

High-resolution Optical Scanning Holography

Huy Nhu Vo

Thesis submitted to the faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of
Master of Science
In
Electrical Engineering

Ting-Chung Poon, Chairman
A. Lynn Abbott, Member
Ahmad Safaai-Jazi, Member

April 27, 2009
Blacksburg, Virginia

Keywords: optical scanning holography, reconstruction, high-resolution

High-resolution Optical Scanning Holography

Huy Nhu Vo

ABSTRACT

Optical scanning holography, which was proposed by Poon^[1], is a fascinating technology to record holographic information. The technique is applied in the operation of scanning holographic microscopy to record the entire three-dimensional volume of a biological specimen in the form of a hologram. With the data captured, a digital reconstruction or decoding is used to reconstruct the hologram of that such specimen. An accurate reconstruction of the recorded data provides with an in-depth analysis in the area where random noise and other imperfection effects may occur. In this thesis, three different approaches of reconstruction process are presented to provide in high-resolution a comparison between theoretical and experimental reconstruction a hologram of fluorescent beads. The first approach is to use only the experimental pinhole hologram recorded to correlate with the hologram of the object to give the reconstruction of the section. The second approach is to use the propagated pinhole hologram to reconstruct at an arbitrary depth. Finally, the third approach is to reconstruct without using the experimental pinhole hologram but with diffraction theory. Comparing these results in high-resolution gives us analysis of the reconstruction noise due to optical aberration.

ACKNOWLEDGEMENTS

This thesis cannot be done without support of many people. I want to express my deepest appreciation to my mentor, Dr. Ting-Chung Poon, for his support and guidance. I am very thankful to have an opportunity to work with him. I am looking forward to have a chance to corporate with him in the future. I am deeply grateful to Dr. Guy Indebetown for his help on getting the holographic data. It has been a pleasure working with him. Also, I want to have warm and sincere thanks to Dr. A. Lynn Abbott and Dr. Ahmad Safaai-Jazi at Virginia Tech for their assistance throughout my program.

In addition, I want to thanks all my friends and everyone at Virginia Tech who helped me to complete my thesis. The financial support from department of Electrical and Computer Engineering at Virginia Tech is gratefully acknowledged.

Finally, my deepest appreciation is devoted to my parents who give me endless support and encouragement to make my dream come true.

TABLE OF CONTENTS

ABSTRACT.....	ii
ACKNOWLEDGMENTS	iii
TABLE OF CONTENTS	iv
LIST OF FIGURES.....	vi
LIST OF TABLES.....	viii
1. CHAPTER 1 INTRODUCTION.....	1
2. CHAPTER 2 OPTICAL SCANNING HOLOGRAPHY	3
2.1 HOLOGRAPHIC BASIS.....	3
2.2 OPTICAL HETERODYNING AND TWO-PUPIL SCANNING SYSTEM.....	8
2.2.1 Optical Heterodyning	8
2.2.2 Two-Pupil Optical Scanning Holography.....	9
2.3 RECONSTRUCTION.....	13
3. CHAPTER 3 SIMULATION AND RESULTS.....	14
3.1 EXPERIMENT SETUP AND PARAMETERS.....	14
3.2 RECONSTRUCTION APPROCHES.....	18
3.2.1 Approach A - Reconstruction using experimental pinhole hologram only.....	18
3.2.2 Approach B – Reconstruction using propagation of a single pinhole hologram ..	24
3.2.3 Approach C – Reconstruction using mathematical $\mathfrak{H}(k_x, k_y; z_R)$ only.....	30

3.3 RESULTS ANALYSIS	35
4. CHAPTER 4 CONCLUSION.....	42
REFERENCES.....	43
APPENDIX A INSTRUCTION TO USE MATLAB PROGRAM	45
APPENDIX B PROGRAM CODE.....	50

LIST OF FIGURES

Figure	Page
1-1. Off-axis point source object recording	4
1-2. Sketch of a holography recording system	6
1-3. Sketch of an optical scanning holography setup based on two-pupil system	10
2-1. Geometry of recording by OSH.....	16
3-1 (a). Amplitude plot: Reconstruction of the pinhole hologram 131207_11 with itself	20
(b). Phase plot: Reconstruction of the pinhole hologram 131207_11 with itself	21
(c). Amplitude plot: Reconstruction of the fluorescent beads hologram 131207_26 with the reference pinhole hologram 131207_11 at depth $z = 85 \mu m$	22
(d). Amplitude plot: Reconstruction of the fluorescent beads hologram 131207_26 with the reference pinhole hologram 131207_18 at depth $z = 50 \mu m$	23
3-2 (a). Amplitude plot: Reconstruction of the pinhole hologram 131207_11 with the pinhole hologram 131207_19 propagated to depth $z = 85 \mu m$	26
(b). Phase plot: Reconstruction of the pinhole hologram 131207_11 with the pinhole hologram 131207_19 propagated to depth $z = 85 \mu m$	27
(c). Amplitude plot: Reconstruction of the fluorescent beads hologram 131207_26 with the reference pinhole hologram 131207_19 propagated to depth $z = 85 \mu m$	28
(d). Amplitude plot: Reconstruction of the fluorescent beads hologram 131207_26 with the reference pinhole hologram 131207_19 propagated to depth $z = 50 \mu m$	29
3-3 (a). Amplitude plot: Reconstruction of the pinhole hologram 131207_11 at depth $z = 85 \mu m$ using only mathematical $\mathfrak{H}(k_x, k_y; z_R)$	31
(b). Phase plot: Reconstruction of the pinhole hologram 131207_11 at depth $z = 85 \mu m$ using only mathematical $\mathfrak{H}(k_x, k_y; z_R)$	32

(c). Amplitude plot: Reconstruction of the fluorescent beads hologram 131207_26 at depth $z = 85 \mu m$ using only mathematical $\mathfrak{H}(k_x, k_y; z_R)$ 33

(d). Amplitude plot: Reconstruction of the fluorescent beads hologram 131207_26 at depth $z = 50 \mu m$ using only mathematical $\mathfrak{H}(k_x, k_y; z_R)$ 34

3-4. Amplitude plot: Reconstruction of the pinhole hologram 131207_11 at resolution 100x100 pixel centered at $(x, y) = (750, 750)$ 37

3-5 (a). Amplitude plot: Reconstruction of the fluorescent beads hologram 131207_26 with the reference pinhole hologram 131207_11 using approach A at depth $z = 85 \mu m$ with resolution 300x300 pixel centered at $(x, y) = (1000, 500)$ 38

(b). Amplitude plot: Reconstruction of the fluorescent beads hologram 131207_26 with the propagated reference pinhole hologram 131207_19 at depth $z = 85 \mu m$ using approach B with resolution 300x300 pixel centered at $(x, y) = (1000, 500)$ 39

(c). Amplitude plot: Reconstruction of the fluorescent beads hologram 131207_26 with the propagated reference pinhole hologram 131207_10 at depth $z = 85 \mu m$ using approach B with resolution 300x300 pixel centered at $(x, y) = (1000, 500)$ 40

(d). Amplitude plot: Reconstruction of the fluorescent beads hologram 131207_26 at depth $z = 85 \mu m$ using approach C with resolution 300x300 pixel centered at $(x, y) = (1000, 500)$ 41

LIST OF TABLES

Table	Page
1. Recording of reference pinhole holograms.....	15
2. Parameters for MATLAB simulation.	17

CHAPTER 1 INTRODUCTION

Holographic recording is a highly sophisticated technology which aims to capture the three-dimensional information into a two-dimensional hologram of complex data. This is an exciting new area with many potential applications. With the growing of computing power in recent years, the power of digital processing has dramatically raised according to Moore's Law. Also, many 3D technologies have been developing in order to recreate the feeling for three-dimension world. However, holography is a unique promising technology which captures the entire information of the scene and all its visual properties including the three dimensional realism. It has a potential in many applications such as cryptography, 3-D display, scanning holographic microscopy, pattern recognition and optical sectioning. The specific form of holographic recording presented in this thesis is optical scanning holography (OSH)^[1-4].

OSH is very promising in delivering a high-resolution three-dimensional recording with the target of use in microscopic imaging of thick biological specimens. The technique of OSH is applied in a wide array of applications especially in biomedical research. The advantage of the OSH is that the entire holographic information of a thick specimen can be captured with a single two-dimensional scan. The idea is based on an optical heterodyning image processor^[1] that can capture the holographic information while preserving the phase information during the recording process. OSH is based on optical heterodyning of two pupils in the optical system where one pupil is modified as uniform function or a plane wave and another pupil is modified as a delta function or a point source. The interference of two pupils can be used to scan two-dimensionally a 3-D object to extract the object's holographic information. The work on this

thesis is based on our previous work^[5]. The data of a slide of fluorescent beads was captured and stored. During the course of the thesis, the recording data is then extracted and manipulated by MATLAB to further explore it in high-resolution. After extracted, the holographic data is used to reconstruct with 3 different methods to have a closer look at the aberration in the optical system.

In chapter 2, the fundamental concept of optical scanning holography is introduced in order to provide readers with some background before representing the simulation process and results. First, the holographic recording of a point source is discussed to introduce the fundamentals of holography. After that, we discuss the physical recording of optical heterodyning and two-pupil scanning system thoroughly. Finally, an explanation of the reconstruction process is presented to give the mathematical background used for the rest of this thesis.

Chapter 3 is the main course of this thesis. Three different approaches in reconstructing hologram are comprehensively explained. For each approach, the simulation procedure is presented with its results. Then, the results are compared in high-resolution side by side to determine the advantage and disadvantage in each method.

Chapter 4 presents the summary of results and concluding remarks as well as discussion of present and future research direction.

CHAPTER 2 OPTICAL SCANNING HOLOGRAPHY

In this chapter, we discuss about the background on Optical Scanning Holography. However, to discuss further into OSH, we want to explore the basic concept of holography. Fundamentally, holography is an image recording technique. It is different from traditional photography on how an image of a 3D scene is recorded on to a 2D plane. The recording of photography is collapsed into the 2D plane as all the depth of the actual scene is lost. On the other hand, holography can fully preserve information of the 3D scene when recording onto a 2D plane.

2.1 HOLOGRAPHY BASIC

In any recording technique, what is actually recorded on the film is the intensity of light, given by $t(x, y) \propto I(x, y) = |\psi_p|^2$ where ψ_p is the complex field distribution of light at the recording plane. Consider a recording of an off-axis point source object located at a distance z_0 illustrated in figure 1-1. The point object can be modeled as a delta function $\delta(x - x_0, y - y_0)$.

The complex field distribution at the plane of the film at the distance z_0 according to Fresnel diffraction is given by

$$\begin{aligned}\psi_p(x, y; z_0) &= \delta(x - x_0, y - y_0) \otimes h(x, y; z_0) \\ &= \exp(-jk_0 z_0) \frac{jk_0}{2\pi z_0} \exp\left(-\frac{jk_0[(x - x_0)^2 + (y - y_0)^2]}{2z_0}\right)\end{aligned}\quad (2-1)$$

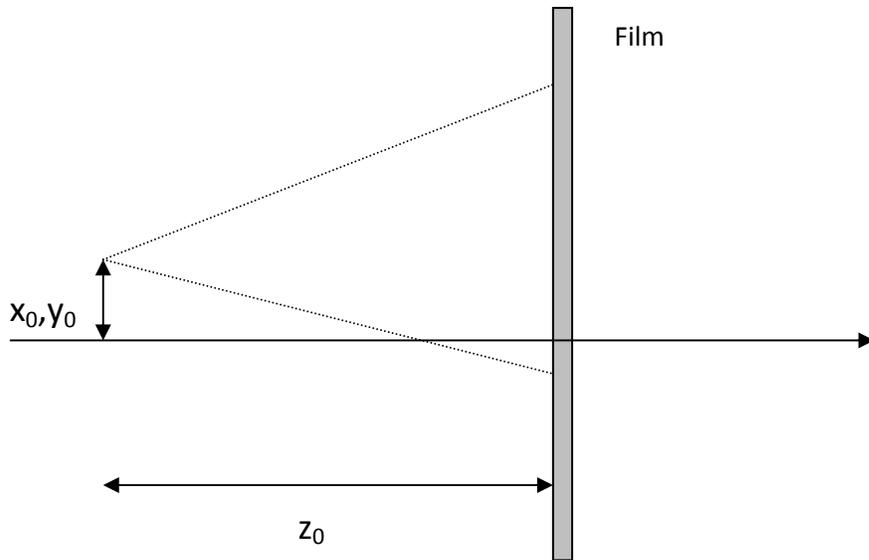


Figure 1-1. Off-axis point source object recording

where $h(x, y; z_0)$ is the spatial impulse response^[1] given by

$$h(x, y; z_0) = e^{-jk_0z} \frac{jk_0}{2\pi z_0} e^{-j\frac{k_0}{2\pi z_0}(x^2+y^2)}$$

and \otimes denotes a convolution. The recorded information on traditional photographic film is given by

$$t(x, y) \propto |\psi_p(x, y; z_0)|^2 = \left(\frac{k_0}{2\pi z_0} \right)^2 \quad (2-2)$$

As we notice, the phase information of the complex field from the point source object is completely lost. In another words, we no longer have the 3D information of the scene. Holography is the technique that was invented to solve this problem. Holography records not only the amplitude but also the phase information of the complex light field on the film. The basic holographic recording of an off-axis point source is illustrated in figure 1-2. The incoming beam is split into two plane waves. One plane wave is used to illuminate the pinhole aperture to generate the point source object while the other one goes directly to the recording film. The wave directly illuminates the recording film is called a *reference wave* represented by ψ_r . The plane wave diffracted by the pinhole aperture is known as an *object wave* represented by ψ_o .

For the above pinhole, the object wave at the film according to Fresnel diffraction is given by

$$\begin{aligned} \psi_o(x, y; z_0) &= \delta(x - x_0, y - y_0) \otimes h(x, y; z_0) \\ &= \exp(-jk_0z_0) \frac{jk_0}{2\pi z_0} \exp\left(-jk_0 \left[(x - x_0)^2 + (y - y_0)^2 \right] / 2z_0\right) \end{aligned} \quad (2-3)$$

where z_0 is the distance between the pinhole and the film.

This object wave is a spherical wave. On the other hand, the reference wave is a plane wave with its field distribution at the film given by $\psi_r(z_0) = a \exp(-jk_0z_0)$, where a is the

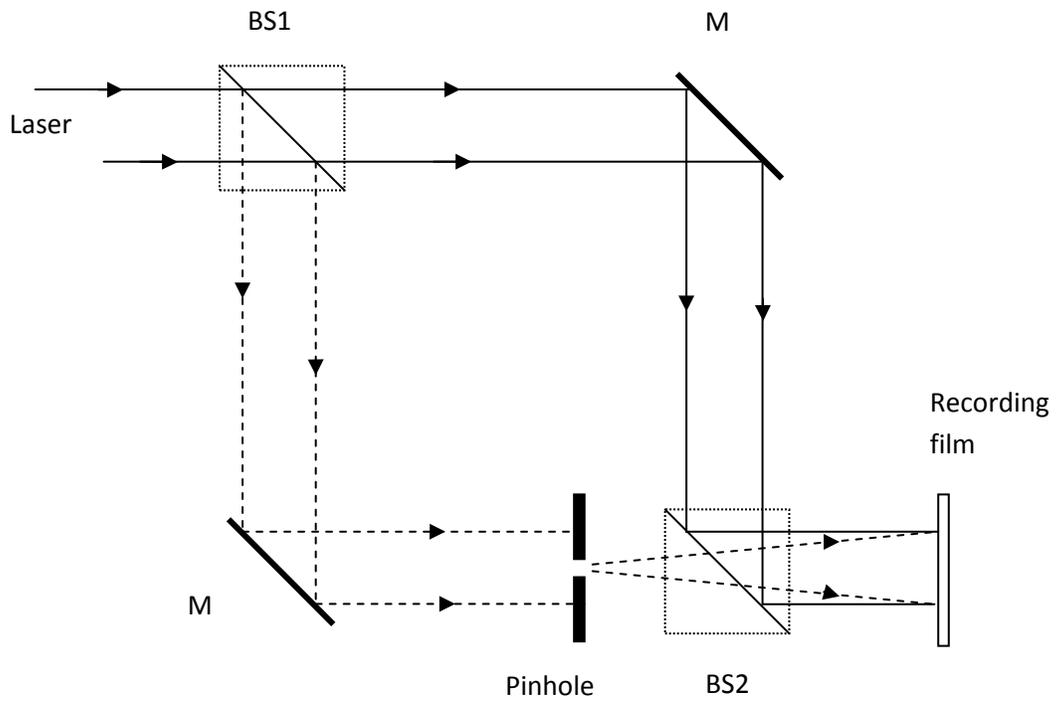


Figure 1-2. Sketch of a holography recording system.
M, mirror; BS1, BS2, beam splitter

amplitude of the plane wave. The intensity of the interference between the object wave and reference wave recorded at the recording film is given by $t(x, y) \propto |\psi_r + \psi_o|^2$. The amplitude transparency recording on the film is then given by

$$\begin{aligned}
 t(x, y) &\propto |\psi_r + \psi_o|^2 \\
 &= \left| a + \frac{k_0}{2\pi z_0} \exp(-jk_0 [(x - x_0)^2 + (y - y_0)^2] / 2z_0) \right|^2 \\
 &= A + B \sin(k_0 [(x - x_0)^2 + (y - y_0)^2] / 2z_0) = FZP(x - x_0, y - y_0; z_0)
 \end{aligned} \tag{2-4}$$

where $A = a^2 + \left(\frac{k_0}{2\pi z_0}\right)^2$ and $B = \frac{k_0}{\pi z_0}$ with $k_0 = 2\pi/\lambda_0$. k_0 is the *propagation constant* and λ_0 is the *wavelength* of the light.

