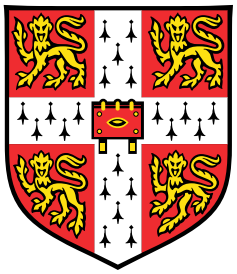
In search for the optimal phase hologram



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

August 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
August 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

Table of contents

List of figures

List of tables

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 [9]

Holography, taking its name from the Greek word $o\lambda o\sigma$ (holos), meaning *whole*, was first introduced in 1948 by Dennis Gabor [18], originally named as *wavefront reconstruction* [19]. 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 [14, 20–22]. 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.

This report starts from the literature review in Chapter 2, then records my current progress of research in Chapter 3 and **??**, and lastly, concludes with the future plan for the upcoming

years of my PhD study in **??**. Because I did my masters project on the same topic in the 2019-2020 academic year for my M.Eng. degree, the literature review in Chapter 2 contains some basic theories in my masters report, and the progress in Chapter 3 continues from where I left off from the unexpected cancellation of Easter term 2020.

Chapter 2

Literature Review

Note: Section 2.1 and Section 2.2 partly contains my masters project's report submitted in 2020 for M.Eng. degree.

2.1 Light

2.1.1 Light Source

For any type of projection, a light source is needed, whether it's artificial or natural. The mechanism of holographic projection is to control the propagation of light in a way that, after diffraction and interference, reconstructs a wavefront that matches the target field. We usually prefer to start from a coherent and monochromatic light source rather than a random source which will be a lot more difficult or even impossible to analyse and predict the interference pattern. Laser, which stands for ***l****ight* ***a****mplification by the* ***s****timulated* ***e****mission of* ***r****adiation*, was first invented by Theodore Maiman in 1959 [23, 24]. 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 [25].

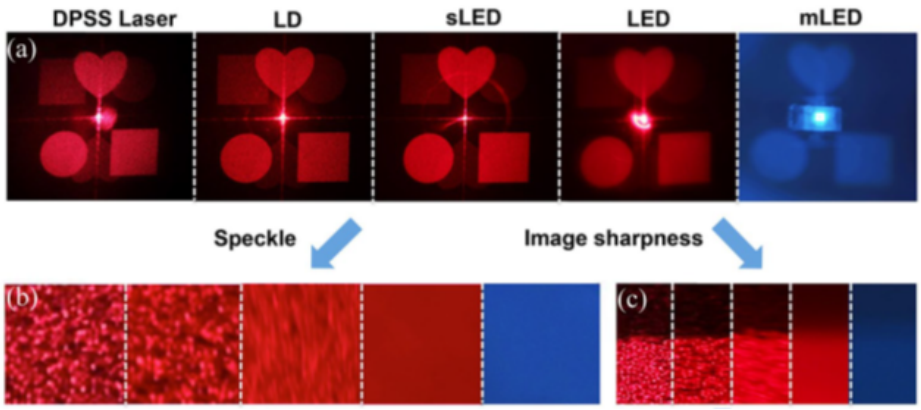


Fig. 2.1 Comparison of commercially available light sources for use of holographic projections. (a) Reconstructed images of the same target image for different light sources. (b) Enlarged images showing the details of speckle in the same area of interest for these light sources. (c) Enlarged images showing image edge in the same area of interest for these light sources [10]

Previous work had compared several commercially available light sources (diode-pumped solid-state (DPSS) laser, laser diode (LD), light emitting diode (LED), super luminescent light emitting diode (sLED) and micro light emitting diode (mLED)) for use of holographic projections [10]. It can be seen from Fig. 2.1 that, when laser (both DPSS laser and laser diode) are used as light source, the reconstruction has sharp edges, but suffers from speckles. When the three types of LEDs are used, the image speckle is greatly reduced, but the edges are a lot blurred due to its incoherence property.

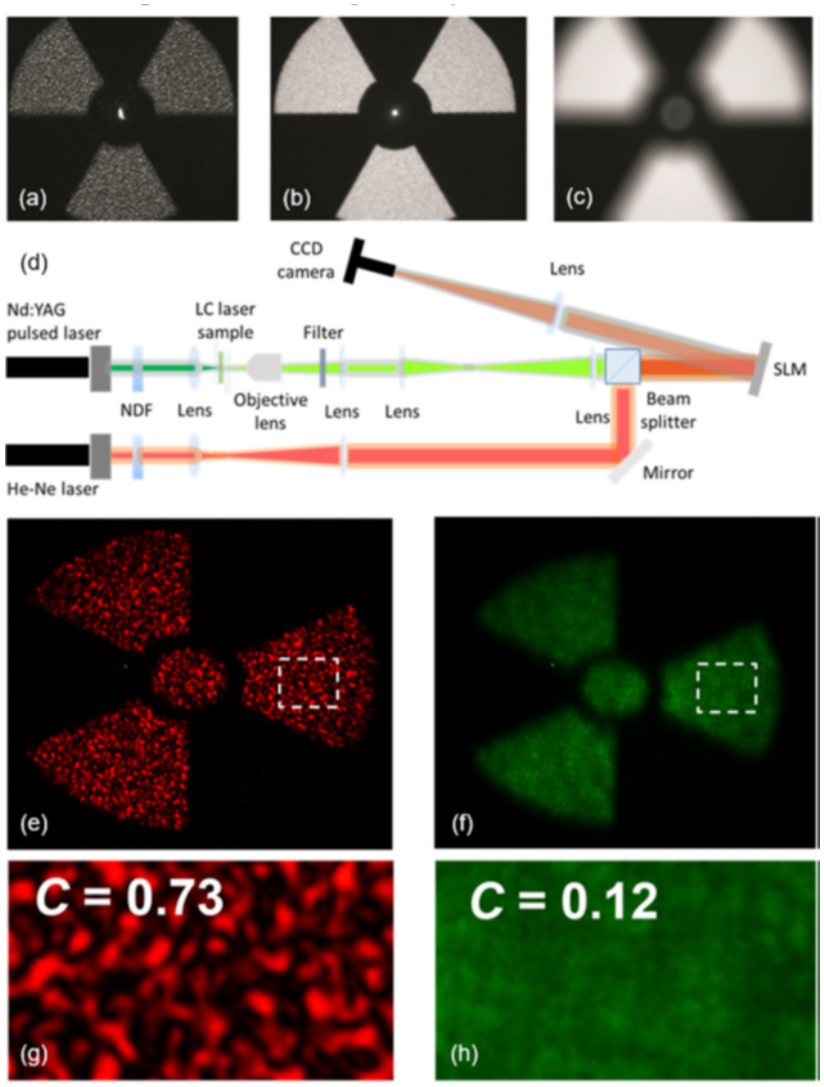


Fig. 2.2 Liquid crystal laser. (a)-(c) Replay field images generated from a multi-level phase hologram illuminated with (a) DPSS laser, (b) LC laser, (c) LED. (d) Experimental setup for computer generated holographic projection, and corresponding images when illuminated by (e) DPSS laser, and (f) LC laser. (g) & (h) show the speckle patterns and speckle contrast values for the highlighted regions in (e) & (f) respectively. [11]

Despite lots of efforts trying to reduce the speckle [26, 27], the development of Liquid Crystal (LC) laser has shone a new light on holography as it not only has an advantage of being tunable in wavelength, but also has a unique feature of low spatial coherence and high temporal coherence [11]. As shown in Fig. 2.2, the holographic projection using the LC laser has much less speckle than traditional DPSS laser, while there is a just a slight blurring of the edge. It is at a good balance between DPSS laser and LED.

