libraire, 1690.

[26] Thomas Young. Ii. the bakerian lecture. on the theory of light and colours. *Philosophical Transactions of the Royal Society of London*, 92:12–48, 1802.

[27] Augustin Jean Fresnel. *Memoir on the Diffraction of Light*. 1826.

[28] John Daintith. *A Dictionary of Physics*. Oxford University Press, 2009.

[29] Timothy D. Wilkinson. *Electrical Data Book*. Cambridge University Engineering Department, 2017.

[30] James Clerk Maxwell. Viii. a dynamical theory of the electromagnetic field. *Philosophical Transactions of the Royal Society of London*, 155:459–512, 1865.

[31] A. Einstein. On a heuristic point of view about the creation and conversion of light. *Annalen der Physik*, 322(6):132–148, 1905.

[32] Gould R. Gordon. The laser, light amplification by stimulated emission of radiation. page 128. Ann Arbor, 6 1959.

[33] Edwin Cartlidge. Theodore maiman 1927–2007. *Physics World*, 20, 2007.

[34] John B Develis, Merrimack College, North Andover, and George O Reynolds. Three dimensional hologram reconstruction and image speckle, 1966.

[35] A. J. Cable, E. Buckley, P. Mash, N. A. Lawrence, T. D. Wilkinson, and W. A. Crossland. 53.1: Real-time binary hologram generation for high-quality video projection applications. *SID Symposium Digest of Technical Papers*, 35:1431, 2004.

[36] Tim Stangner, Hanqing Zhang, Tobias Dahlberg, Krister Wiklund, and Magnus Andersson. Step-by-step guide to reduce spatial coherence of laser light using a rotating ground glass diffuser. *Applied Optics*, 56:5427, 7 2017.

[37] Linxiao Deng, Tianhao Dong, Yuwei Fang, Yuhua Yang, Chun Gu, Hai Ming, and Lixin Xu. Speckle reduction in laser projection based on a rotating ball lens. *Optics and Laser Technology*, 135, 3 2021.

[38] Philip J W Hands, Calum M Brown, Daisy K E Dickinson, Stephen M Morris, and Jia-De Lin. Liquid-crystal lasers: Recent advances and future opportunities, 2022.

[39] Joseph W. Goodman. *Introduction to Fourier Optics, Fourth Edition*. W. H. Freeman, 2017.

[40] M. Schadt and W. Helfrich. Voltage-dependent optical activity of a twisted nematic liquid crystal. *Applied Physics Letters*, 18, 1971.

[41] Dennis R. Pape and Larry J. Hornbeck. Characteristics of the deformable mirror device for optical information processing. *Optical Engineering*, 22, 1983.

[42] Kristina M. Johnson, Douglas J. McKnight, and Ian Underwood. Smart spatial light modulators using liquid crystals on silicon. *IEEE Journal of Quantum Electronics*, 29, 1993.

[43] Yongmin Lee, James Gourlay, William J. Hossack, Ian Underwood, and Anthony J. Walton. Multi-phase modulation for nematic liquid crystal on silicon backplane spatial light modulators using pulse-width modulation driving scheme. *Optics Communications*, 236, 2004.

[44] S. E. Broomfield, M. A.A. Neil, E. G.S. Paige, and G. G. Yang. Programmable binary phase-only optical device based on ferroelectric liquid crystal slm. *Electronics Letters*, 28, 1992.

[45] Zhengzhong Huang and Liangcai Cao. Quantitative phase imaging based on holography: trends and new perspectives. *Light: Science & Applications*, 13(1):145, Jun 2024.

[46] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[47] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220, 1983.

[48] Guo zhen Yang, Bi zhen Dong, Ben yuan Gu, Jie yao Zhuang, and Okan K. Ersoy. Gerchberg–saxton and yang–gu algorithms for phase retrieval in a nonunitary transform system: a comparison. *Appl. Opt.*, 33(2):209–218, Jan 1994.

[49] Haichao Wang, Weirui Yue, Qiang Song, Jingdan Liu, and Guohai Situ. A hybrid gerchberg–saxton-like algorithm for doe and cgh calculation. *Optics and Lasers in Engineering*, 89:109–115, 2017. 3DIM-DS 2015: Optical Image Processing in the context of 3D Imaging, Metrology, and Data Security.

[50] Pengcheng Zhou, Yan Li, Shuxin Liu, and Yikai Su. Dynamic compensatory gerchberg-saxton algorithm for multiple-plane reconstruction in holographic displays. *Optics Express*, 27:8958, 3 2019.

[51] E. Buckley. Computer-generated holograms for real-time image display and sensor applications, 2006. PhD thesis, Department of Engineering, University of Cambridge, United Kingdom.