Notice that the phase information is preserved in holographic recording. The information specified the location x_0 , y_0 and z_0 are encoded in the hologram by the spatial variation of a sine function with a quadratic spatial dependence.

The expression in (2-4) called the sinusoidal Fresnel zone plate (FZP) is a hologram of a point source object. For an on-axis point source; i.e. $x_0 = y_0 = 0$, we have the FZP as follows

$$FZP(x, y; z_0) = A + B \sin\left(k_0 \frac{(x^2 + y^2)}{2z_0}\right)$$

In order to retrieve the information from the recorded hologram, a reconstruction or decoding process can be done by illuminating the hologram with a reconstruction wave. We will discuss about the reconstruction process later in this chapter.

2.2 OPTICAL HETERODYNING AND TWO-PUPIL SCANNING SYSTEM

2.2.1 Optical Heterodyning

In the last section, we have discussed basic idea of hologram recording. Next, we will discuss the physical system used to record optical hologram electronically. Optical heterodyne scanning uses two optical beam modified by two pupils in the system. The system combines the two laser beams, reference signal beam and information signal beam separated by frequency Ω . The information signal and reference signal are considered as the plane wave expressed as $Ae^{j\phi} \exp(j\omega_0 t)$ and $B \exp[j(\omega_0 + \Omega)t]$, respectively. Hence, the total field distribution at the photodetector is given by

$$\psi_t = Ae^{j\phi} \exp(j\omega_0 t) + B \exp[j(\omega_0 + \Omega)t]$$

Only the intensity of the distribution field is recorded by the photodetector as a current signal given by

$$\begin{aligned} i &\propto \int |\psi_t|^2 dx dy \\ &= \int \left| Ae^{j\phi} \exp(j\omega_0 t) + B \exp[j(\omega_0 + \Omega)t] \right|^2 dx dy \\ &\propto A^2 + B^2 + 2AB \cos(\Omega t - \phi) \end{aligned} \quad (2-5)$$

The term $A^2 + B^2$ is recorded as the DC signal while the term $2AB \cos(\Omega t - \phi)$ is recorded as the AC signal. Note that, the baseband current can be discarded by using a bandpass filter to obtain the heterodyne current $i_\Omega \propto \cos(\Omega t - \phi)$.

2.2.2 Two-Pupil Optical Scanning Holography

From equation (2-5), we learn it is crucial to have two pupils in the optical system in order to have two optical beams to mix or heterodyne. This system is called a two-pupil heterodyning system as shown in figure 1-3. A laser source at temporal frequency ω_0 is split by a beam splitter BS1. The first beam at temporal frequency ω_0 illuminates the pupil function $p_1(x, y)$. The other beam offset by a frequency shifter (FS) at temporal frequency $\omega_0 + \Omega$ illuminates the pupil function $p_2(x, y)$. The two pupils located at the front focal plane of lenses L1 and L2 are then recombined by the beam splitter BS2. The field distribution in the back of focal plane of lenses L1 and L2 due to Fresnel diffraction is given by

$$P_i\left(\frac{k_0x}{f}, \frac{k_0y}{f}\right) = \mathfrak{F}\{p_i(x, y)\}_{\substack{k_x=k_0x/f \\ k_y=k_0y/f}} \quad (2-7)$$

where \mathfrak{F} denotes Fourier transform defined by $\mathfrak{F}\{f(x, y)\} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \exp(jk_x x + jk_y y) dx dy$.

Then, the field distribution at z away from the scanning mirror due to Fresnel diffraction is given by

$$P_{iz}\left(\frac{k_0x}{f}, \frac{k_0y}{f}\right) = P_i\left(\frac{k_0x}{f}, \frac{k_0y}{f}\right) \otimes h(x, y; z) \text{ for } i = 1, 2 \quad (2-8)$$

and $h(x, y; z)$ is the free-space spatial impulse response at a distance z away from the scanning mirror. The 2D Fourier transform of $h(x, y; z)$ gives us the spatial frequency transfer function in Fourier Optics

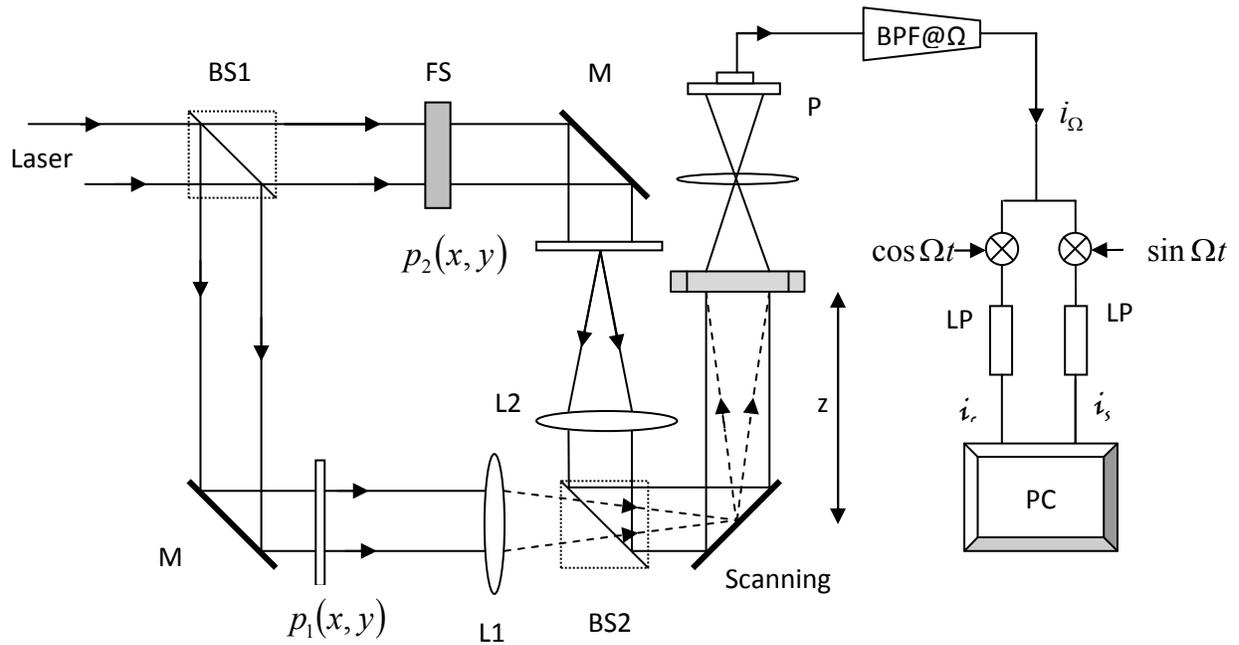


Figure 1-3. Sketch of an optical scanning holography setup based on two-pupil system.
 L1,L2, lenses; M, mirror; BS1,BS2, beam splitter; FS, frequency shifter;
 PD, photodetector; BPF, band pass filter; LPF, low pass filter;
 \otimes electric multiplier; PC, personal computer

$$\begin{aligned}
H(k_x, k_y, z) &= \mathcal{F}\{h(x, y; z)\} \\
&= \exp(-jk_0 z) \exp\left(\frac{j(k_x^2 + k_y^2)z}{2k_0}\right) \quad (2-9a)
\end{aligned}$$

Without any paraxial approximation $k_x^2 + k_y^2 \ll k_0^2$, $H(k_x, k_y; z)$ becomes

$$\mathcal{H}(k_x, k_y, z) = \exp\left(-jk_0 z \sqrt{1 - \frac{(k_x^2 + k_y^2)}{k_0^2}}\right) \quad (2-9b)$$

Equation (2-9a) is simpler and often used in analysis of OSH. However, equation (2-9b) is our choice because we want to have more accurate results. In fact, both forms of $H(k_x, k_y; z)$ and $\mathcal{H}(k_x, k_y; z)$ exist in the MATLAB program but only equation (2-9b) is used to generate the result present in this paper.

The combined of the scanning field $\psi_s(x, y; z)$ is used to scan an object with the transparency of $\Gamma_0(x, y; z)$ located at z from the scanning mirrors as shown in figure 1-3

$$\psi_s(x, y; z) = P_{1z}\left(\frac{k_0 x}{f}, \frac{k_0 y}{f}\right) \exp(j\omega_0 t) + P_{2z}\left(\frac{k_0 x}{f}, \frac{k_0 y}{f}\right) \exp(j(\omega_0 + \Omega)t). \quad (2-10)$$

Finally, the photodetector (PD) records the intensity of the complex scattered field in form of a current signal given by

$$\begin{aligned}
i(x, y; z) &\propto \iint |\psi_s(x', y', z) \Gamma_0(x'+x, y'+y; z)|^2 dx' dy' \quad (2-11) \\
&= \iint \left[\left[P_{1z}\left(\frac{k_0 x'}{f}, \frac{k_0 y'}{f}\right) \exp(j\omega_0 t) + P_{2z}\left(\frac{k_0 x'}{f}, \frac{k_0 y'}{f}\right) \exp(j(\omega_0 + \Omega)t) \right] \times \Gamma_0(x'+x, y'+y; z) \right]^2 dx' dy'
\end{aligned}$$

After going through the bandpass filter (BPF) tuned at frequency of Ω , the current signal becomes

$$\begin{aligned}
i_{\Omega}(x, y; z) &= \text{Re} \left[\iint P_{1z}^* \left(\frac{k_0 x'}{f}, \frac{k_0 y'}{f} \right) P_{2z} \left(\frac{k_0 x'}{f}, \frac{k_0 y'}{f} \right) \times |\Gamma_0(x'+x, y'+y; z)|^2 dx' dy' \exp(j\Omega t) \right] \\
&= \text{Re} [i_{\Omega p}(x, y; z) \exp(j\Omega t)]
\end{aligned} \tag{2-12}$$

where

$$\begin{aligned}
i_{\Omega p}(x, y; z) &= \iint P_{1z}^* \left(\frac{k_0 x'}{f}, \frac{k_0 y'}{f} \right) P_{2z} \left(\frac{k_0 x'}{f}, \frac{k_0 y'}{f} \right) \times |\Gamma_0(x'+x, y'+y; z)|^2 dx' dy' \\
&= P_{1z} \left(\frac{k_0 x}{f}, \frac{k_0 y}{f} \right) P_{2z}^* \left(\frac{k_0 x}{f}, \frac{k_0 y}{f} \right) \oplus |\Gamma_0(x, y; z)|^2
\end{aligned} \tag{2-13}$$

where \oplus denotes the correlation^[1]

From equation (2-13), we can define the optical transfer function (OTF) of the system to be

$$\begin{aligned}
OTF(k_x, k_y; z) &= \mathfrak{F} \{ i_{\Omega p}(x, y; z) \} / \mathfrak{F} \{ |\Gamma_0(x'+x, y'+y; z)|^2 \} \\
&= \mathfrak{F} \left\{ P_{1z}^* \left(\frac{k_0 x'}{f}, \frac{k_0 y'}{f} \right) P_{2z} \left(\frac{k_0 x'}{f}, \frac{k_0 y'}{f} \right) \right\}
\end{aligned} \tag{2-14}$$

with P_{1z} and P_{2z} being defined in equation (2-7).

In OSH, we let one pupil function as a point source and another pupil as a plane wave. The interference of these specified pupils generates a complex Fresnel-zone-plate impulse response. Mathematically, we define the two-pupil functions as $p_1(x, y) = 1$ and $p_2(x, y) = \delta(x, y)$. With this choice of these pupil functions, the OTF of the heterodyne scanning system is given by^[1]

$$OTF_{OSH}(k_x, k_y; z) = H^*(k_x, k_y; z) \tag{2-15}$$

The corresponding spatial impulse response is

$$h_{OSH}(x, y; z) = \mathfrak{F}^{-1} \{ OTF_{OSH}(k_x, k_y; z) \} \tag{2-16}$$

Finally, the current signal of the hologram is processed by a lock-in amplifier (multiplier \otimes and low pass filter LPF shown in figure 1-3) to give two current signals i_c and i_s , which is stored and manipulated digitally to give a 2-D complex hologram as follow

$$H_c(x, y) = \int |\Gamma_0(x, y, z)|^2 \otimes h_{OSH}(x, y, z) dz \quad (2-17)$$

where \otimes defines a 2D convolution integral over the transverse coordinates^{[1][4]}.

2.3 RECONSTRUCTION

Once we have the data of the hologram, a reconstruction or decoding can be obtained by illuminating it with the reconstruction wave. For digital reconstruction, we simply convolute the hologram with the spatial impulse response in order to simulate Fresnel diffraction. The reconstruction of the hologram $H_c(x, y)$ focused at a chosen depth z_R is given by

$$\begin{aligned} R(x, y; z_R) &= H_c(x, y) \otimes h(x, y; z_R) \\ &= \mathcal{F}^{-1} \{ \mathcal{F} \{ H_c(x, y) \} \times \mathcal{F} \{ h(x, y; z_R) \} \} \end{aligned} \quad (2-18)$$

From equation (2-15), we can write (2-18) as

$$R(x, y; z_R) = \mathcal{F}^{-1} \{ \mathcal{F} \{ H_c(x, y) \} \times OTF_{OSH}^*(k_x, k_y; z_R) \} \quad (2-19)$$

CHAPTER 3 SIMULATION AND RESULTS

3.1 EXPERIMENTAL SETUP AND PARAMETERS

In 2009, a sample of the fluorescent beads was recorded by optical scanning holography at Virginia Tech by Professor Indebetow^[4]. The experiment sample is a slide of Duke fluorescent beads with $3\mu\text{m}$ in diameter, excitation at 542nm and emission at 612 nm. The pupils are illuminated with light from diode-pumped solid-state laser with the wavelength at 532nm. The radius of curvature of the scanning spherical wave is $z_0 = 85\mu\text{m}$ for a nominal object plane as shown in figure 2-1. This is a 3-D test sample with the distance between the two planes with the most beads approximately $35\mu\text{m}$. The beads is likely to stick on top surface of the slide (at $z = 85\mu\text{m}$) and the bottom surface (at $z = 50\mu\text{m}$). The microscope objective, lens L, is a Mitutoyo 20 with focal length of 10mm and a nominal NA of 0.42. The recording of a series of reference pinhole hologram (diameter of the pinhole is $0.5\mu\text{m}$) is stored digitally in computer with the filename 131207_08 to 131207_19 as shown in table 1. The recording of the fluorescent beads hologram is stored with filename 131207_26.

Δz	Data file	Depth(z)
-15 μm	131207_8	100 μm
-10 μm	131207_9	95 μm
-5 μm	131207_10	90 μm
0 μm	131207_11	85 μm
5 μm	131207_12	80 μm
10 μm	131207_13	75 μm
15 μm	131207_14	70 μm
20 μm	131207_15	65 μm
25 μm	131207_16	60 μm
30 μm	131207_17	55 μm
35 μm	131207_18	50 μm
40 μm	131207_19	45 μm

Table 1. Recording of reference pinhole holograms

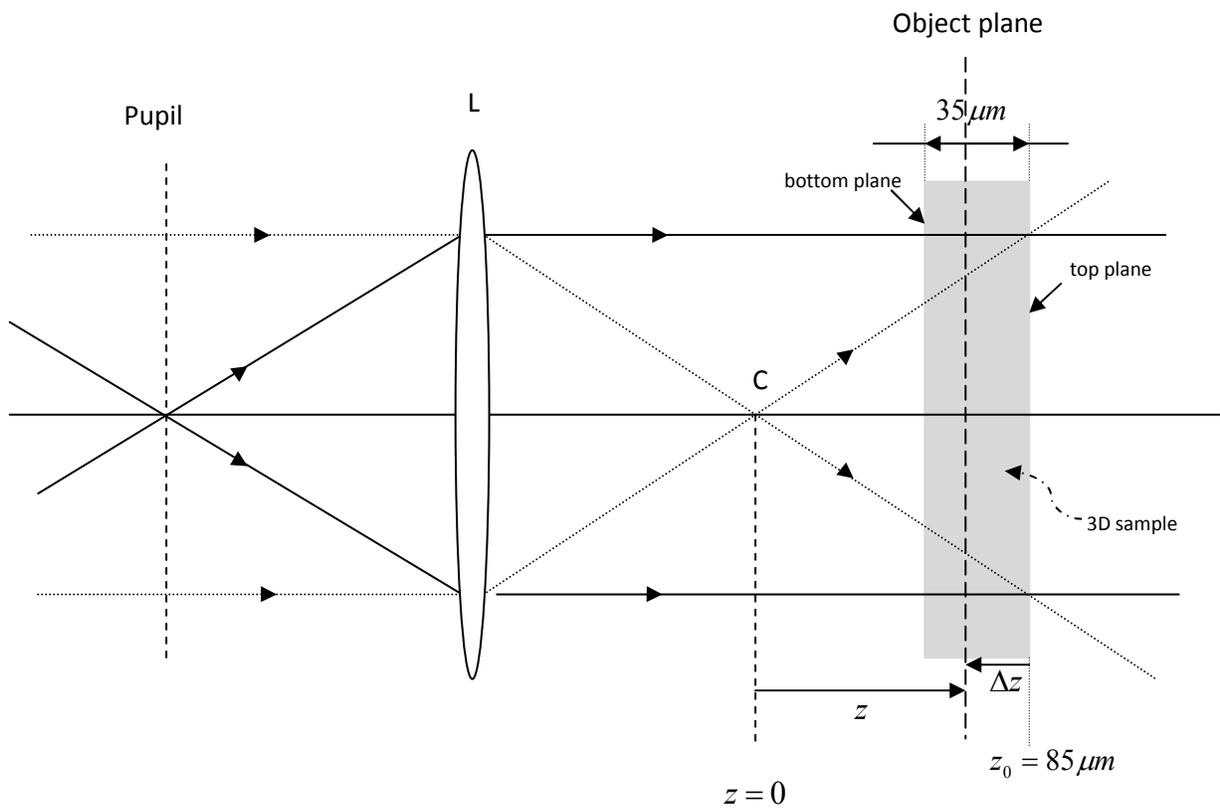


Figure 2-1. Geometry of recording by OSH.