2.1.2 Light Propagation

Maxwell Equations

In 1864, James Clerk Maxwell proposed a set of four equations describing the space and time dependence of the electromagnetic field, which are:

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

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

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

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

where **D** is the electric flux density, **E** is the electric field intensity, **B** is the magnetic flux density, **H** is the magnetic field intensity, ρ is the volume charge density, and **J** is the current density [28].
And the relation between **D** and **E** and between **B** and **H** for linear materials (such as free space) are:

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

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

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

Wave Equation

Light is an electromagnetic wave, so for propagation of light in any material in absence of free charge, the Maxwell equations in Eq. (2.1) - Eq. (2.4) can be simplified as:

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

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

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

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

Taking the curl of Eq. (2.7), and using the vector identity of $\nabla \times (\nabla \times \mathbf{u}) = \nabla(\nabla \cdot \mathbf{u}) - \nabla^2\mathbf{u}$:

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

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

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

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

Hence, we have a generic form of wave equation:

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

A valid solution to Eq. (2.14) is:

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

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 simply a phase shift. This suggests that, if we have a coherent light source and a phase modulator, we can essentially manipulate the phase of light to produce an interference pattern reconstructing the target field we desire, and such method is called holographic projection.

2.1.3 Diffraction

Diffraction Through a Two-Dimensional (2D) Aperture

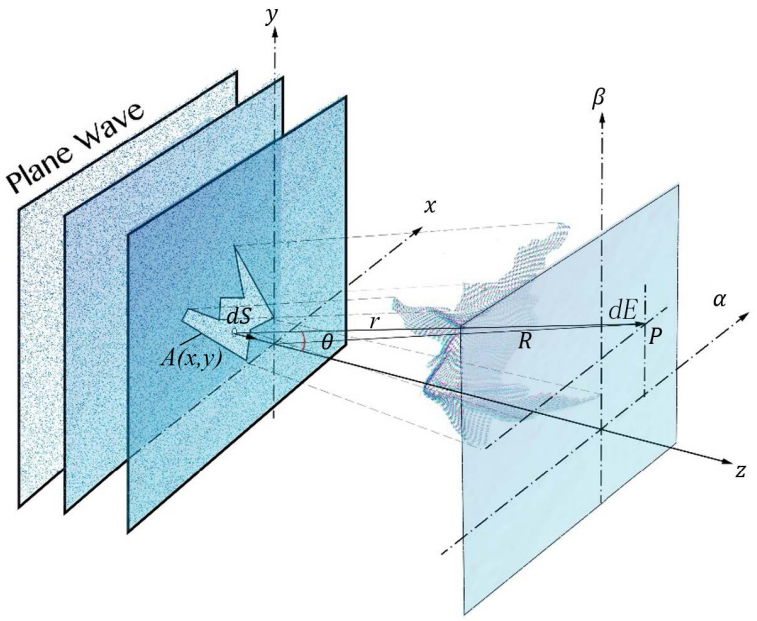


Fig. 2.3 Diffraction geometry

To model how light diffracts through a 2D aperture, we first set up a coordinate system as shown in Fig. 2.3, where the aperture is denoted by $A(x,y)$ and the diffracted field is denoted by $E(\alpha,\beta,z)$.

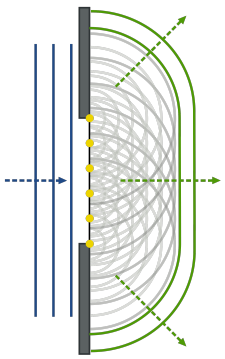


Fig. 2.4 Huygens-Fresnel wavelet principle [12]

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 $r \gg \lambda$ [30]:

$$E(\alpha,\beta,z) = \frac{1}{j\lambda}\iint A(x,y)\frac{e^{jkr}}{r}cos(\theta)dxdy \tag{2.16}$$

And by trigonometry we can have the following identities:

$$cos(\theta) = \frac{z}{r} \tag{2.17}$$

$$R^2 = \alpha^2 + \beta^2 + z^2 \tag{2.18}$$

$$r^2 = (\alpha - x)^2 + (\beta - y)^2 + z^2 \tag{2.19}$$

Then Eq. (2.16) 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) can only be solved analytically for few specific aperture functions $A(x,y)$, so we have to make some approximations to solve for arbitrary $A(x,y)$, the common methods are *Fresnel* and *Fraunhofer* approximations for regions depicted in Fig. 2.5 below.

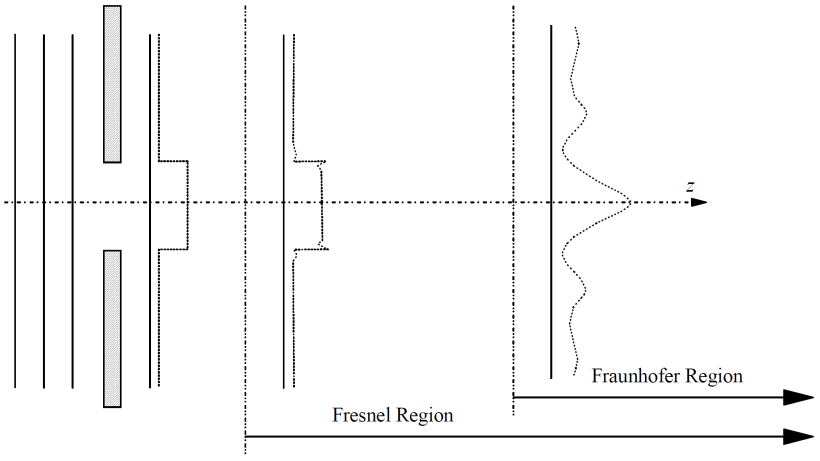


Fig. 2.5 Fresnel and Fraunhofer region [13]

Fresnel Approximation

Fesnel approximation replaces expressions for spherical waves by quadratic-phase exponentials, using the binomial expansion of the square root to approximate r in Eq. (2.20) [30]:

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

Retaining only the first two terms of the expansion in Eq. (2.22), and substituting in Eq. (2.19) 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, but for the r appearing in the exponent in the numerator of Eq. (2.20), errors are much more critical [30]. 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:

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

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

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

where $\mathscr{F}$ is the Fourier Transform.