[52] Edward Buckley. 70.2: Invited paper: Holographic laser projection technology. *SID Symposium Digest of Technical Papers*, 39(1):1074–1079, 2008.

[53] Jingzhao Zhang, Nicolas Pégard, Jingshan Zhong, Hillel Adesnik, and Laura Waller. 3d computer-generated holography by non-convex optimization. *Optica*, 4:1306, 10 2017.

[54] Shujian Liu and Yasuhiro Takaki. Optimization of phase-only computer-generated holograms based on the gradient descent method. *Applied Sciences (Switzerland)*, 10, 2020.

[55] Suyeon Choi, Jonghyun Kim, Yifan Peng, and Gordon Wetzstein. Optimizing image quality for holographic near-eye displays with michelson holography. *Optica*, 8:143, 2 2021.

[56] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, 2006.

[57] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12:2121–2159, 2011.

[58] Tijmen Tieleman, Geoffrey Hinton, et al. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4:26–31, 2012.

[59] Diederik P. Kingma and Jimmy Lei Ba. Adam: A method for stochastic optimization. 2015.

[60] Dong C. Liu and Jorge Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical Programming*, 45, 1989.

[61] R C Gonzalez and R E Woods. *Digital Image Processing*. Prentice Hall, 2002.

[62] Xuetun Zhao. Suzhou center mall, 2017. Suzhou, Jiangsu, China.

[63] Lemarechal Claude. Cauchy and the gradient method. *Doc Math Extra*, pages 251–254, 2012.

[64] Roger Fletcher. *Practical methods of optimization*. Wiley, 1987.

[65] Claude Sammut and Geoffrey I. Webb, editors. *Mean Squared Error*, pages 653–653. Springer US, Boston, MA, 2010.

[66] G. Cybenko, D.P. O'Leary, and J. Rissanen. *The Mathematics of Information Coding, Extraction and Distribution*. The IMA Volumes in Mathematics and its Applications. Springer New York, 1998.

[67] S. Kullback and R. A. Leibler. On Information and Sufficiency. *The Annals of Mathematical Statistics*, 22(1):79 – 86, 1951.

[68] Xunying Liu, Shansong Liu, Jinze Sha, Jianwei Yu, Zhiyuan Xu, Xie Chen, and Helen Meng. Limited-memory bfgs optimization of recurrent neural network language models for speech recognition. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2018-April:6114–6118, 9 2018.

[69] Nik Clark. Utah teapot (solid), 2015.

[70] Jun Amako, Hirotsuna Miura, and Tomio Sonehara. Speckle-noise reduction on kinoform reconstruction using a phase-only spatial light modulator. *Appl. Opt.*, 34(17):3165–3171, Jun 1995.

[71] Christoph Bay, Nils Hübner, Jon Freeman, and Tim Wilkinson. Maskless photolithography via holographic optical projection. *Optics Letters*, 35(13):2230–2232, Jul 2010.

[72] Andrzej Kaczorowski, George S. D. Gordon, and Timothy D. Wilkinson. Adaptive, spatially-varying aberration correction for real-time holographic projectors. *Opt. Express*, 24(14):15742–15756, Jul 2016.

[73] Nicolas Bacaër. *Verhulst and the logistic equation (1838)*, pages 35–39. Springer London, London, 2011.

[74] Jinze Sha, Antoni Wojcik, Benjamin Wetherfield, Jianghan Yu, and Timothy D. Wilkinson. Research data supporting "multi-frame holograms batched optimization". Apollo - University of Cambridge Repository, 2024.

[75] P. W. M. Tsang, T.-C. Poon, and Y. M. Wu. Review of fast methods for point-based computer-generated holography. *Photon. Res.*, 6(9):837–846, Sep 2018.

[76] C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27, 1948.

[77] Joel S Kollin. Design and information considerations for holographic television, 1988.

[78] W T Cochran, J W Cooley, D L Favin, H D Helms, R A Kaenel, W W Lang, G C Maling, D E Nelson, C M Rader, and P D Welch. What is the fast fourier transform? *Proceedings of the IEEE*, 55:1664–1674, 1967.

[79] Kieran G. Larkin. Reflections on shannon information: In search of a natural information-entropy for images, 2016.

[80] Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. July 2017.

[81] A. Papoulis. Generalized sampling expansion. *IEEE Transactions on Circuits and Systems*, 24(11):652–654, 1977.

[82] Jinze Sha, Andrew Kadis, Benjamin Wetherfield, Roubing Meng, Zhongling Huang, Dilawer Singh, Antoni Wojcik, and Timothy D. Wilkinson. Research data supporting 'information capacity of phase-only computer-generated holograms for holographic displays', 2024.