Number of lines	1500
Line separation	0.2 μ m
Field Length	300 μ
Number of valid samples	10000
Number of invalid samples	8000
Number of initial invalid samples	1000
Sampling frequency	50000Hz
Modulation frequency	12500Hz
Filter bandpass	7500Hz
Radius of PSF	150

Table 2. Parameters for MATLAB simulation.

The holographic current signal i_{Ω} is stored digitally into two set of signal files ending with R and S representing the signal and modulated signal, respectively. The hologram data is extracted from these R and S files using “*extracted_data.m*” MATLAB program. The signal data is decoded with the parameters listed in table 2. The extracted data is then stored into their respective “.mat” (such as “*12107_08.mat*”) data file in the form of two matrices representing the real and imaginary parts of the recorded hologram (variable name *realHolo* and *imgHolo*). These matrices will be used for holographic reconstruction using MATLAB programs “*reconstruct_option1.m*”, “*reconstruct_option2.m*” and “*reconstruct_option3.m*”.

3.2 RECONSTRUCTION APPROACHES

In this portion of the chapter, we present 3 different approaches to reconstruct the hologram of the fluorescent beads recorded as mentioned above. The purpose is to demonstrate the effect of the aberration present in the system. Our focus is to reconstruct the hologram of the beads at the depth where the beads has the best “lumps” at $z = 85\mu\text{m}$ plane and the best “dilute” at $z = 50\mu\text{m}$ plane.

3.2.1 Approach A - Reconstruction using experimental pinhole holograms only

The first approach is to use only the experimental pinhole hologram as the reference for reconstruction using “*reconstruct_option1.m*”. The mathematical representation of the reconstruction is given by

$$R(x, y; z_R) = \mathcal{F}^{-1} \left\{ \mathcal{F} \{ \text{Object} \} \times \mathcal{F}^* \left\{ \text{Reference}_{z_R} \right\} \right\}$$

where Reference_{z_R} is the reference pinhole hologram at depth $z_R = z_0 - \Delta z$. $R(x, y; z_R)$ is the reconstruction of the object hologram at the depth of the reference hologram z_R . The MATLAB code expresses this reconstruction process

```
RefHologram = fftshift(fft2(RefHologram));
ObjHologram = fftshift(fft2(ObjHologram));
Reconstruction =
    fftshift(ifft2(ifftshift(ObjHologram.*conj(RefHologram))));
```

Using this method, we first find a reconstruction of a pinhole hologram at $z = 85\mu\text{m}$. This can be done by correlation of the pinhole hologram 131207_11 with itself. Setting both the reference and object hologram to be the pinhole hologram of 131207_11, we can retrieve the

reconstruction of the pinhole at $z = 85\ \mu\text{m}$ with the amplitude shown in figure 3-1(a) and phase shown in figure 3-1(b).

Extend this approach to reconstruct the section of the fluorescent beads for two planes with the depth of $85\ \mu\text{m}$ and $50\ \mu\text{m}$ corresponding to the reference pinhole hologram of 131207_11 and 131207_18, respectively. The fluorescent beads hologram 131207_26 is set as the object hologram and the pinhole hologram 131207_11 with $\Delta z = 0\ \mu\text{m}$ is set as the reference hologram. The result is the reconstruction of the fluorescent beads focused on the plane with the best lumps at $z = 85\ \mu\text{m}$ as shown in the figure 3-1(c). Whereas, the reconstruction of the fluorescent beads hologram 131207_26 and the reference pinhole hologram 131207_18 ($\Delta z = 35\ \mu\text{m}$) gives a reconstruction at the best dilute plane $z = 50\ \mu\text{m}$ shown in figure 3-1(d).

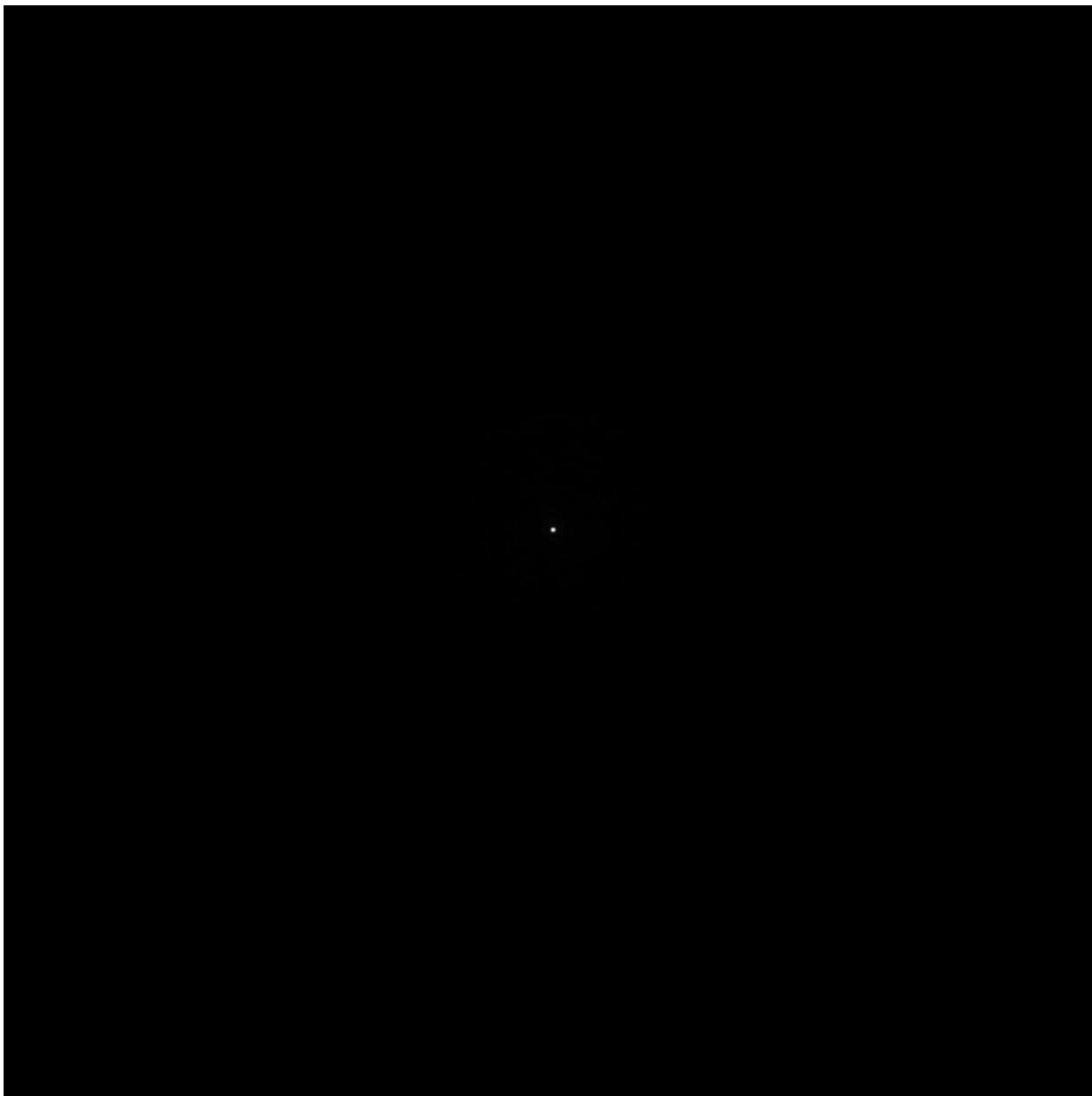


Figure 3-1(a). Amplitude plot: Reconstruction of the pinhole hologram 131207_11 with itself.



Figure 3-1(b). Phase plot: Reconstruction of the pinhole hologram 131207_11 with itself.

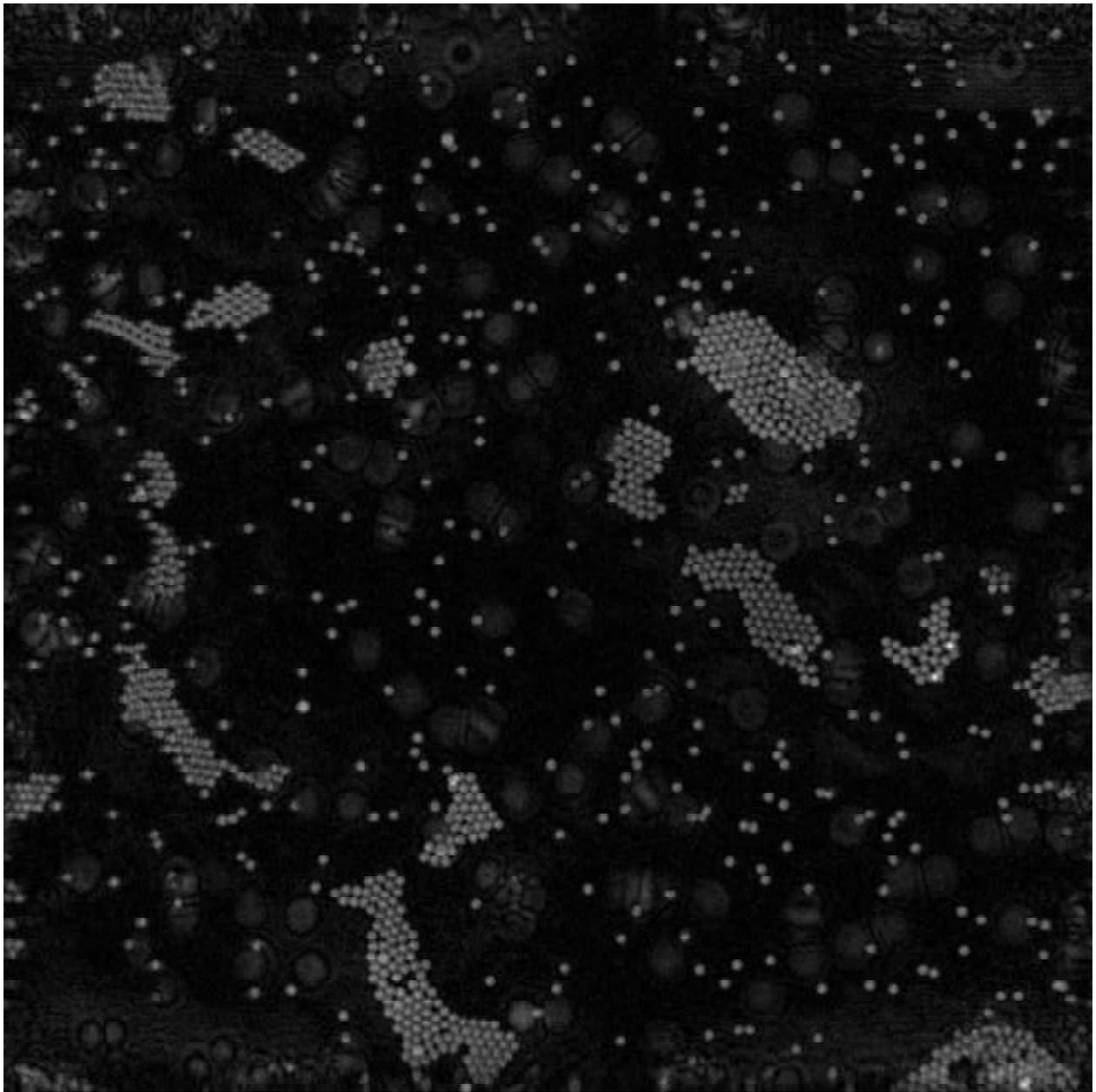


Figure 3-1(c). Amplitude plot: Reconstruction of the fluorescent beads hologram 131207_26 with the reference pinhole hologram 131207_11 at depth $z = 85\mu m$.

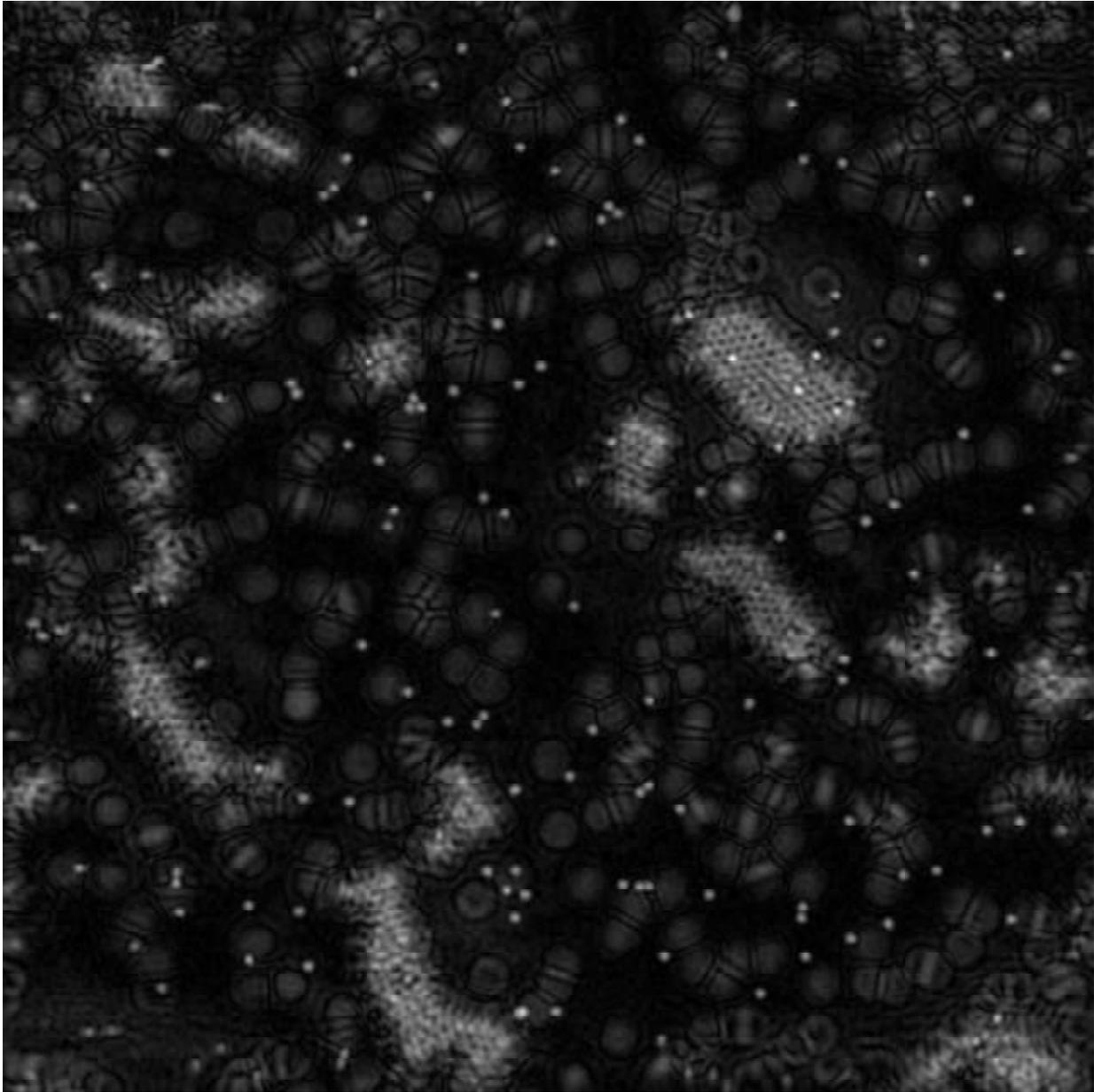


Figure 3-1(d). Amplitude plot: Reconstruction of the fluorescent beads hologram 131207_26 with the reference pinhole hologram 131207_18 at depth $z = 50 \mu m$.