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 [13]. 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. (2.29) 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 Computer-Generated Hologram (CGH)

CGH is the method of digitally generating the interference patterns and displaying it via a spatial light modulator (SLM), therefore reconstructing the 3D target. This section discusses the available SLM's, and algorithms to compute the holograms.

2.2.1 Modulation Schemes

Currently, available display devices fall into four modulation categories, as illustrated in Fig. 2.6 [14].

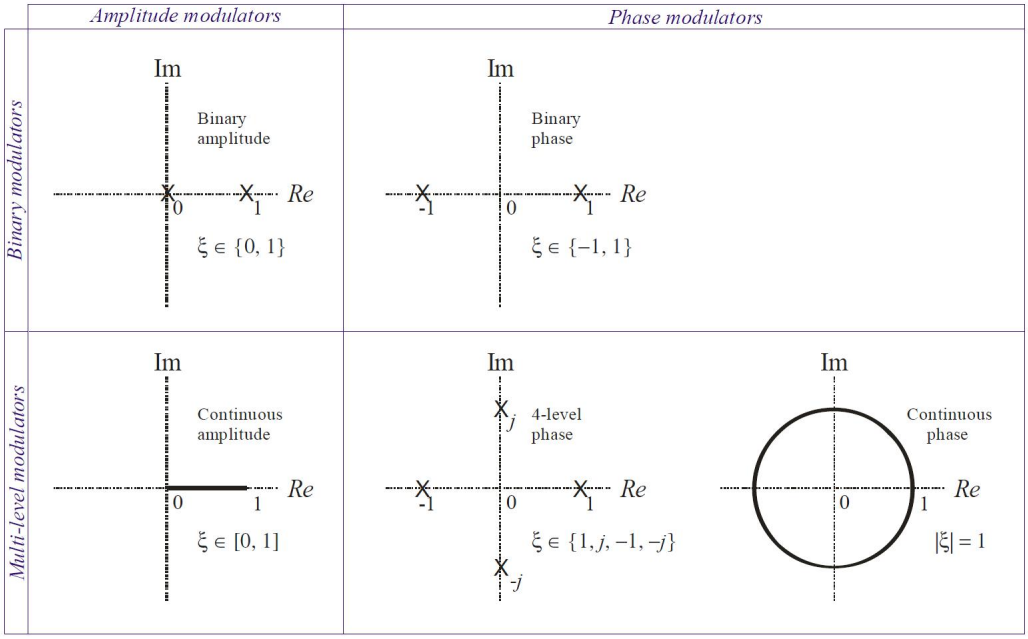


Fig. 2.6 Modulation loci in the complex plane [14]

The four modulation schemes are [14]:

- **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 [31], 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 [32], ferroelectric liquid crystal display [33], 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 [34])
- **Binary Phase** modulators can switch each pixel for a phase shift of either 0 or π radians. (e.g. Ferroelectric liquid crystal displays [35])

Among the four modulation schemes, phase modulations are of higher interests, because amplitude modulations, either multi-level or binary, blocks light at the spatial light modulator (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 forcing the hologram to have zero average, because the average of phase hologram can lie on 0 if well-designed.

The most common phase modulators are still only providing binary phase modulation, as the binary phase modulation is purely real (as it's only switching between 0° and 180°), 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 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. The simplest workaround for this issue is to use only half of the reconstruction field, and the computation of CGH for binary SLM will naturally need binary quantisation. Although few multi-level phase modulators are available, their bit depth is still limited (e.g. 4-bit 8-bit), so quantisation is still needed for the discrete levels, the effect of which is analysed in [3].

In summary, before the invention of a complex modulator, we need algorithms to generate phase only hologram, such process is called phase retrieval. There are currently many algorithms for this purpose, which are discussed in Section 2.2.2 - Section 2.2.6.

2.2.2 Naive Algorithm

The naive algorithm to get a phase hologram is by directly using the phase of the reverse propagation from the target field to the hologram plane (e.g. for Fraunhofer propagation, the hologram is simply the inverse Fourier transform $\mathscr{F}^{-1}$ of the target image), while discarding the amplitude component. The pseudocode of naive algorithm is shown in Algorithm 1 below:

Algorithm 1 Naive algorithm

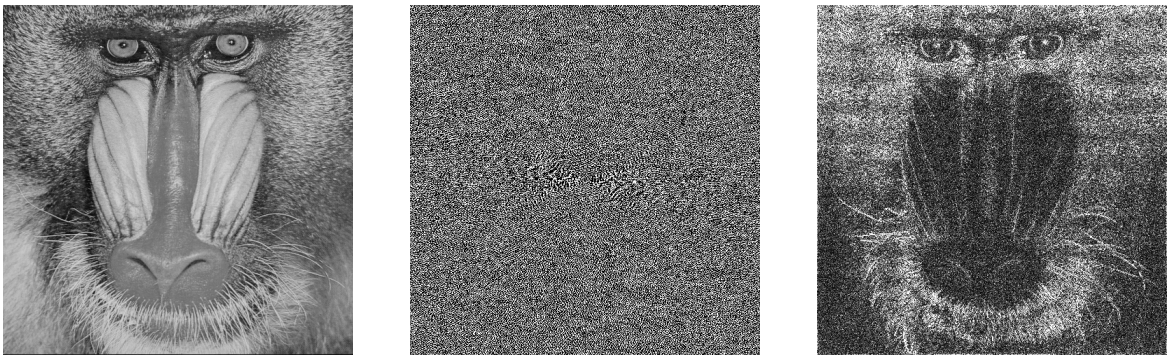
Input: Target field T, Propagation function $\mathscr{P}$ (e.g. Fresnel or Fraunhofer propagation)

Output: Phase hologram H and its reconstruction R

$H \leftarrow \angle\{\mathscr{P}^{-1}[T]\}$

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

where $j = \sqrt{-1}$. Naive algorithm (as described in Algorithm 1) was then implemented in MATLAB and the output results are shown in Fig. 2.7 below:



(a) Target field (T) (b) Hologram (H) (c) Reconstruction (R)

Fig. 2.7 Naive algorithm output

As shown in the simulation result, the reconstruction (Fig. 2.7c) is very far from the desired target image. It has shown that, discarding amplitude introduces a significant loss of information.

Moreover, in order to display the hologram on a binary phase modulator or multi level phase modulator with limited bit-depth, the phase needs to be quantised again, introducing additional quantisation error.

2.2.3 Direct Binary Search (DBS) Algorithm

Direct Binary Search (DBS) algorithm [20] is specifically designed for binary phase modulators, it generates the hologram by randomly flipping each pixel in the SLM between binary states, one by one for many times in order to minimise the difference between its reconstruction R and the target image T. The detailed algorithm is described in Algorithm 2 below:

Algorithm 2 Direct Binary Search (DBS) algorithm

Input: Target field T, Propagation function $\mathscr{P}$, Loss function $\mathscr{L}$, Number of iterations N

Output: Phase hologram H and its reconstruction R

```
// Start with a random hologram with a size matching T
H ← Rand(Size(T))
R ← |𝒫[e^{jH}]|
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}]|
    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
```

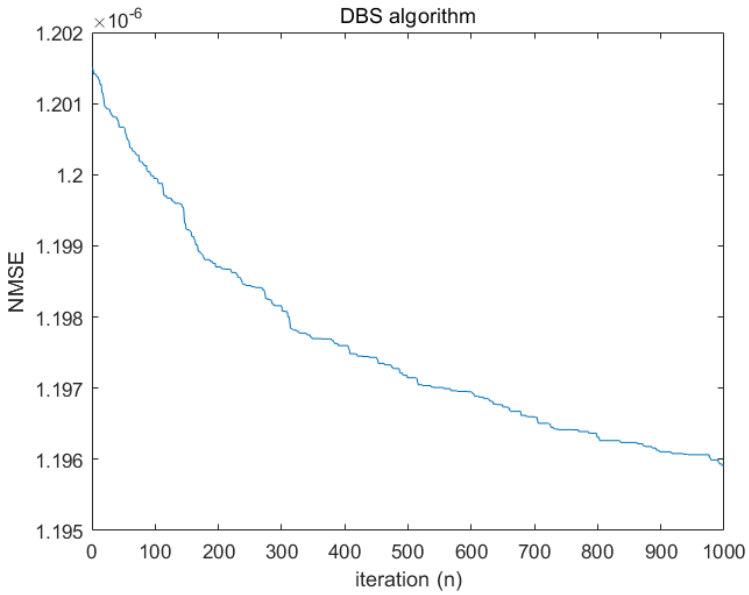


Fig. 2.8 DBS algorithm NMSE plot

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. 2.8, even 1000 iterations has not reached good convergence) and each iteration requires a Fourier transform which is computationally heavy. 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.

2.2.4 Simulated Annealing (SA) Algorithm

Simulated Annealing (SA) is an improvement on the DBS algorithm, [36]. 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 [21]. To implement this idea, we then need a function ($\mathscr{Q}$) 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. The pseudocode for this algorithm is listed in Algorithm 3.

Algorithm 3 Simulated Annealing (SA) algorithm

Input: Target field T, Propagation function $\mathscr{P}$, Loss function $\mathscr{L}$, Number of iterations N, Probability function $\mathscr{Q}$ (e.g. Boltzmann Distribution), Probability threshold p_t

Output: Phase hologram H and its reconstruction R

```
// Start with a random hologram with a size matching T
H ← Rand(Size(T))
R ← |𝒫[e^{jH}]|
L ← ℒ[R_0, 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}]|
    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
    else
        // Calculate the probability of the hologram
        p_n ← 𝒬[H_n]
        if p_n > p_t then
            // Accept the new hologram if the probability exceeds the threshold
            H ← H_n
            R ← R_n
            L ← L_n
        end if
    end if
end for
```

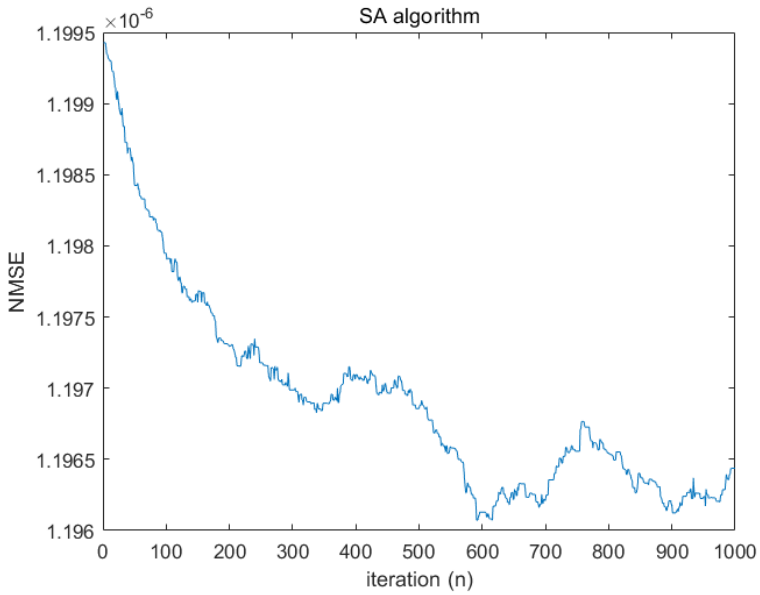


Fig. 2.9 SA algorithm NMSE plot

An implementation of SA algorithm with $p_t = 0.8$ was carried out, and the result is as shown in Fig. 2.9. It can be seen that, instead of monotonic decrease observed in Fig. 2.8 for DBS algorithm, the SA algorithm has occasional rises in NMSE, where the probability p_n exceeds the threshold p_t.

2.2.5 Gerchberg-Saxton (GS) algorithm

Gerchberg-Saxton (GS) algorithm functions that it iteratively determines the phase profile of the hologram required to reconstruct a target image; it loops between the hologram and the reconstruction plane, and applying constraints to each plane accordingly during each iteration [22]. GS algorithm is very easy to implement, its pseudocode is shown in Algorithm 4.