3.2.2 Approach B - Reconstruction using propagation of a single pinhole hologram

This approach uses one pinhole hologram as the reference hologram and propagates it to reconstruct the object hologram at an arbitrary plane z . The propagating of the reference hologram for a distance z_R is obtained by using Fresnel propagation. The reconstruction is given by

$$R(x, y; z_0) = \mathcal{F}\left\{\mathcal{F}\{\text{Object}\} \times \left[\mathcal{F}\{\text{Reference}_{z_R}\} \times \mathcal{H}(k_x, k_y; z_R)\right]^*\right\}$$

The corresponding MATLAB code for equation is

```
RefHologram = fftshift(fft2(RefHologram));  
ObjHologram = fftshift(fft2(ObjHologram));  
P = RefHologram.*P;  
Reconstruction = fftshift(ifft2 (ifftshift(ObjHologram.*conj(P))));
```

where P is the spatial frequency transfer function $\mathcal{H}(k_x, k_y; z_R)$ given by equation (2-9b)

The rest of the MATLAB program for this reconstruction approach can be found in “*reconstruct_option2.m*”. The program varies the distance z_R to propagate the reference pinhole hologram to different depth. The propagated pinhole hologram then can be used as the reference hologram at the new depth, z_R away from its original depth. Therefore, we can practically generate any number of reference holograms with just one pinhole hologram.

In particular, we pick the object hologram as the pinhole hologram 131207_11 and the reference hologram the pinhole hologram 131207_19. The reference pinhole hologram 131207_19 is propagated a distance $\Delta z = 40 \mu\text{m}$, $\mathcal{H}(k_x, k_y, z_R = 45 \mu\text{m})$, to the plane of the object pinhole hologram 131207_11. The correlation of the object hologram 131207_11 and the propagated reference hologram 131207_19 generates the reconstruction of the pinhole hologram with the amplitude shown in figure 3-2(a) and phase shown in figure 3-2(b).

Using this method, we have an ability to sweep through an array of depth by varying the depth z_R . Next, we select the object hologram to be the fluorescent beads hologram 131207_26 and the reference pinhole hologram to be the pinhole hologram 131207_19. The reference hologram is then propagated by the choice of z with an initial value of $40\mu\text{m}$, a final value of $100\mu\text{m}$ and an increment of $5\mu\text{m}$. The program will reconstruct the fluorescent beads hologram 131207_26 at depths of 40, 45, 50, 55...100 μm . Particularly, we are interested in two depths where many beads are in focus located at $85\mu\text{m}$ and $50\mu\text{m}$ as shown in figure 3-2(c) and figure 3-2(d), respectively. They are corresponding to the best lumps and best dilute planes.

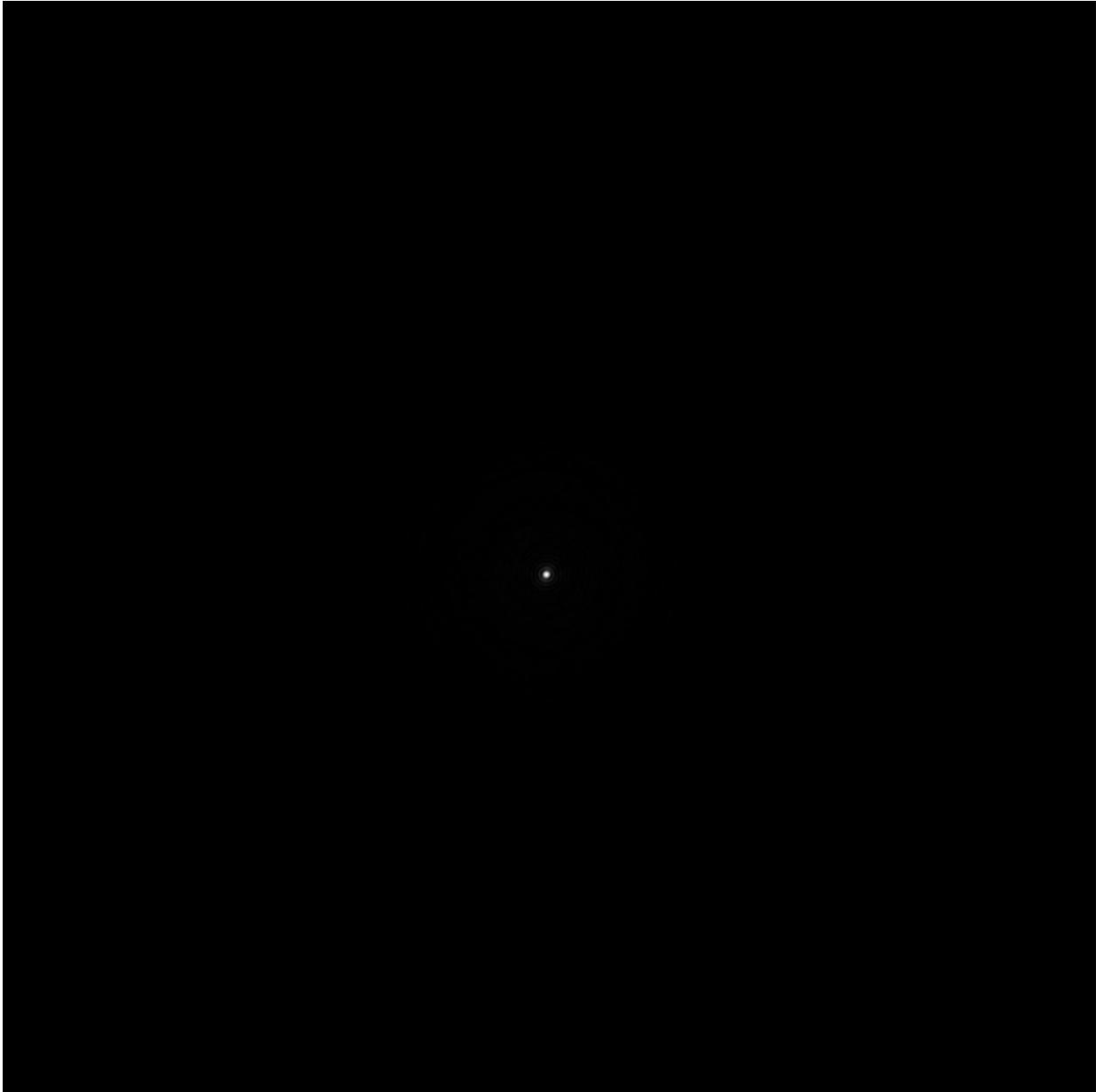


Figure 3-2(a) Amplitude plot: Reconstruction of the pinhole hologram 131207_11 with the pinhole hologram 131207_19 propagated to depth $z = 85 \mu m$.



Figure 3-2(b) Phase plot: Reconstruction of the pinhole hologram 131207_11 with the pinhole hologram 131207_19 propagated to depth $z = 85 \mu\text{m}$.

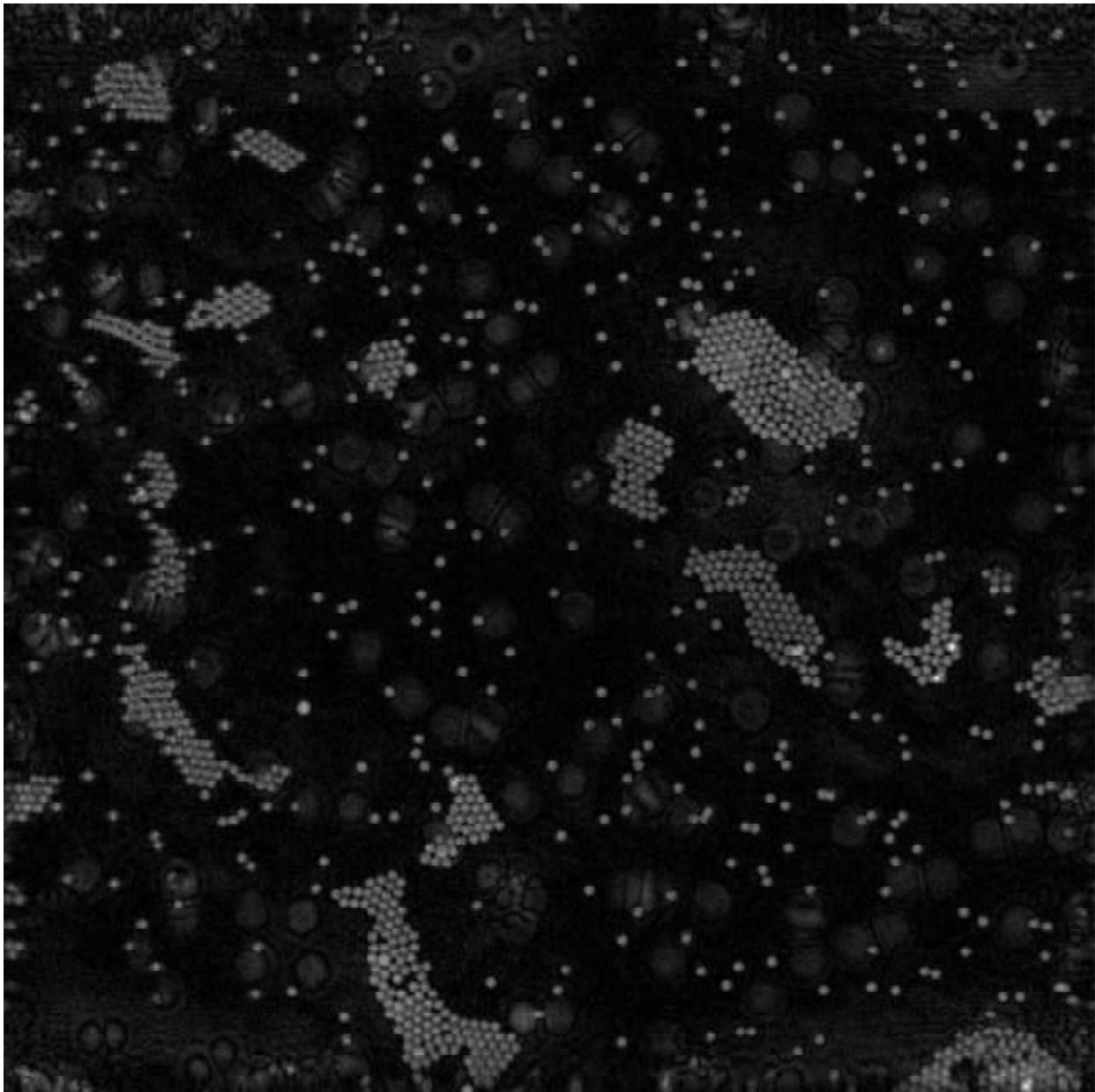


Figure 3-2(c) Amplitude plot: Reconstruction of the fluorescent beads hologram 131207_26 with the reference pinhole hologram 131207_19 propagated to depth $z = 85 \mu m$.

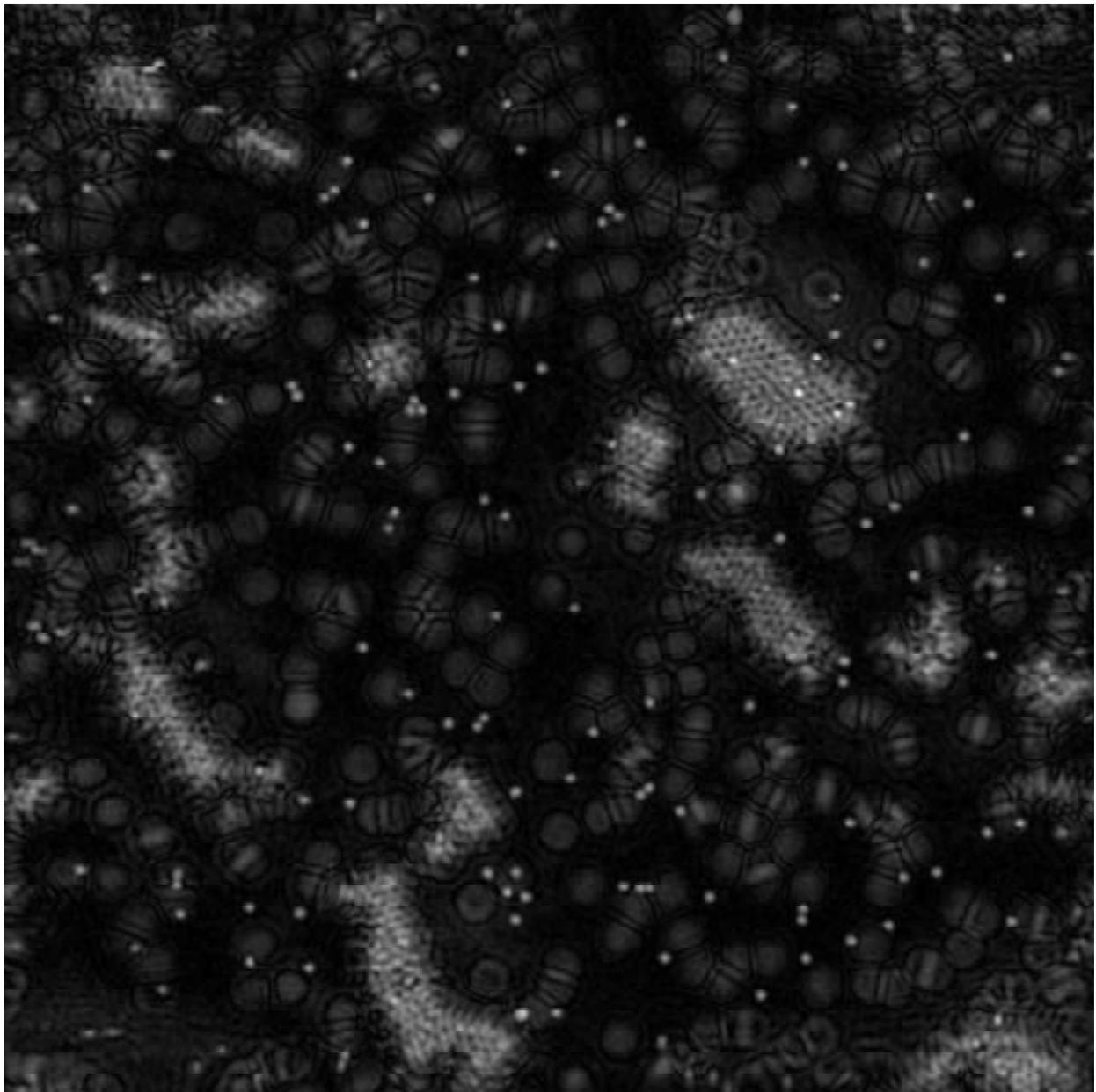


Figure 3-2(d) Amplitude plot: Reconstruction of the fluorescent beads hologram 131207_26 with the reference pinhole hologram 131207_19 propagated to depth $z = 50 \mu m$.

3.2.3 Approach C - Reconstruction using mathematical $\mathcal{H}(k_x, k_y; z_R)$ only

The final approach is simply using mathematical calculation to reconstruct the object hologram without the use of any pinhole hologram as the reference hologram. Mathematical representation of this reconstruction method is given by

$$R(x, y, z_0) = \mathcal{F}^{-1}\{\mathcal{F}\{\text{Object}\} \times \mathcal{H}(k_x, k_y; z_R)\}$$

The MATLAB code corresponding is as follows

```
Hologram = fftshift(fft2(Hologram));  
Reconstruction = ifft2(ifftshift(Hologram.*(P)));
```

where P is $\mathcal{H}(k_x, k_y; z_R)$ given by equation (2-9b). The rest of the MATLAB program can be found in “*reconstruct_option3.m*”.

Similar to Approach B, the object hologram can be reconstructed at any arbitrary depth. To confirm the mathematical calculation, we first reconstruct the pinhole hologram 131207_11 at the depth of 85 μm . The reconstruction amplitude and phase of the pinhole is shown in figure 3-3(a) and 3-3(b). Next, we reconstruct the fluorescent beads 131207_26 at the planes with the most beads 85 μm and 50 μm . The reconstruction amplitude are shown in figure 3-3(c) and figure 3-3(d).