Algorithm 4 Gerchberg-Saxton (GS) Algorithm

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

Output: Phase hologram H and its reconstruction R

// Initiate E with amplitude T and initial phase Φ

$E \leftarrow T * e^{j\Phi}$

for n = 1 to N **do**

// Compute the hologram plane

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

// Apply the phase-only constraint at the hologram plane

$A \leftarrow e^{j\angle A}$

// Compute the propagation for the new hologram

$E \leftarrow \mathscr{P}[A]$

// Apply the target field amplitude constraint at the hologram plane

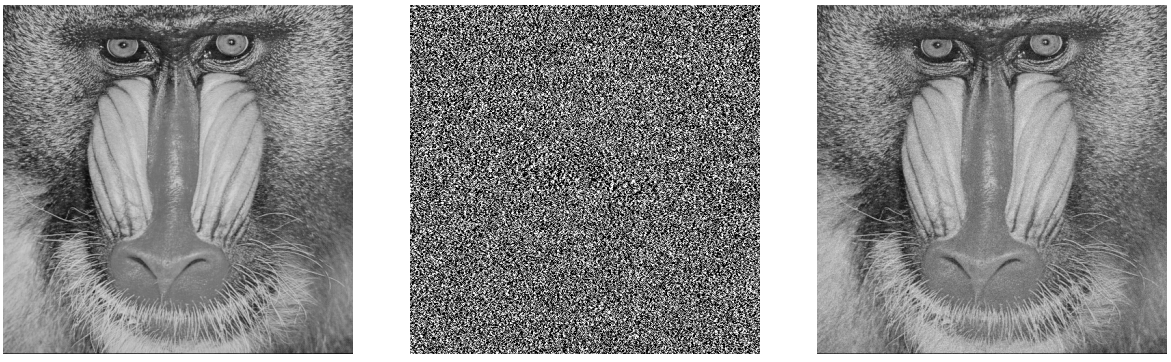
$E \leftarrow T * e^{j\angle E}$

end for

$H \leftarrow \angle A$

$R \leftarrow |\mathscr{P}[A]|$

The GS algorithm (described in Algorithm 4) was implemented in MATLAB and the output results are shown in Fig. 2.10 below:



(a) Target field (T) (b) Hologram (H) (c) Reconstruction (R)

Fig. 2.10 GS algorithm output for 30 iterations ($N = 30$)

It can be seen from Fig. 2.10c that, the reconstruction of the hologram after 30 iterations of GS algorithm reached a very good result. Then the quantitative analysis was carried out by measuring the normalised mean squared error (NMSE) as the iteration number (n) increments, the result is plotted in Fig. 2.11.

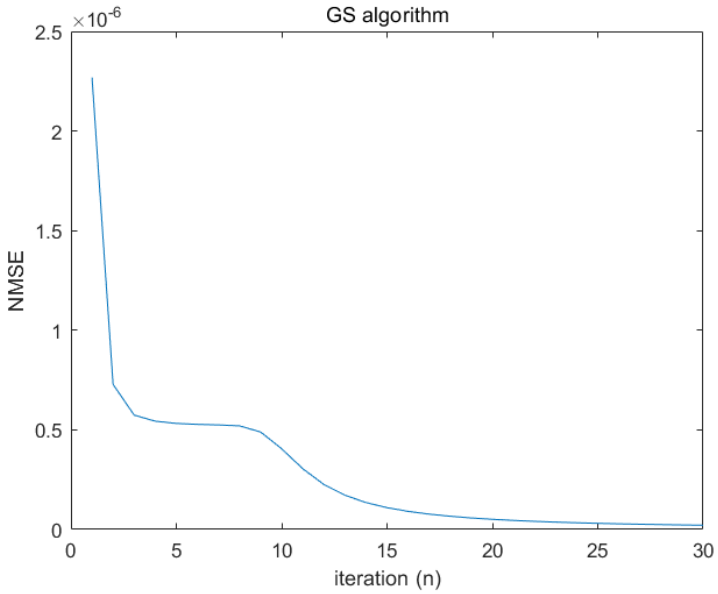


Fig. 2.11 GS algorithm NMSE plot

It can be seen from Fig. 2.11 that GS algorithm converges quickly, providing very good result in tens of iterations. The major disadvantage of the GS algorithm is that it is more computationally expensive at every iteration, as it needs to compute both a forward and an inverse propagation function, causing the need for two Fourier transforms at each iteration. Nevertheless, the algorithm has very good performance over all.

2.2.6 One-Step Phase Retrieval (OSPR) Algorithm

OSPR algorithm was first demonstrated by Buckley [37]. OSPR 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 [14]. The random noises are averaged by the eye, while the target image stays, hence the average noise is reduced. The perceived noise is lessened by the temporal average detected by the eye, rather than computational optimisation of the hologram [14].

Algorithm 5 One-Step Phase Retrieval (OSPR) algorithm

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

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

```
// 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] ← ∠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
```

The major advantage of the OSPR algorithm is that it is superfast, and also it can reduce the perceived speckle noise, providing a very good visual quality of holographic projection.

2.2.7 Three-Dimensional (3D) CGH

Section 2.2.2 - Section 2.2.6 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.

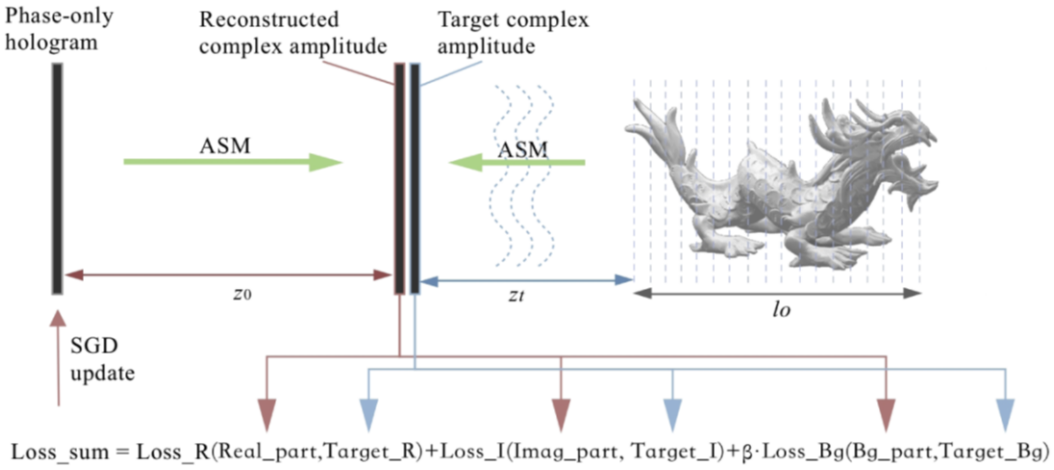


Fig. 2.12 Schematic diagram of the intermediate plane method [15]

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. 2.12 [15].

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.

2.3 Numerical Optimisation Methods

In addition to the conventional CGH algorithms described in Section 2.2, literature review has also found some recent work that compute CGH using numerical optimisation methods [38–40, 15, 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 **??**.

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 [41]. 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 2.3.1 - Section 2.3.4.

2.3.1 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. [41]

To work around the disadvantage, a few variants have emerged, such as AdaGrad [42], RMSProp [43] and Adam [44] 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 [44]. Adam method computes individual adaptive learning rates for different parameters from estimates of first and second moments of the gradients [44]. Although some improvements are observed, it still does not fix entirely.