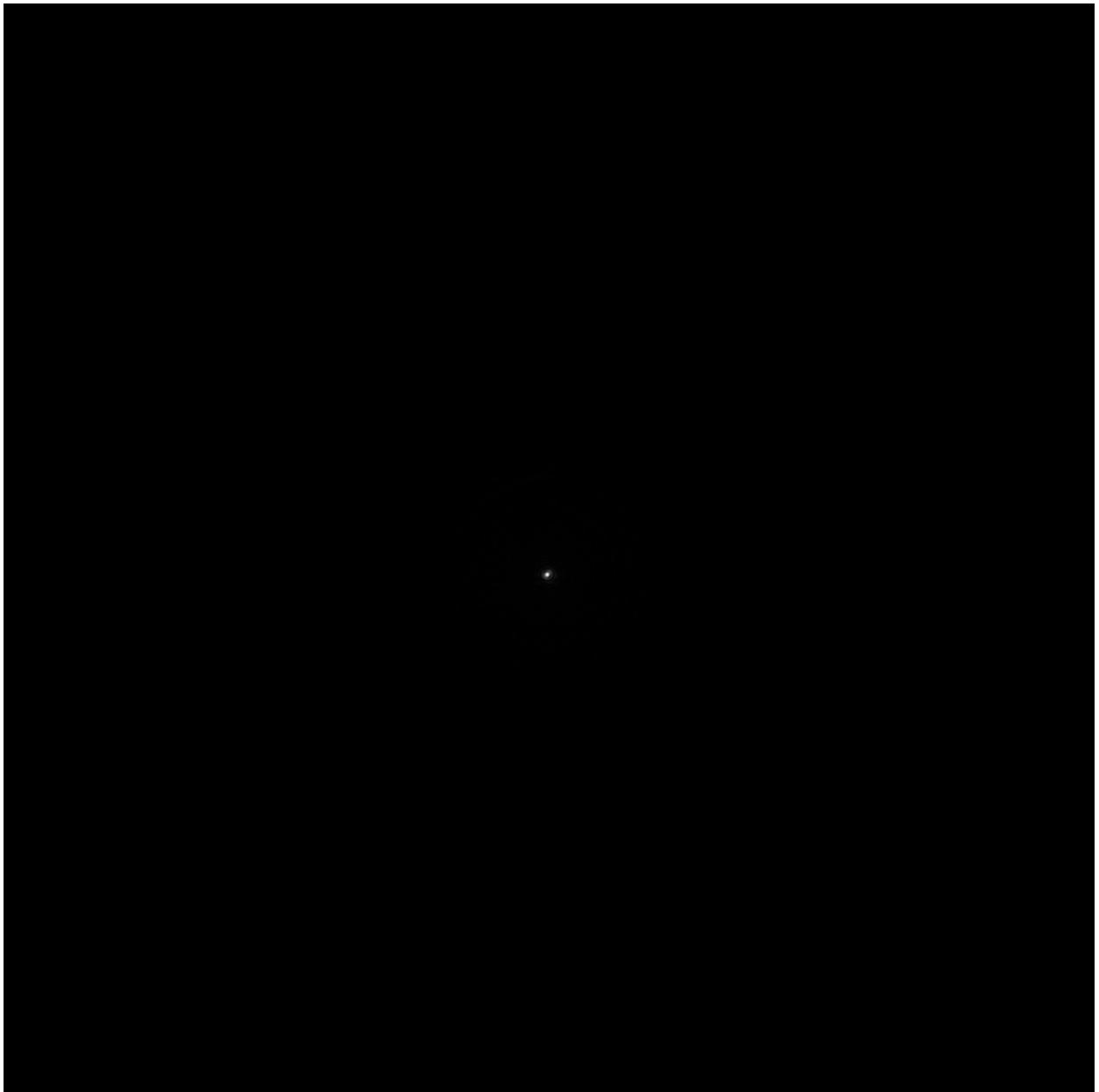


Figure 3-3(a) Amplitude plot: Reconstruction of the pinhole hologram 131207_11 at depth $z = 85 \mu m$ using only mathematical $\mathfrak{H}(k_x, k_y; z_R)$.



Figure 3-3(b) Phase plot: Reconstruction of the pinhole hologram 131207_11 at depth $z = 85 \mu m$ using only mathematical $\mathfrak{H}(k_x, k_y; z_R)$.

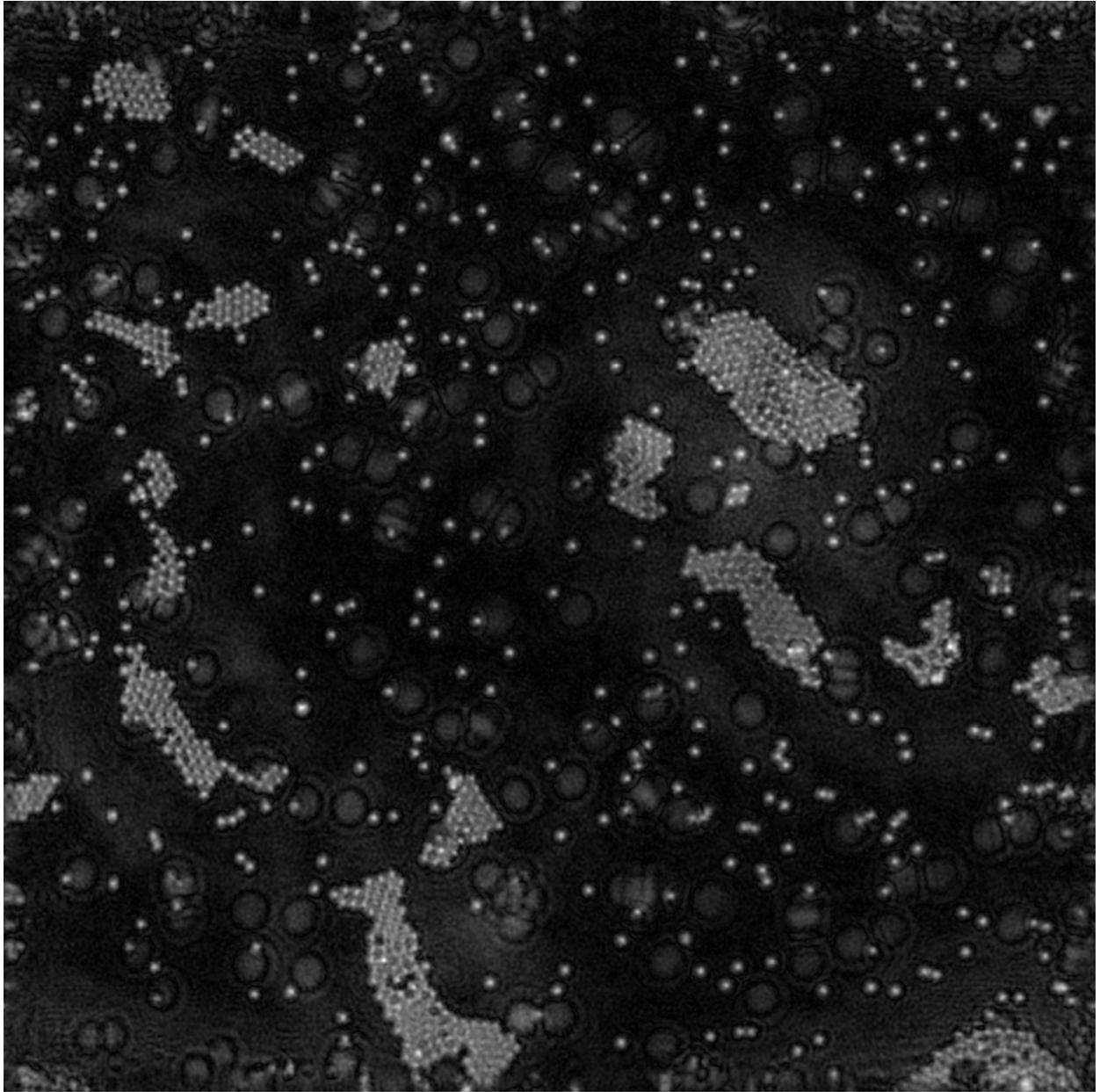


Figure 3-3(c) Amplitude plot: Reconstruction of the fluorescent beads hologram 131207_26 at depth $z = 85 \mu m$ using only mathematical $\mathfrak{H}(k_x, k_y; z_R)$.

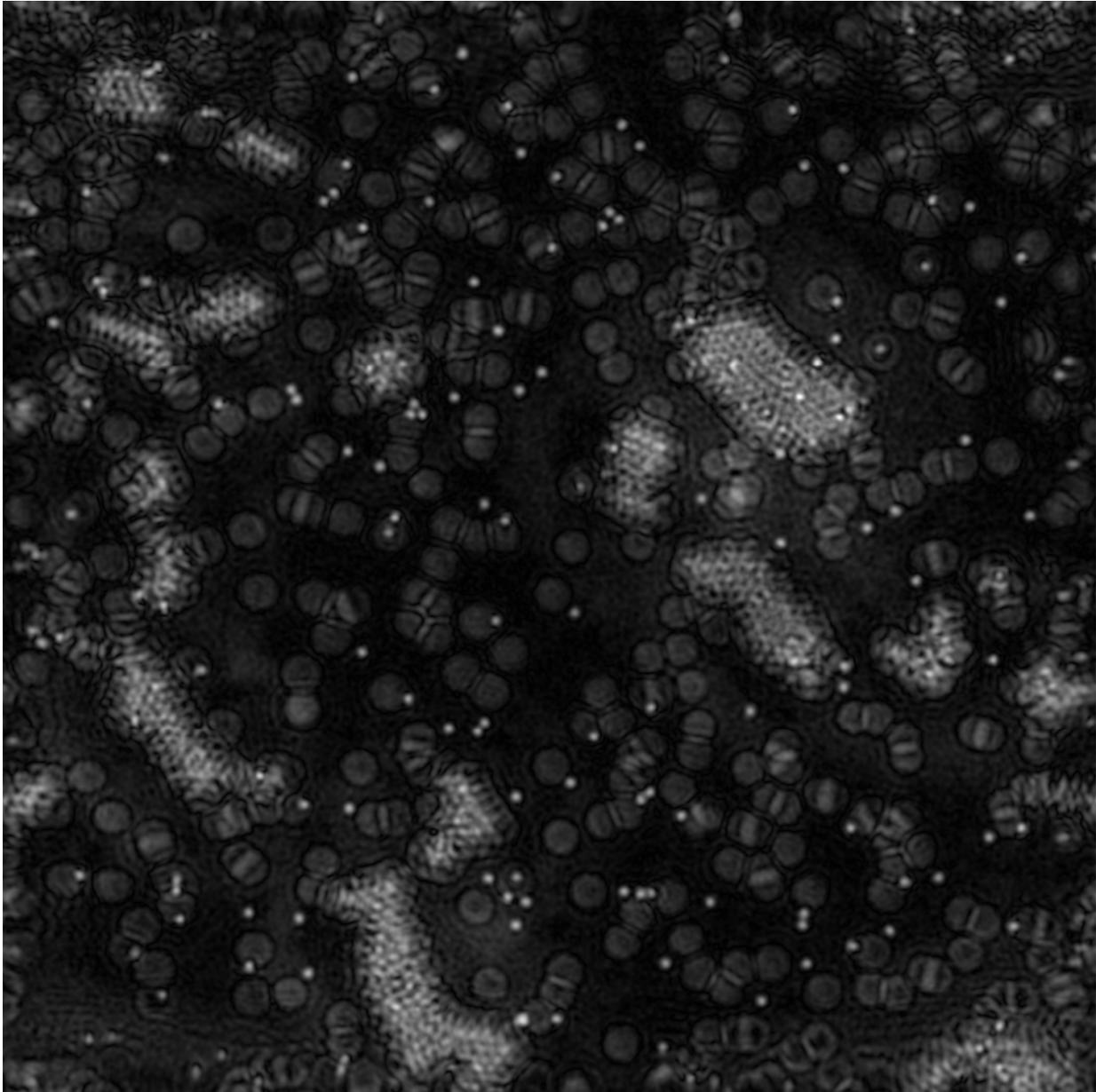


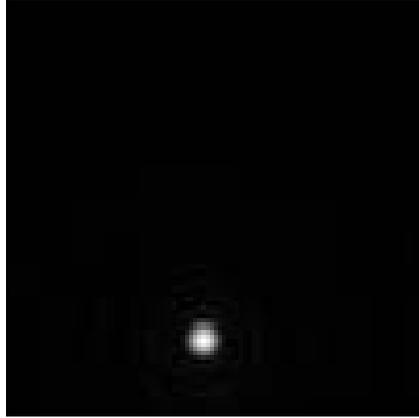
Figure 3-3(d) Amplitude plot: Reconstruction of the fluorescent beads hologram 131207_26 at depth $z = 50 \mu m$ using only mathematical $\mathfrak{H}(k_x, k_y; z_R)$.

3.3 RESULTS ANALYSIS

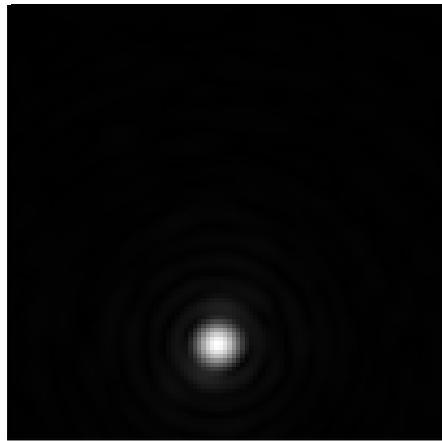
Once we have the reconstruction using different methods, a comparison of results is discussed in this section. It is difficult to tell the difference with the results shown previously so we need to zoom in to have a closer look. The MATLAB programs have an option to reduce the image size with a given lower bound and upper bound. If the lower bound of x,y is set to 700 and upper bound of x,y set to 800. Recall the default resolution of the reconstruction hologram is 1500x1500 pixel. That means we have a 100x100 pixel reconstruction of the pinhole with the center at $(x,y) = (750,750)$. Figure 3-4 shows the pinhole reconstruction with such resolution, the output images shown in figure 3-4 is zoomed in 15 times compare to figures 3-1(a), 3-2(a) and 3-3(a) respectively. Note that the pinhole is not at the center of the hologram. With reference to figure 3-4, comparing (a), (b) and (c), we notice that pinhole (a) has the smallest size. Measured in pixel, the reconstructed pinhole in figure 3-4(a) is approximately 4 pixels in diameter, corresponding to the size of $0.8\mu\text{m}$. Recall that the size of the physical pinhole is $0.5\mu\text{m}$. The diameter of pinhole in figure 3-4(b) is measured around 9 pixels or $1.8\mu\text{m}$ while the diameter of the pinhole in figure 3-4(c) is measured approximately 21 pixels or $4.1\mu\text{m}$. Now, we take a look at reconstruction of the fluorescent beads 131207_26 at depth of $z = 85\mu\text{m}$ shown in figure 3-5. Figure 3-5 (a), (b) and (d) shows the reconstruction of the fluorescent beads obtained in figure 3-1(c), 3-2(c) and 3-3(c) magnified at the resolution of 300x300 pixel and centered at $(x,y) = (1000,500)$, respectively. We choose this area because of a high number of in-focus beads presented. According to figure 3-5, it is clearly shown that (a) gives the best reconstruction result. The reconstruction in figure 3-5(b) has some white noise in the

background and the point source is not really in focus. The experimental hologram in figure (b) is obtained by propagating the experimental hologram 131207_19 by a distance $\Delta z = 40 \mu m$ then correlated with the object hologram. In-focus points are not reconstructed clearly and white blurry noise can be found in the background. We can notice the shape and focus points in this reconstruction but the quality is not good enough. However, the reconstruction as shown in figure 3-5(c) has a better quality as the reference pinhole hologram 131207_10 is propagated a smaller distance $\Delta z = 5 \mu m$. Therefore, we can conclude that the smaller propagated distance, the less error in the reconstruction. The worst case is the reconstruction in figure 3-5(d). Some in-focus points are totally blurry and there are a lot of aberrations in the background. It is understandable because the pinhole reconstructed previously has the biggest error.

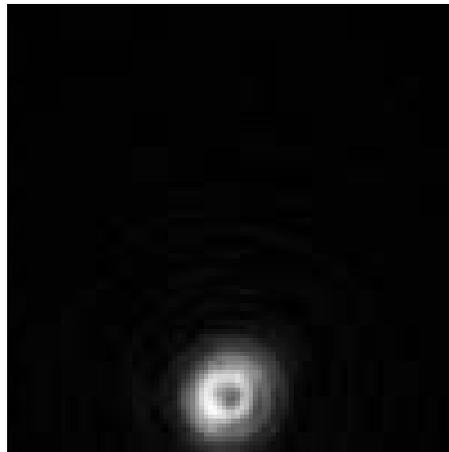
As discussed, reconstruction using only experimental pinhole hologram as described in approach A has shown to be the best way to reconstruct as they give the least aberration in the reconstruction of the fluorescent beads. However, the drawback of this approach is that we need to measure experimentally many pinhole holograms, one for a specific plane. In contrast, approach C contains too much error to be considered a reliable method in reconstructing. With approach B, the reconstruction is not as good as approach A, but it has more flexibility to create the reconstruction at any depth. If the propagation distance is not too large, approach B could give an acceptable reconstruction. This is exactly what we expected because when the actual experimental hologram is used for reconstruction as the reference, the aberration presented in the object space is cancelled out by the correlation with the experimental hologram recorded with the same aberration.



(a) Approach A - Using experimental pinhole hologram only



(b) Approach B - Using propagated pinhole hologram



(c) Approach C - Using mathematical $\mathcal{H}(k_x, k_y; z_R)$ only

Figure 3-4. Amplitude plot: Reconstruction of the pinhole hologram 131207_11 at resolution 100x100 pixel centered at $(x, y) = (750, 750)$.

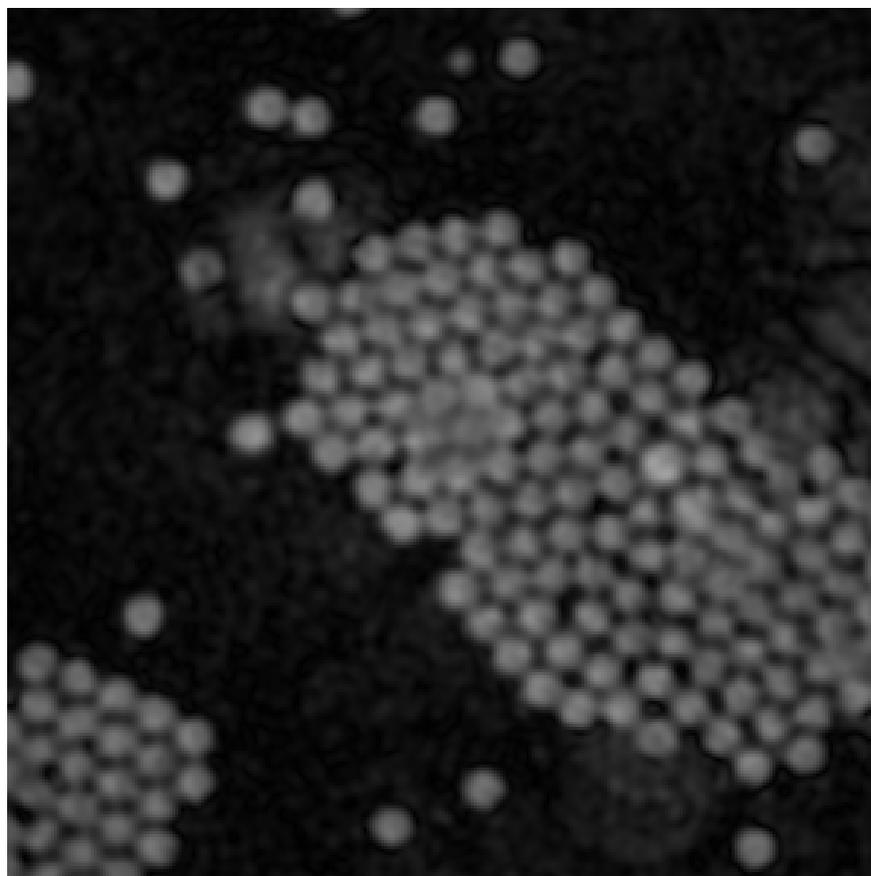


Figure 3-5(a). Amplitude plot: Reconstruction of the fluorescent beads hologram 131207_26 with the reference pinhole hologram 131207_11 using approach A at depth $z = 85 \mu m$ with resolution 300x300 pixel centered at $(x, y) = (1000, 500)$.

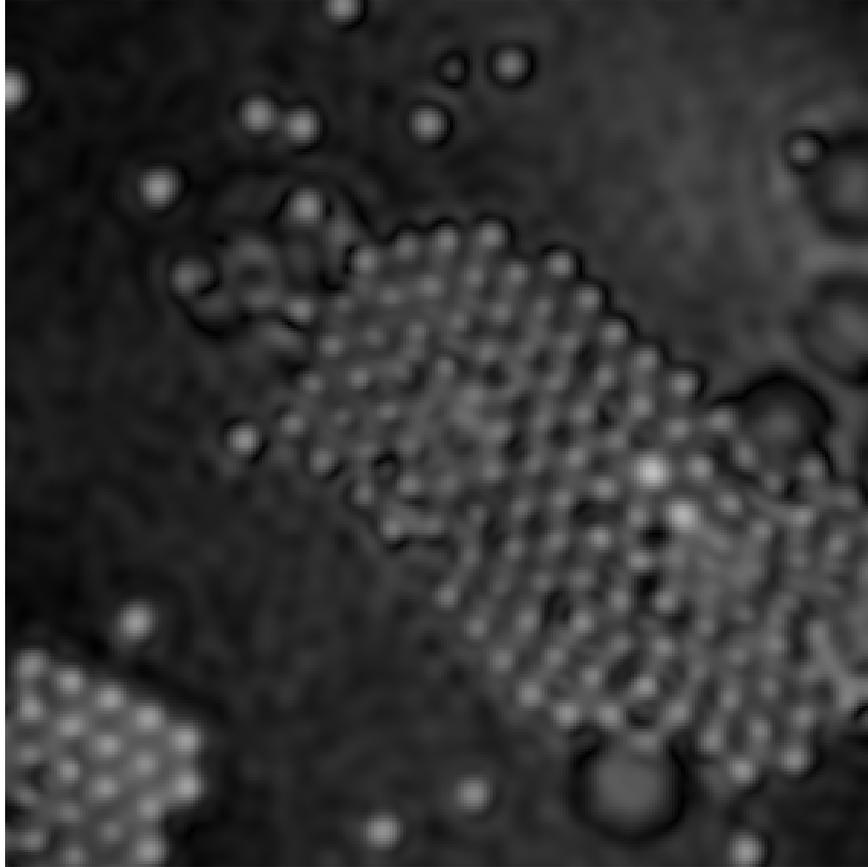


Figure 3-5(b). Amplitude plot: Reconstruction of the fluorescent beads hologram 131207_26 with the propagated reference pinhole hologram 131207_19 at depth $z = 85 \mu m$ using approach B with resolution 300x300 pixel centered at $(x, y) = (1000, 500)$.

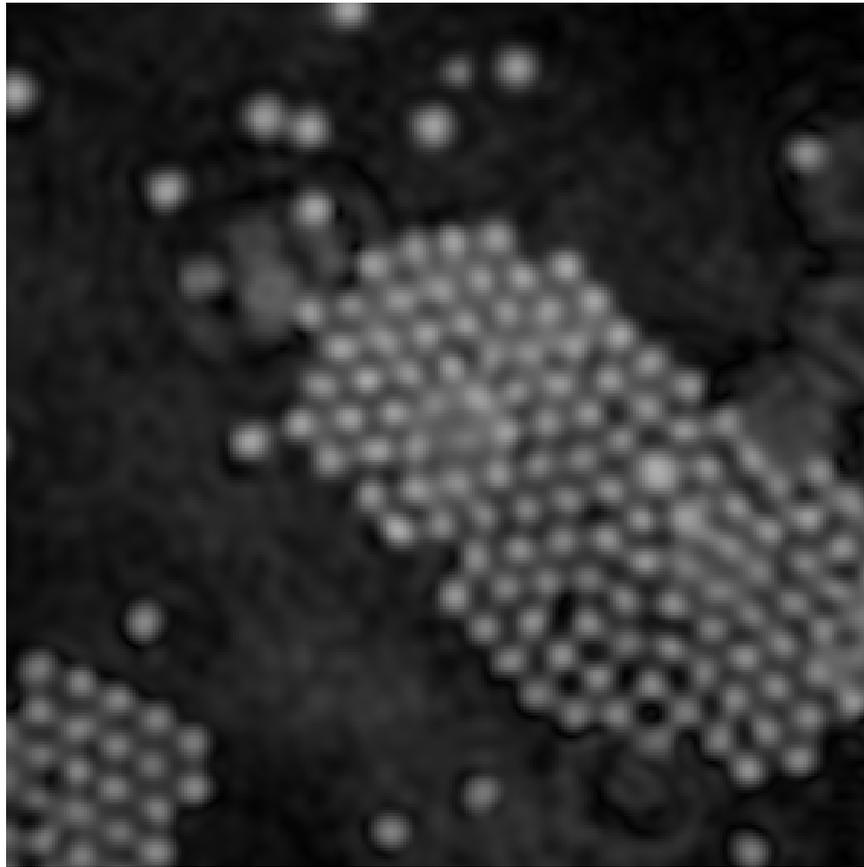


Figure 3-5(c). Amplitude plot: Reconstruction of the fluorescent beads hologram 131207_26 with the propagated reference pinhole hologram 131207_10 at depth $z = 85 \mu m$ using approach B with resolution 300x300 pixel centered at $(x, y) = (1000, 500)$.

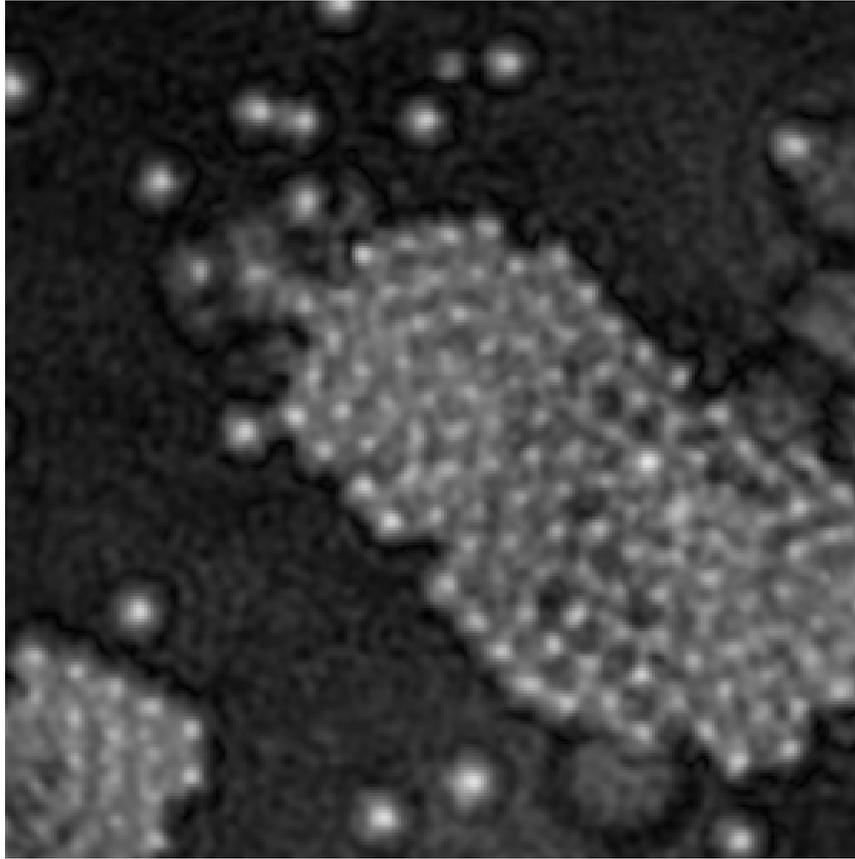


Figure 3-5(d). Amplitude plot: Reconstruction of the fluorescent beads hologram 131207_26 at depth $z = 85 \mu m$ using approach C with resolution 300x300 pixel centered at $(x, y) = (1000, 500)$.

CHAPTER 4 CONCLUSIONS

In this thesis, we have introduced the principle of holography and optical scanning holography application on three-dimensional microscopy. Several methods to reconstruct hologram of fluorescent beads are demonstrated. We also have showed that the reconstruction of the object hologram using the experimental pinhole hologram. We have presented in high resolution the reconstruction of the fluorescent beads. Reconstruction method using only experimental hologram gives us the best result; however, it has its limitation. On the other hand, the propagation method is more flexible in reconstruction at different depths but with more errors, depending how far we propagate a single experimental reference hologram. The choice of which method to be used is up to the requirement of the research.

In the future, the holographic data presented in this thesis has a lot of potential to be used in further research on OSH. The data is already extracted and analyzed to give easier access to work with. In fact, it has been used in optical sectioning presented in “Three-dimensional microscopy and sectional image reconstruction using optical scanning holography”^[5].

References

1. T.-C. Poon, *Optical Scanning Holography with MATLAB*, Springer, 2007.
2. T.-C. Poon and T. Kim, *Engineering Optics with MATLAB*, World Scientific, 2006.
3. T.-C. Poon, "Scanning holography and two-dimensional image processing by acousto-optic two-pupil synthesis," *Journal of the Optical Society of America A*, Vol.2, pp.521-527, 1985.
4. G. Indebetouw, "Scanning holographic microscopy with spatially incoherent sources: reconciling the holographic advantage with the sectioning advantage," *Journal of the Optical Society of America A*, Vol. 26, pp.252-258, 2009.
5. E. Y. Lam, X. Zhang, H. Vo, T.-C. Poon, and G. Indebetouw, "Three-dimensional microscopy and sectional image reconstruction using optical scanning holography," *Applied Optics*, Vol. 48, Issue 34, pp.H113-H119, 2009.
6. G. Indebetouw and W. Zhong, "Scanning holographic microscopy of three-dimensional fluorescent specimens," *Journal of the Optical Society of America A*, Vol. 23, 1699-1707, 2006.
7. P. Ferraro, S. D. Nicola and G. Coppola, "Controlling image reconstruction process in digital holography" in *Digital Holography and Three-Dimensional Display*, Springer, 2006.
8. J. W. Goodman, *Introduction to Fourier Optics*, Roberts & Company, 2004, 3rd Edition.
9. T.-C. Poon, K. Doh, B. Schilling, M. Wu, K. Shinoda, and Y. Suzuki, "Three-dimensional microscopy by optical scanning holography," *Optical Engineering*, Vol. 34, 1338, 1995.
10. A. W. Lohmann and W. T. Rhodes, "Two-pupil synthesis of optical transfer functions," *Applied Optics*, Vol. 17, pp.1141-1151, 1978.
11. G. Indebetouw, P. Klysubun, T. Kim, and T.-C. Poon, "Imaging properties of scanning holographic microscopy," *Journal of the Optical Society of America A*, Vol. 17, pp.380-390, 2000.
12. T.-C. Poon and A. Korpel, "Optical transfer function of an acousto-optic heterodyning image processor," *Optical Letter*, Vol. 4, pp.317-319, 1979.
13. B. D. Duncan and T.-C. Poon, "Gaussian Beam Analysis of Optical Scanning Holography," *Journal of the Optical Society of America A*, Vol 9, pp.229-236, 1992.

14. G. Indebetouw, W. Zhong, and D. Chamberlin-Long, "Point-spread function synthesis in scanning holographic microscopy," *Journal of the Optical Society of America A*, Vol. 23, pp.1708-1717, 2006.
15. G. Indebetouw, "Properties of a scanning holographic microscope: improved resolution, extended depth of focus and/or optical sectioning," *Journal of Modern Optics*, Vol. 49, pp.1479-1500, 2002.
16. B. Schilling, T.-C. Poon, G. Indebetouw, B. Storrie, K. Shinoda, and M. Wu, "Three-dimensional holographic fluorescence microscopy", *Optical Letter*, Vol. 22, pp.1506-1508, 1997.
17. G. W. Goodman, *Introduction to Fourier Optics*, McGraw-Hill, 1998
18. I. Yamaguchi and T. Zhang, "Phase-shifting digital holography," *Optical Letter*, Vol. 22, pp.1268-1270, 1997.
19. J. Rosen and G. Brooker, "Digital spatially incoherent Fresnel holography," *Optical Letter*, Vol. 22, pp.912-914, 2007

Appendix A

Instruction - How to use the MATLAB programs and Hologram data for reconstruction

I. (Optional) Extract data from signal files. (extract_data.m)

This step is optional because all signal files are already extracted into “.mat” files. However, this instruction may be helpful.

1. Run MATLAB file “*extract_data.m*”.
2. Select 1 if the parameter file exists or 0 to manually input parameters. The file “*Parameters.mat*” contains the parameters listed in Table 2.
3. Select the reference signal file ending with R. For example “*131207_11R.sig*”.
4. Select the modulated reference signal file ending with S. For example “*131207_11S.sig*”.
5. The program will take a while to extract the signal files to the output “.mat” file.

II. (Optional) Generate reference hologram (reference_holo.m)

This step is to generate a reference pinhole hologram for use in reconstruction.

1. Run MATLAB file “*reference_holo.m*”
2. Select 1 if the parameter file exists or 0 to manually input parameters. The file “*Parameters.mat*” contains the parameters listed in Table 2.
3. Select “.mat” pinhole hologram file (ending with 8 to 19 in “*Hologram mat*” folder).

Note: if you want to keep the experimental hologram untouched, select 0 for step 4, 5 and 6.

4. When asked to “*Center it ?*”, if you want to center the reference hologram then select 1 and follow step a to c below. Otherwise, select 0.
 - a. Use “*Data Cursor*” tool in MATLAB to spot the center point in “*Phase of Experimental Pinhole Hologram*” graph.
 - b. Center it to $x=750$ and $y=750$.
For example, if the Data Cursor gives the location $x=747$ and $y=783$, then in the MATLAB command windows,

1. input 3 for number of columns rotate
 2. Input -33 for number of rows rotate
 - c. Repeat a and b if necessary.
5. When asked to *"Zero the outside"*, select 1 if you want to do so, the radius should be 150.
 6. When asked to *"Make the amplitude of hologram 1 within the specified radius"*, select 1 to make the amplitude 1 within the specified radius.
 7. Save the reference pinhole hologram. For my reference, I save it as *"11Ref.mat"*. This file is used as the reference pinhole hologram in III, IV and V.
 8. When asked to *"Do you want to save parameters to a file ?"*, select 1 to save if you don't already have *"Parameters.mat"*.