2.3.2 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. (2.40) 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. [41]

2.3.3 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. [41]

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. [41]

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

L-BFGS algorithm [45] modifies the technique described in Section 2.3.3 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. [41]

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. (2.44). 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. (2.44)). 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: [41]

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

From this expression we can derive a recursive procedure (Algorithm 6) to compute the product $\mathbf{H}_k \nabla f_k$ efficiently.

Algorithm 6 L-BFGS two-loop recursion [41]

$\mathbf{q} \leftarrow \nabla f_k$
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). [41]

2.4 Analytical Solution for Fourier Transforms of Polygons

Most CGH methods are based on pixelated target images and discrete Fourier transforms, and the generated hologram is then displayed on a pixelated spatial light modulator (SLM). However, the real world is not pixelated. So this section aims to investigate the plausibility of finding the analytical solutions of Fourier transforms for non-pixelated shapes.

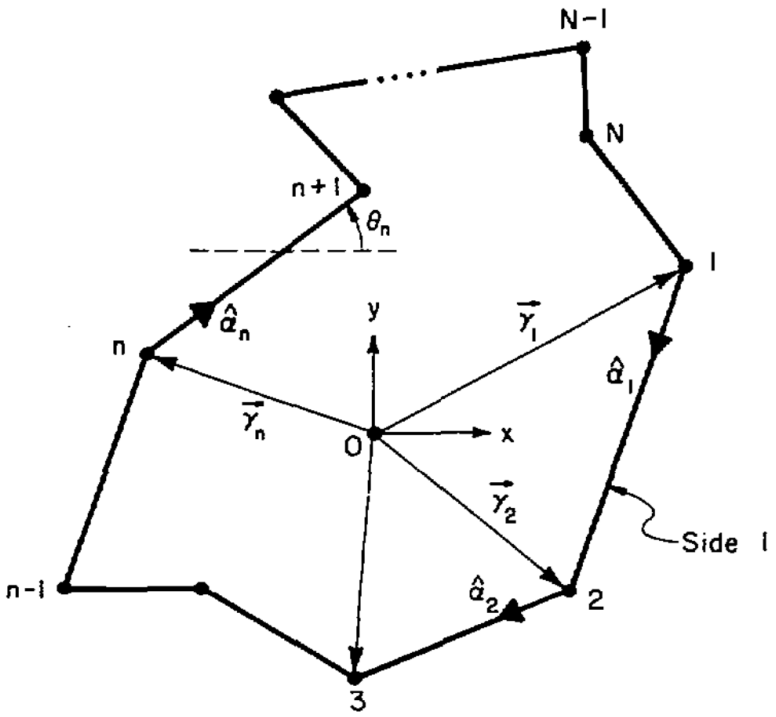


Fig. 2.13 N-sided polygon (Σ) symbol definition [16]

For an N-sided polygon (Σ) as shown in Fig. 2.13, its shape function can be defined as:

$$s(x,y) = \begin{cases} 1 & \text{if } (x,y) \text{ is in } \Sigma \\ 0 & \text{otherwise} \end{cases} \tag{2.47}$$

Then 2D Fourier transform of $s(x,y)$ is defined as:

$$S(u,v) = \iint_{-\infty}^{\infty} s(x,y) e^{j(ux+vy)} dxdy \tag{2.48}$$

For rectangular and circular aperture (Σ), $S(u,v)$ are well known, which are expressed in terms of *Bessel* and *sinc* functions respectively. But for a general N-sided polygon, it is much more complicated. In 1983, Lee and Mittra [16] derived the solution of $S(u,v)$ for the polygon Σ in Fig. 2.13 as:

$$S(u,v) = \sum_{n=1}^{N} e^{j\vec{\omega}\cdot\vec{\gamma_n}} \left[\frac{\hat{n} \times \hat{\alpha_n} \cdot \hat{\alpha_{n-1}}}{(\vec{\omega} \cdot \hat{\alpha_n})(\vec{\omega} \cdot \hat{\alpha_{n-1}})} \right] \tag{2.49}$$

$$\text{where} \quad \hat{n} = +\hat{z} \tag{2.50}$$

$$\vec{\gamma_n} = x_n\hat{x} + y_n\hat{y} \tag{2.51}$$

$$\hat{\alpha_n} = \frac{\vec{\gamma_{n+1}} - \vec{\gamma_n}}{|\vec{\gamma_{n+1}} - \vec{\gamma_n}|} \tag{2.52}$$

$$\vec{\omega} = u\hat{x} + v\hat{y} \tag{2.53}$$

Alternatively, Eq. (2.49) can also be written in terms of the slopes of the polygon sides:

$$S(u,v) = \sum_{n=1}^{N} e^{j\vec{\omega}\cdot\vec{\gamma_n}} \left[\frac{p_{n-1} - p_n}{(u + p_{n-1}v)(u + p_n v)} \right] \tag{2.54}$$

$$\text{where} \quad p_n = \frac{y_{n+1} - y_n}{x_{n+1} - x_n} = tan(\theta_n) \tag{2.55}$$

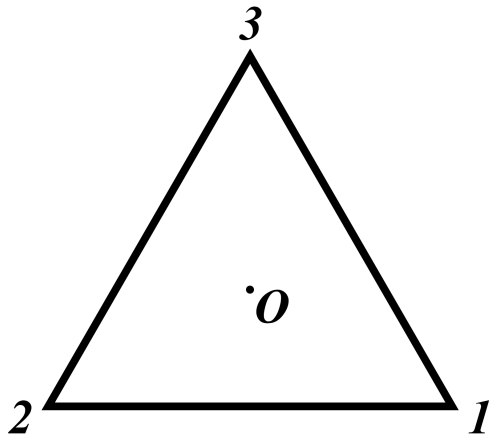


Fig. 2.14 Triangle

Taking an example equilateral triangle as shown in Fig. 2.14 of side length L, we have:

$$\overrightarrow{\gamma_n}\begin{cases} \overrightarrow{\gamma_1} &= \frac{L}{2}\hat{x}+(-\frac{L}{2\sqrt{3}}\hat{y}) \\ \overrightarrow{\gamma_2} &= -\frac{L}{2}\hat{x}+(-\frac{L}{2\sqrt{3}}\hat{y}) \\ \overrightarrow{\gamma_3} &= 0\hat{x}+\frac{L}{\sqrt{3}}\hat{y} \end{cases} \tag{2.56}$$

$$p_n\begin{cases} p_1 &= tan(0)=0 \\ p_2 &= tan(\frac{\pi}{3})=\sqrt{3} \\ p_3 &= tan(-\frac{\pi}{3})=-\sqrt{3} \end{cases} \tag{2.57}$$