III. **Reconstruction using experimental hologram only** (reconstruct_option1.m)

Make sure you have a reference hologram created with instruction II above before continue.

1. Run the MATLAB file *"reconstruct_option1.m"*.
2. When asked *"Do you want to load parameters from data files ?"*, select 1 if the parameter file exists or 0 to manually input parameters. Load file *"Parameters.mat"* which contains the parameters listed in Table 2.
3. When asked for *"Loading Object Hologram"*, pick the object hologram. For example *"131207_26.mat"* for the hologram of the fluorescent beads.
4. When asked for *"Loading Reference Hologram"*, pick the reference hologram that is created by instruction II. For example *"11Ref.mat"* as mentioned above.
5. When asked *"Where do you want to save your figures ?"*, select a folder to save figures. Remember this location to find the reconstructed images.
6. When asked *"Do you want to reduce image size"*, input 1 if you want to zoom in to a smaller resolution and follow step a to c below. Otherwise, input 0 if you want to keep it in the original 1500x1500 resolution.

- a. Use “*Data Cursor*” tool to find the center (x_0, y_0) .
- b. Input the lower bound and upper bound of x. $(=x_0 - \text{resolution}/2)$.
- c. Input the lower bound and upper bound of y. $(=y_0 - \text{resolution}/2)$.

For example: to have a 300x300 resolution centered at (1000,500), input

- lower x=850
- upper x = 1150
- lower y = 350
- upper y = 650

7. When asked to “*Do you want to save parameters to a file ?*”, select 1 to save if you don’t already have “*Parameters.mat*”.

IV. **Reconstruction using propagated pinhole hologram** (reconstruct_option2.m)

Make sure you have a reference hologram created with instruction II above before continue.

1. Run the MATLAB file “*reconstruct_option2.m*”.
2. When asked “*Do you want to load parameters from data files ?*”, select 1 if the parameter file exists or 0 to manually input parameters. Load file “*Parameters.mat*” which contains the parameters listed in Table 2.
3. When asked for “*Loading Object Hologram*”, pick the object hologram. For example, “*131207_26.mat*” for the hologram of the fluorescent beads.
4. When asked for “*Loading Reference Hologram*”, pick the reference hologram that is created by instruction II. For example “*11Ref.mat*” as mentioned above.
5. Input “*Radius of Point Spread Function*”: 150
6. Input “*Length of the Field*”: 300
7. Input “*Initial Z*”, “*Final Z*”, and “*Increment of Z*”. These Z values indicate the range of depth z of the reconstruction.
8. Input “*Location of the reference pinhole hologram Z*”, look up for this value in table 1. For example, the depth of reference pinhole hologram 131207_16 is 60 μm .

8. When asked “Where do you want to save your figures ?”, select a folder to save figures. Remember this location to find the reconstructed images.
9. When asked “Do you want to reduce image size”, input 1 if you want to zoom in to a smaller resolution and follow step a to c below. Otherwise, input 0 if you want to keep it in the original 1500x1500 resolution.
 - a. Use “Data Cursor” tool to find the center (x_0, y_0) .
 - b. Input the lower bound and upper bound of x. $(=x_0 - \text{resolution}/2)$.
 - c. Input the lower bound and upper bound of y. $(=y_0 - \text{resolution}/2)$.

For example: to have a 300x300 resolution centered at (1000,500), input

 - lower x=850
 - upper x = 1150
 - lower y = 350
 - upper y = 650
10. When asked to “Do you want to save parameters to a file ?”, select 1 to save if you don’t already have “Parameters.mat”.

V. Reconstruction using mathematical $\mathcal{H}(k_x, k_y, z_R)$ only (reconstruct_option3.m)

1. Run MATLAB file “reconstruct_option3.m”.
2. When asked “Do you want to load parameters from data files ?”, select 1 if the parameter file exists or 0 to manually input parameters. Load file “Parameters.mat” which contains the parameters listed in Table 2.
3. When asked for “Loading Object Hologram”, pick the object hologram. For example, “131207_26.mat” for the hologram of the fluorescent beads.
4. Input “Radius of Point Spread Function”: 150
5. Input “Length of the Field”: 300
6. Input “Initial Z”, “Final Z”, and “Increment of Z”. These Z values indicate the range of depth z of the reconstruction.
7. When asked “Where do you want to save your figures ?”, select a folder to save figures. Remember this location to find the reconstructed images.

8. When asked *“Do you want to reduce image size”*, input 1 if you want to zoom in to a smaller resolution and follow step a to c below. Otherwise, input 0 if you want to keep it in the original 1500x1500 resolution.

a. Use *“Data Cursor”* tool to find the center (x_0, y_0) .

b. Input the lower bound and upper bound of x. $(=x_0 - \text{resolution}/2)$.

c. Input the lower bound and upper bound of y. $(=y_0 - \text{resolution}/2)$.

For example: to have a 300x300 resolution centered at (1000,500), input

- lower x=850
- upper x = 1150
- lower y = 350
- upper y = 650

9. When asked to *“Do you want to save parameters to a file ?”*, select 1 to save if you don't already have *“Parameters.mat”*.

Appendix B

```
function extract_data()

    close all;
    clear all;

%=====Reading the original
matrix=====
    if input('Do you want to load parameters from data files ? (yes = 1, no =
0) ') == 1
        [Reference_File, File_Path] = uigetfile('*.mat', 'Select the variable
file');
        load
        (Reference_File, 'N_Lines', 'N_Columns', 'N_Valid', 'N_Invalid', 'N_Initial_Invali
d', 'Sampling_Rate', 'Filter_Bandwidth', 'Modulation_Frequency');
        display(['Number of Data Lines = ', num2str(N_Lines)]);
        display(['Number of Valid Sampling Points = ', num2str(N_Valid)]);
        display(['Number of Invalid Sampling Points = ', num2str(N_Invalid)]);
        display(['Number of Initial Invalid Sampling Points =
', num2str(N_Initial_Invalid)]);

        display(['Sampling Rate Fs = ', num2str(Sampling_Rate)]);

        display(['Band Pass Filter Bandwidth DeltaF =
', num2str(Filter_Bandwidth)]);
        display(['Modulation Frequency Fm =
', num2str(Modulation_Frequency)]);
        ref_change = input('Do you want to change ? (yes=1, no=0) ');
        end

    while(ref_change==1)
        N_Lines = input('Number of Data Lines = ');
        N_Valid = input('Number of Valid Sampling Points = ');
        N_Invalid = input('Number of Invalid Sampling Points = ');
        N_Initial_Invalid = input('Number of Initial Invalid Sampling Points
= ');

        Sampling_Rate = input('Sampling Rate Fs = ');

        Filter_Bandwidth = input('Band Pass Filter Bandwidth DeltaF = ');
        Modulation_Frequency = input('Modulation Frequency Fm = ');
        ref_change = input('Do you want to change ? (yes=1, no=0) ');
        end

%=====
    display('Loading Signal File (S)');
    [RefHoloFile, File_Path] = uigetfile('*.sig', 'Select the Signal File');
    display('Loading Modulated Signal File (R)');
    [ModHoloFile, File_Path] = uigetfile('*.sig', 'Select the Modulated
Signal File');
```

```

% =====
% ----- Band Pass Filter Parameters -----
% =====

N_Columns = Filter_Bandwidth*N_Valid/Sampling_Rate;

Filter_Center =
(Sampling_Rate/2+Modulation_Frequency)*N_Valid/Sampling_Rate;
X_Center = N_Columns/2;
Y_Center = N_Lines/2;

Band_Begin = Filter_Center-X_Center+1;
Band_End = Filter_Center+X_Center;

% =====
% ----- File Information -----
% =====

Hologram = zeros(N_Lines,N_Columns);

disp(['Reading Signal Files....']);

Filename1 = [File_Path,'\ ',RefHoloFile];
Filename2 = [File_Path,'\ ',ModHoloFile];

Reference_Signal(7) = '-';
Modulated_Reference_Signal(7) = '-';

% =====
% Skip 512 bytes of file header and 2*Initial_Invalid bytes of invalid data
% =====

Fid1 = fopen(Filename1,'r');
Status = fseek(Fid1,512+2*N_Initial_Invalid,-1);

Fid2 = fopen(Filename2,'r');
Status = fseek(Fid2,512+2*N_Initial_Invalid,-1);

% =====
% ----- Creating Holograms -----
% =====

for k = 1:N_Lines

% =====
% Read in 2*Valid bytes of data and skip 2*Invalid bytes of invalid data --
% =====

```

```

    CCDData(1:N_Valid) =
fread(Fid1,N_Valid,[num2str(N_Valid),'*int16'],2*N_Invalid);
    CCDData = conj(CCDData)*(-1/6550);

    Modulated_Data(1:N_Valid) =
fread(Fid2,N_Valid,[num2str(N_Valid),'*int16'],2*N_Invalid);

    Modulated_Data = conj(Modulated_Data)*(-1/6550);

% =====
% ----- Fourier Transform -----
% =====

    CCDData = fftshift(fft(CCDData));
    Modulated_Data = fftshift(fft(Modulated_Data));

% =====
% ----- Band Pass Filter -----
% =====

    CCDData = CCDData(Band_Begin:Band_End);
    Modulated_Data = Modulated_Data(Band_Begin:Band_End);

% ----- Inverse Fourier Transform -----

    CCDData = ifft(ifftshift(CCDData));
    Modulated_Data = ifft(ifftshift(Modulated_Data));

% ----- Subtraction of Phase -----
    CCDData = CCDData.*conj(-1*Modulated_Data);

% ----- Arranging data in 2D format -----

    Hologram(k,1:N_Columns) = CCDData;
end

fclose(Fid1);
fclose(Fid2);

realHolo = real(Hologram);
imgHolo = imag(Hologram);

File_Name = RefHoloFile(1:strfind(RefHoloFile, 'S')-1);

disp('Writing...');

```

```
%Save using binary compression.  
%Output_Filebinary = [File_Path, '\', File_Name, '-binary.mat'];  
%save(Output_Filebinary, 'realHolo', 'imgHolo');
```

```
%Save using MATLAB v7.3 compression  
Output_Filev73 = [File_Path, '\', File_Name, '.mat'];  
save(Output_Filev73, 'realHolo', 'imgHolo', '-v7.3');
```

```
disp('Writing completed');
```

```
end
```

```

function reference_holo()

%=====Loading parameters=====
    ref_change = 1;
    if input('Do you want to load default parameters from a file ? (yes = 1,
no = 0) ') == 1
        [Reference_File, File_Path] = uigetfile('*.mat', 'Select the
parameters file');
        load
        (Reference_File, 'N_Lines', 'N_Columns', 'N_Valid', 'N_Invalid', 'N_Initial_Invalid', 'Sampling_Rate', 'Filter_Bandwidth', 'Modulation_Frequency');
        display(['Number of Data Lines = ', num2str(N_Lines)]);
        display(['Number of Valid Sampling Points = ', num2str(N_Valid)]);
        display(['Number of Invalid Sampling Points = ', num2str(N_Invalid)]);
        display(['Number of Initial Invalid Sampling Points =
', num2str(N_Initial_Invalid)]);

        display(['Sampling Rate Fs = ', num2str(Sampling_Rate)]);

        display(['Band Pass Filter Bandwidth DeltaF =
', num2str(Filter_Bandwidth)]);
        display(['Modulation Frequency Fm =
', num2str(Modulation_Frequency)]);
        ref_change = input('Do you want to change ? (yes=1, no=0) ');
    end

    while(ref_change==1)
        N_Lines = input('Number of Data Lines = ');
        N_Valid = input('Number of Valid Sampling Points = ');
        N_Invalid = input('Number of Invalid Sampling Points = ');
        N_Initial_Invalid = input('Number of Initial Invalid Sampling Points
= ');

        Sampling_Rate = input('Sampling Rate Fs = ');

        Filter_Bandwidth = input('Band Pass Filter Bandwidth DeltaF = ');
        Modulation_Frequency = input('Modulation Frequency Fm = ');
        ref_change = input('Do you want to change ? (yes=1, no=0) ');
    end

%=====Loading pinhole hologram=====
    display('Loading pinhole hologram to make it as reference hologram');
    [InputFile, File_Path] = uigetfile('*.mat', 'Select the pinhole hologram
');
    RefHoloFile = [File_Path, InputFile];
    load(RefHoloFile, 'realHolo', 'imgHolo');
    display(['Loaded pinhole hologram: ', InputFile]);
    Hologram = realHolo + i*imgHolo;

    %Bandpass filter
    N_Columns = Filter_Bandwidth*N_Valid/Sampling_Rate;
    Filter_Center =
(Sampling_Rate/2+Modulation_Frequency)*N_Valid/Sampling_Rate;

```

```

X_Center = N_Columns/2;
Y_Center = N_Lines/2;
Band_Begin = Filter_Center-X_Center+1;
Band_End = Filter_Center+X_Center;

%construct the reference hologram
figure;
imagesc(angle(Hologram));
colormap('Gray');
title('Phase of Experimental Pinhole Hologram');
set(gca, 'XGrid', 'on', 'YGrid', 'on', 'GridLineStyle', '-');
axis square

%=====Center the hologram=====
j = 1;
while j
    if input('Center it (Yes = 1; No = 0) ? ') == 0
        j = 0;
    else
        X_Shift = input('Number of columns to rotate left (Negative) or
right (Positive) = ');
        Y_Shift = input('Number of rows to rotate up (Negative) or down
(Positive) = ');
        Hologram = circshift(Hologram, [Y_Shift X_Shift]);

        close(gcf);
        figure;
        imagesc(angle(Hologram));
        colormap('Gray');
        title('Phase of Experimental Pinhole Hologram');
        set(gca, 'XGrid', 'on', 'YGrid', 'on', 'GridLineStyle', '-');
        axis square
    end
end

%zero reference
if input('Zero the outside with a selected radius (Yes = 1; No = 0) ? ')
== 1
    Radius = input('Radius = ');
    Radius2 = Radius^2;

    if input('Make the amplitude of hologram 1 within the specified
radius (Yes = 1; No = 0) ? ') == 1
        Hologram = exp(i*angle(Hologram));
    end

    Box_X_Begin = X_Center-Radius-10;
    Box_X_End = X_Center+Radius+10;

    Box_Y_Begin = Y_Center-Radius-10;
    Box_Y_End = Y_Center+Radius+10;

```

```

for j = Box_Y_Begin:Box_Y_End
    for k = Box_X_Begin:Box_X_End
        Check = (j-Y_Center)^2 + (k-X_Center)^2;
        if Check > Radius2
            Hologram(j,k) = 0;
        end
    end
end

Hologram(1:Box_Y_Begin,1:N_Columns) = 0;
Hologram(Box_Y_End:N_Lines,1:N_Columns) = 0;
Hologram(Box_Y_Begin:Box_Y_End,1:Box_X_Begin) = 0;
Hologram(Box_Y_Begin:Box_Y_End,Box_X_End:N_Columns) = 0;

disp('Zeroing Noise: COMPLETED!');

close(gcf);
figure;
imagesc(angle(Hologram));
colormap('Gray');
title(['Phase of Experimental Pinhole Hologram ']);
set(gca,'XGrid','on','YGrid','on','GridLineStyle','-');
axis square

1 elseif input('Zero the outside with a threshold (Yes = 1; No = 0) ? ') ==

Threshold = input('Threshold = ');

for j = 1:N_Lines
    for k = 1:N_Columns
        if abs(Hologram(j,k)) < Threshold
            Hologram(j,k) = 0;
        end
    end
end

disp('Zeroing Noise: COMPLETED!');

close(gcf);
figure;
imagesc(angle(Hologram));
colormap('Gray');
title(['Phase of Experimental Pinhole Hologram : ']);
set(gca,'XGrid','on','YGrid','on','GridLineStyle','-');
axis square

end

%=====saving reference pinhole hologram=====

display('Saving Reference Pinhole Hologram');
uisave('Reference Hologram');