Then, using Eq. (2.54), the analytical solution of the Fourier transform $S_\Delta(u,v)$ of the equilateral triangle with side length L can be obtained:

$$\begin{aligned} S_\Delta(u,v) &= e^{j(\frac{L}{2}u-\frac{L}{2\sqrt{3}}y)}\left[\frac{-\sqrt{3}-0}{(u-\sqrt{3}v)(u+0v)}\right] \\ &\quad + e^{j(-\frac{L}{2}u-\frac{L}{2\sqrt{3}}v)}\left[\frac{0-\sqrt{3}}{(u+0v)(u-\sqrt{3}v)}\right] \\ &\quad + e^{j(0u+\frac{L}{\sqrt{3}}v)}\left[\frac{\sqrt{3}-(-\sqrt{3})}{(u+\sqrt{3}v)(u-\sqrt{3}v)}\right] \qquad (2.58) \\ &= \frac{-\sqrt{3}}{u(u-\sqrt{3}v)}\left[e^{jL(\frac{1}{2}u-\frac{1}{2\sqrt{3}}v)}+e^{jL(-\frac{1}{2}u-\frac{1}{2\sqrt{3}}v)}\right] \\ &\quad + \frac{2\sqrt{3}}{(u+\sqrt{3}v)(u-\sqrt{3}v)}e^{j\frac{L}{\sqrt{3}}v} \qquad (2.59) \end{aligned}$$

So theoretically, we can analytically compute continuous hologram for any N-sided polygons. But as non-pixelated complex spatial light modulator is not currently possible, and won't be invented in the near (and probably far) future, the holograms generated by this method still needs to be sampled and quantised. Literature review has found some work that compute CGH based on triangular primitives [46–48], but none of them uses this method to compute CGH. Further discussion on my current progress of implementing this method will be discussed in **??**.

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[49]. 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

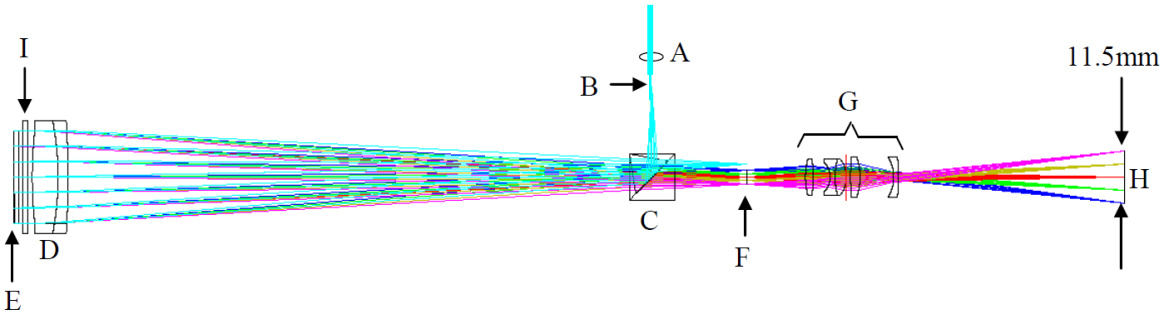
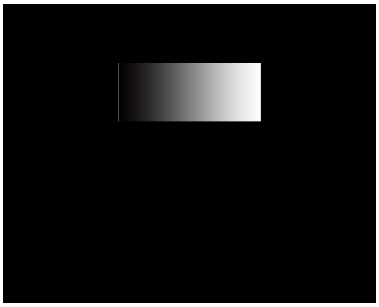


Fig. 3.1 Optical setup [17]

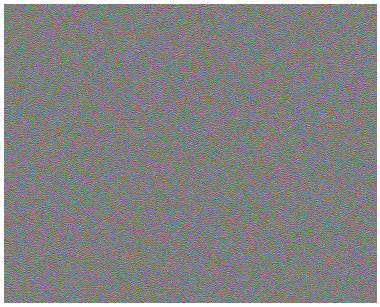
The holographic projector used in this experiment was a Fourier projection system developed by Freeman [17], as shown in Fig. 3.1. The design is consisted of a diode-pumped solid-state (DPSS) 532 nm 50mW laser source, focussed down by an aspheric singlet (A), the focus of which becomes the diffraction 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). [17]

The holograms displayed on the SLM are generated using the one-step phase retrieval (OSPR) algorithm [14] explained in Section 2.2.6, 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).

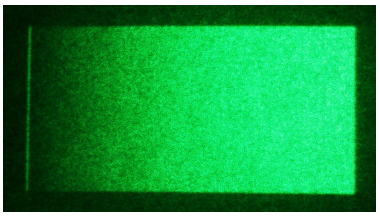
3.2 Determining the Gamma Correction Curve



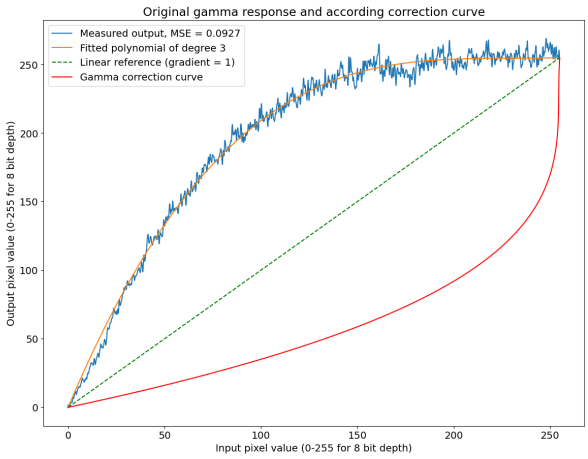
(a) Input linear grey-scale ramp



(b) CGH of Fig. 3.2a



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



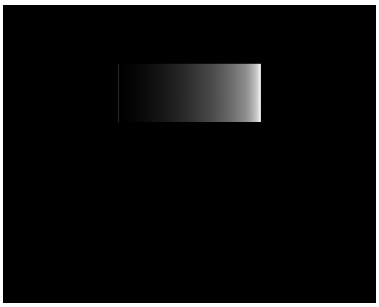
(d) Gamma response curve and according correction curve

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

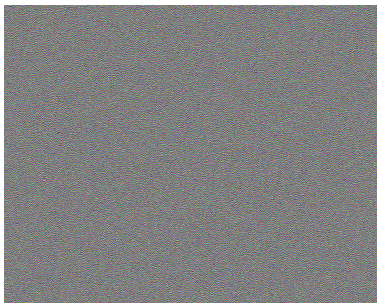
To determine the gamma correction curve of the holographic projection system, the gamma response needs to be measured first. A hologram was generated to form a linear grey-scale ramp of brightness from 0 to 255, as shown in Fig. 3.2a, along with a single pixel white (255) strip at the left end as a fiducial marker to demonstrate the beginning of the grey-scale region [14].

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).

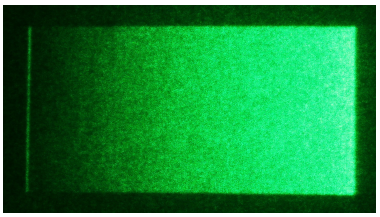
The resultant gamma response curve exhibits a high degree of non-linearity. By taking the mean of the square of the error between the measured output (blue line) and the linear reference (green dashed line), the NMSE of the measured output was calculated to be 0.0927. To correct the gamma response, the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



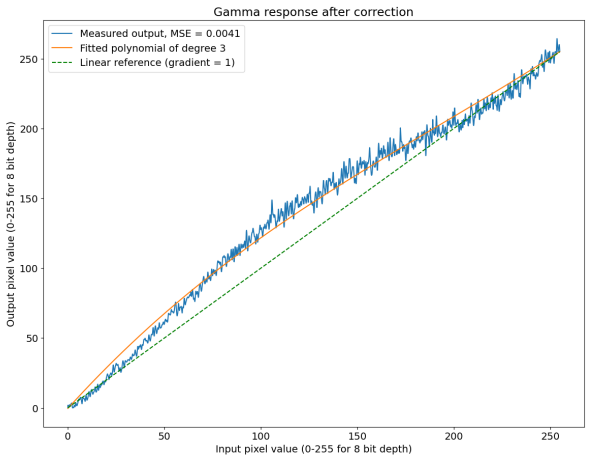
(a) Input gamma corrected ramp



(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

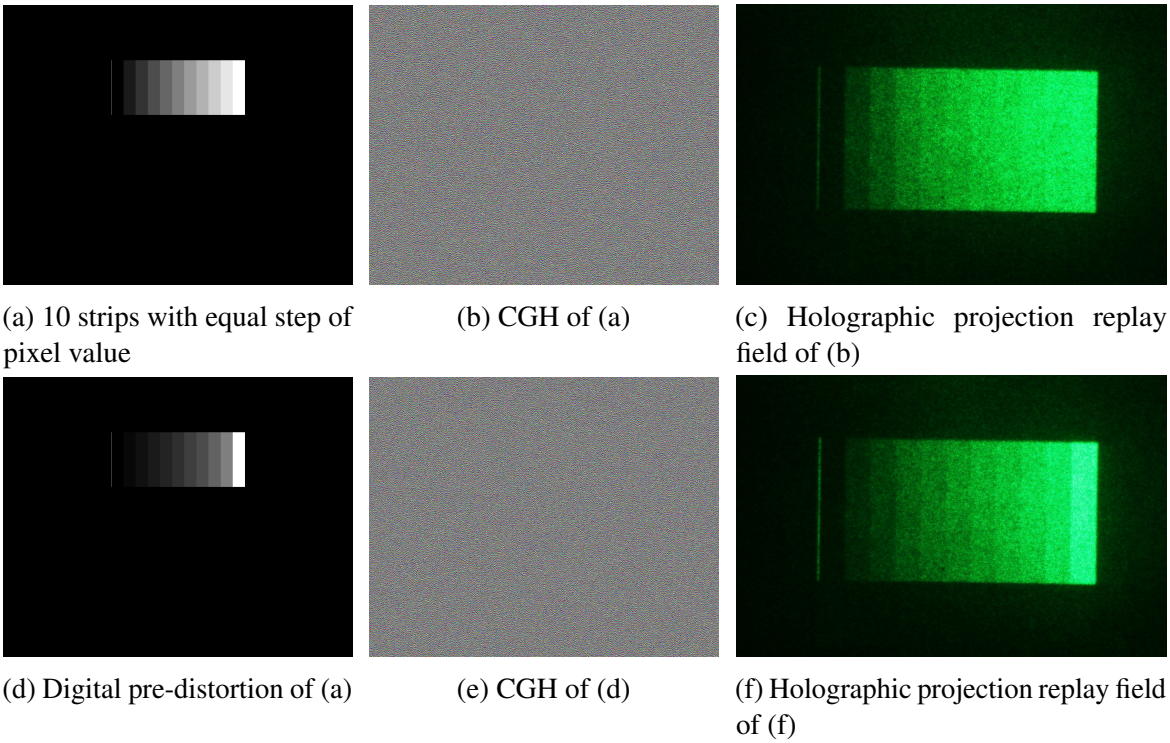
Subsequently, the gamma correction curve was implemented to adjust the grey-scale ramp, achieving the gamma corrected grey-scale ramp as shown in Fig. 3.3a. The gamma corrected projection output was captured as shown in Fig. 3.3c. By using the same method of averaging columns of pixels, the gamma corrected 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

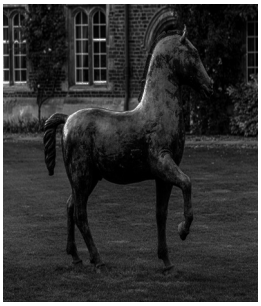
As shown in Fig. 3.4, when CGH is computed for the 10 strips with equal step of pixel value Fig. 3.4a, the right few strips in Fig. 3.4c are barely distinguishable. After applying the correction curve obtained in Section 3.2, it can be seen that each pair of adjacent strips in

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

(a) Sample image 1: City Scene [50]

(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.

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

[10] Yuanbo Deng and Daping Chu. Coherence properties of different light sources and their effect on the image sharpness and speckle of holographic displays. *Scientific Reports*, 7, 12 2017.

[11] 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.

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

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

[14] 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.

[15] 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.

[16] Shung Wu Lee and Raj Mittra. Fourier transform of a polygonal shape function and its application in electromagnetics. *IEEE Transactions on Antennas and Propagation*, 31:99–103, 1983.

[17] 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.

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

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

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

[21] 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.

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

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

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

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

[26] 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.

[27] 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.

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

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

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

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

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

[33] 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.

[34] 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.

[35] 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.

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

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

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

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

[40] 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.

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

[42] 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.

[43] 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.

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

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

[46] Yijie Pan, Yongtian Wang, Juan Liu, Xin Li, and Jia Jia. Fast polygon-based method using 2d fourier analysis of 3d affine transformation, 2013.

[47] Brian B. Maranville. An implementation of an efficient direct fourier transform of polygonal areas and volumes. 4 2021.

[48] Yaping Zhang, Houxin Fan, Fan Wang, Xianfeng Gu, Xiaofan Qian, and Ting-Chung Poon. Polygon-based computer-generated holography: a review of fundamentals and recent progress [invited]. *Applied Optics*, 61:B363, 2 2022.

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

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