```

```
%saving reference variables
if input('Do you want to save parameters to a file ? (yes=1, no=0) ')
    %Reference_File = input('Enter parameter variable name: ','s');
    %create a file path.
    %Reference_Path = uigetdir(File_Path);

uisave({'N_Lines','N_Columns','N_Valid','N_Invalid','N_Initial_Invalid','Sampling_Rate','Filter_Bandwidth','Modulation_Frequency'},'Parameters');
end

display('COMPLETED');

end
```

```

function reconstruct_option1()

    close all;
    clear all;

%=====Loading parameters=====

    if input('Do you want to load parameters from data files ? (yes = 1, no =
0) ') == 1
        [Reference_File, File_Path] = uigetfile('*.mat', 'Select the variable
file');
        load
        (Reference_File, 'N_Lines', 'N_Columns', 'N_Valid', 'N_Invalid', 'N_Initial_Invali
d', 'Sampling_Rate', 'Filter_Bandwidth', 'Modulation_Frequency');
        display(['Number of Data Lines = ', num2str(N_Lines)]);
        display(['Number of Valid Sampling Points = ', num2str(N_Valid)]);
        display(['Number of Invalid Sampling Points = ', num2str(N_Invalid)]);
        display(['Number of Initial Invalid Sampling Points =
', num2str(N_Initial_Invalid)]);

        display(['Sampling Rate Fs = ', num2str(Sampling_Rate)]);

        display(['Band Pass Filter Bandwidth DeltaF =
', num2str(Filter_Bandwidth)]);
        display(['Modulation Frequency Fm =
', num2str(Modulation_Frequency)]);
        ref_change = input('Do you want to change ? (yes=1, no=0) ');
    end

    while(ref_change==1)
        N_Lines = input('Number of Data Lines = ');
        N_Valid = input('Number of Valid Sampling Points = ');
        N_Invalid = input('Number of Invalid Sampling Points = ');
        N_Initial_Invalid = input('Number of Initial Invalid Sampling Points
= ');

        Sampling_Rate = input('Sampling Rate Fs = ');

        Filter_Bandwidth = input('Band Pass Filter Bandwidth DeltaF = ');
        Modulation_Frequency = input('Modulation Frequency Fm = ');
        ref_change = input('Do you want to change ? (yes=1, no=0) ');
    end

%=====Loading Holograms=====

    display('Loading Object Hologram');
    [ObjHoloFile, File_Path] = uigetfile('*.mat', 'Select the Object Hologram
');
    ObjHoloPathFile = [File_Path, ObjHoloFile];
    load(ObjHoloPathFile, 'realHolo', 'imgHolo');
    ObjHologram = realHolo + 1i*imgHolo;
    display(['Loaded ', ObjHoloFile, ' as the Object Hologram']);

```

```

display('Loading Reference Hologram');
[RefHoloFile, File_Path] = uigetfile('*.mat', 'Select the Reference
Hologram ');
RefHoloPathFile = [File_Path, RefHoloFile];
load(RefHoloPathFile, 'Hologram');
RefHologram = Hologram;
display(['Loaded ', RefHoloFile, ' as the Reference Hologram']);

%====Graph the loaded hologram=====

%getting figure path.
display('Where do you want to save your figures ?');
Figure_Path = uigetdir(File_Path, 'Select directory to save figures');

figure(1);
imagesc(abs(RefHologram));
colormap('Gray');
title(['Amplitude plot of pinhole reference hologram']);
axis square
saveas(gcf, [Figure_Path, '\', 'Ref_Amp.tiff'], 'tiffn');

figure(2);
imagesc(angle(RefHologram));
colormap('Gray');
title(['Phase plot of pinhole reference hologram']);
axis square
saveas(gcf, [Figure_Path, '\', 'Ref_Phase.tiff'], 'tiffn');

%Reference
figure(3);
imagesc(abs(ObjHologram));
colormap('Gray');
title(['Amplitude plot of the object hologram']);
axis square
saveas(gcf, [Figure_Path, '\', 'Obj_Amp.tiff'], 'tiffn');

figure(4);
imagesc(angle(ObjHologram));
colormap('Gray');
title(['Phase plot of the object hologram']);
axis square
saveas(gcf, [Figure_Path, '\', 'Obj_Phase.tiff'], 'tiffn');

if input('close figures (Yes = 1; No = 0) ? ') == 1
    close(gcf);
    close(gcf);
    close(gcf);
    close(gcf);
end

```

```

%=====Calculating the convolution of obj*ref=====
    display('Calculating reconstruction...');
    RefHologram = fftshift(fft2(RefHologram));
    ObjHologram = fftshift(fft2(ObjHologram));
    Reconstruction =
fftshift(ifft2(ifftshift(ObjHologram.*conj(RefHologram))));

    max1=max(max(abs(Reconstruction)));
    max2=max(max(angle(Reconstruction)));

    figure;
    imagesc(abs(Reconstruction));
    colormap('Gray');
    title(['Amplitude Plot - Reconstruction of ',ObjHoloFile,' Object with
',RefHoloFile,' Reference']);
    axis square

    if input('Do you want to reduce image size (Yes = 1; No = 0) ? ') == 1
        Xl = input('Lower value of x?');
        Xu = input('Upper value of x?');
        Yl = input('Lower value of y?');
        Yu = input('Upper value of y?');

        Temp = Reconstruction(Yl:1:Yu,Xl:1:Xu);
        clear Reconstruction;
        Reconstruction = Temp;
        clear Temp;

    end
    close(gcf);

    %Saving figures

imwrite(abs(Reconstruction)./max1,[Figure_Path,'\Reconstruction_Amp',ObjHoloF
ile,'_with_',RefHoloFile,'_op1.bmp'],'bmp');
    figure;
    imagesc(abs(Reconstruction));
    colormap('Gray');
    title(['Reconstruction Amplitude of ',ObjHoloFile,' with ',RefHoloFile
]);
    axis square
    saveas(gcf,[Figure_Path,'\Reconstruction_Amplitude ',ObjHoloFile,' with
',RefHoloFile,'_op1.tiff'],'tiffn');

imwrite(angle(Reconstruction)./max2,[Figure_Path,'\Reconstruction_Phase',ObjH
oloFile,'_with_',RefHoloFile,'_op1.bmp'],'bmp');
    figure;
    imagesc(angle(Reconstruction));

```

```

colormap('Gray');
title(['Reconstruction Phase of ',ObjHoloFile,' with ',RefHoloFile]);
axis square
saveas(gcf,[Figure_Path,'\Reconstruction_Phase ',ObjHoloFile,' with
',RefHoloFile,'_op1.tiff'],'tiffn');

if input('close figures (Yes = 1; No = 0) ? ') == 1
    close(gcf);
    close(gcf);
end

%Saving parameters
if input('Do you want to save parameters to a file ? (yes=1, no=0) ')

uisave({'N_Lines','N_Columns','N_Valid','N_Invalid','N_Initial_Invalid','Samp
ling_Rate','Filter_Bandwidth','Modulation_Frequency'},'Parameters');
end
display('COMPLETED!!!');

end

```

```

function reconstruct_option2()

    close all;
    clear all;

%=====Loading parameters=====

    if input('Do you want to load parameters from data files ? (yes = 1, no =
0) ') == 1
        [Reference_File, File_Path] = uigetfile('*.mat', 'Select the variable
file');
        load
        (Reference_File, 'N_Lines', 'N_Columns', 'N_Valid', 'N_Invalid', 'N_Initial_Invali
d', 'Sampling_Rate', 'Filter_Bandwidth', 'Modulation_Frequency');
        display(['Number of Data Lines = ', num2str(N_Lines)]);
        display(['Number of Valid Sampling Points = ', num2str(N_Valid)]);
        display(['Number of Invalid Sampling Points = ', num2str(N_Invalid)]);
        display(['Number of Initial Invalid Sampling Points =
', num2str(N_Initial_Invalid)]);

        display(['Sampling Rate Fs = ', num2str(Sampling_Rate)]);

        display(['Band Pass Filter Bandwidth DeltaF =
', num2str(Filter_Bandwidth)]);
        display(['Modulation Frequency Fm =
', num2str(Modulation_Frequency)]);
        ref_change = input('Do you want to change ? (yes=1, no=0) ');
    end

    while(ref_change==1)
        N_Lines = input('Number of Data Lines = ');
        N_Valid = input('Number of Valid Sampling Points = ');
        N_Invalid = input('Number of Invalid Sampling Points = ');
        N_Initial_Invalid = input('Number of Initial Invalid Sampling Points
= ');

        Sampling_Rate = input('Sampling Rate Fs = ');

        Filter_Bandwidth = input('Band Pass Filter Bandwidth DeltaF = ');
        Modulation_Frequency = input('Modulation Frequency Fm = ');
        ref_change = input('Do you want to change ? (yes=1, no=0) ');
    end

%=====Loading object and reference holograms=====

    display('Loading Object Hologram');
    [ObjHoloFile, File_Path] = uigetfile('*.mat', 'Select the Object Hologram
');
    ObjHoloPathFile = [File_Path, ObjHoloFile];
    load(ObjHoloPathFile, 'realHolo', 'imgHolo');
    ObjHologram = realHolo + 1i*imgHolo;
    display(['Loaded ', ObjHoloFile, ' as the Object Hologram']);

```

```

display('Loading Reference Hologram');
[RefHoloFile, File_Path] = uigetfile('*.mat', 'Select the Reference
Hologram ');
RefHoloPathFile = [File_Path, RefHoloFile];
load(RefHoloPathFile, 'Hologram');
display(['Loaded ', RefHoloFile, ' as the Reference Hologram']);

Hologram = fftshift(fft2(Hologram));
ObjHologram = fftshift(fft2(ObjHologram));
Lambda = 0.532;      %wavelength constant

%Bandpass filter
N_Columns = Filter_Bandwidth*N_Valid/Sampling_Rate;
Filter_Center =
(Sampling_Rate/2+Modulation_Frequency)*N_Valid/Sampling_Rate;
X_Center = N_Columns/2;
Y_Center = N_Lines/2;
Band_Begin = Filter_Center-X_Center+1;
Band_End = Filter_Center+X_Center;

%Radius = 340;
Radius = input('Radius of Point Spread Function (in Pixel Number) = ');
Radius2 = Radius^2;

Field_Length = input('Length of the Field (in Micron) = ');

Z_Initial = input('Initial Z (in Micron) = ');
Z_Final = input('Final Z (in Micron) = ');
Z_Increment = input('Increment of Z (DeltaZ, in Micron) = ');

Z0 = input('Location of the reference pinhole hologram Z (in Micron) =
');
Z = Z_Initial-Z0;

display('Where do you want to save your figures?');
Figure_Path = uigetdir(File_Path, 'Select directory to save figures');

reduce_img = input('Do you want to reduce image size (Yes = 1; No = 0) ?
');
if reduce_img == 1
    Xl = input('Lower value of x?');
    Xu = input('Upper value of x?');
    Yl = input('Lower value of y?');
    Yu = input('Upper value of y?');
end

%=====Calculating reconstruction=====
C = 0;

while Z <= Z_Final-Z0

    C = C + 1;

```

```

display(['Processing z = ', num2str(Z+Z0), ' ... '])

P = zeros(N_Lines,N_Columns);

for j = 1:N_Lines
    for k = 1:N_Columns
        r2 = (j-Y_Center)^2+(k-X_Center)^2;
        if r2 <= Radius2
            %P(j,k) = exp(-1i*pi*Z*Lambda*r2/Field_Length^2);
%equation 2-9a
            P(j,k) = exp(-i*2*pi*Z/Lambda*(1-sqrt(1-
Lambda^2*r2/Field_Length^2))); %equation 2-9b

        end
    end
end

P = Hologram.*P;

Reconstruction = fftshift(iff2(iff2shift(ObjHologram.*conj(P))));

max1=max(max(abs(Reconstruction)));
max2=max(max(angle(Reconstruction)));

if (reduce_img == 1)
    Temp = Reconstruction(Yl:Yu,Xl:Xu);
    clear Reconstruction;
    Reconstruction = Temp;
    clear Temp;

end

%Saving figures
close(gcf);

imwrite(abs(Reconstruction)./max1,[Figure_Path,'\Reconstruction_Amp_',ObjHolo
File,'_',RefHoloFile,'_Z=',num2str(Z+Z0),'_op2.bmp'],'bmp');
figure;
imagesc(abs(Reconstruction));
colormap('Gray');
title(['Reconstruction Amplitude of ',ObjHoloFile,' using propagated
',RefHoloFile,' at Z = ',num2str(Z+Z0)]);
axis square

saveas(gcf,[Figure_Path,'\Reconstruction_Amp_',ObjHoloFile,'_',RefHoloFile,'_
Z=',num2str(Z+Z0),'_op2.tiff'],'tiffn');
close(gcf);

imwrite(angle(Reconstruction)./max2,[Figure_Path,'\Reconstruction_Phase_',Obj
HoloFile,'_',RefHoloFile,'_Z=',num2str(Z+Z0),'_op2.bmp'],'bmp');
figure;
imagesc(angle(Reconstruction));
colormap('Gray');

```

```

        title(['Reconstruction Phase of ',ObjHoloFile,' using propagated
',RefHoloFile,' at Z = ',num2str(Z+Z0)]);
        axis square

saveas(gcf,[Figure_Path,'\Reconstruction_Phase_',ObjHoloFile,'_',RefHoloFile,
'_Z=',num2str(Z+Z0),'_op2.tiff'],'tiffn');
        close(gcf);

        Z = Z+Z_Increment;
end

%Saving parameters
if input('Do you want to save parameters to a file ? (yes=1, no=0) ')

uisave({'N_Lines','N_Columns','N_Valid','N_Invalid','N_Initial_Invalid','Samp
ling_Rate','Filter_Bandwidth','Modulation_Frequency'],'Parameters');
end
display('COMPLETED!!!');

end

```

```

function reconstruct_option3()

%=====Loading parameters=====
    ref_change = 1;
    if input('Do you want to load parameters from data files ? (yes = 1, no =
0) ') == 1
        [Reference_File, File_Path] = uigetfile('*.mat', 'Select the variable
file');
        load
(Reference_File, 'N_Lines', 'N_Columns', 'N_Valid', 'N_Invalid', 'N_Initial_Invali
d', 'Sampling_Rate', 'Filter_Bandwidth', 'Modulation_Frequency');
        display(['Number of Data Lines = ', num2str(N_Lines)]);
        display(['Number of Valid Sampling Points = ', num2str(N_Valid)]);
        display(['Number of Invalid Sampling Points = ', num2str(N_Invalid)]);
        display(['Number of Initial Invalid Sampling Points =
', num2str(N_Initial_Invalid)]);

        display(['Sampling Rate Fs = ', num2str(Sampling_Rate)]);

        display(['Band Pass Filter Bandwidth DeltaF =
', num2str(Filter_Bandwidth)]);
        display(['Modulation Frequency Fm =
', num2str(Modulation_Frequency)]);
        ref_change = input('Do you want to change ? (yes=1, no=0) ');
    end

    while(ref_change==1)
        N_Lines = input('Number of Data Lines = ');
        N_Valid = input('Number of Valid Sampling Points = ');
        N_Invalid = input('Number of Invalid Sampling Points = ');
        N_Initial_Invalid = input('Number of Initial Invalid Sampling Points
= ');

        Sampling_Rate = input('Sampling Rate Fs = ');

        Filter_Bandwidth = input('Band Pass Filter Bandwidth DeltaF = ');
        Modulation_Frequency = input('Modulation Frequency Fm = ');
        ref_change = input('Do you want to change ? (yes=1, no=0) ');
    end

%=====Loading object holograms=====
    display('Loading Object Hologram');
    [InputFile, File_Path] = uigetfile('*.mat', 'Select the Object Hologram
');
    HoloFile = [File_Path, InputFile];
    load(HoloFile, 'realHolo', 'imgHolo');
    Hologram = realHolo + 1i*imgHolo;
    display(['Loaded ', HoloFile, ' as the Object Hologram']);

%Bandpass filter
N_Columns = Filter_Bandwidth*N_Valid/Sampling_Rate;
X_Center = N_Columns/2;

```

```

Y_Center = N_Lines/2;

%Radius = 150;
Radius = input('Radius of Point Spread Function (in Pixel Number) = ');
Radius2 = Radius^2;

Lambda = 0.532;

%Field_Length = 300 (
Field_Length = input('Length of the Field (in Micron) = ');

Z_Init = input('Initial depth (Z, in Micron) = ');
Z_Final = input('Final depth (Z, in Micron) = ');
Z_Inc = input('Increment step = ');

display('Where do you want to save your figures ?');
Figure_Path = uigetdir(File_Path, 'Select directory to save figures');

reduce_img = input('Do you want to reduce imagesize (Yes = 1; No = 0) ?
');
if (reduce_img == 1)
    Xl = input('Lower value of x?');
    Xu = input('Upper value of x?');
    Yl = input('Lower value of y?');
    Yu = input('Upper value of y?');
end

%=====Calculating reconstruction=====
Hologram = fftshift(fft2(Hologram));
Z0 = 85;
Z = Z_Init-Z0;
while(Z<=Z_Final-Z0)

    P = zeros(N_Lines,N_Columns);

    display(['Processing z = ',num2str(Z+Z0),' ... '])

    for j = 1:N_Lines
        for k = 1:N_Columns
            r2 = (j-Y_Center)^2+(k-X_Center)^2;
            if r2 <= Radius2
                %P(j,k) = exp(-1i*pi*Z0*Lambda*r2/Field_Length^2);
%equation 2-9a
                P(j,k) = exp(-i*2*pi*(Z0-Z)/Lambda*(1-sqrt(1-
Lambda^2*r2/Field_Length^2))); %equation 2-9b
            end
        end
    end
end
end

```

```

%P = fftshift(fft2(P));

Reconstruction = ifft2(ifftshift(Hologram.*(P)));

close(gcf);

max1=max(max(abs(Reconstruction)));
max2=max(max(angle(Reconstruction)));

if (reduce_img == 1)
    Temp = Reconstruction(Yl:Yu,Xl:Xu);
    clear Reconstruction;
    Reconstruction = Temp;
    clear Temp;

end

%Saving figures

imwrite(abs(Reconstruction)./max1,[Figure_Path,'\Reconstruction_Amp_',InputFile,
'_Z=',num2str(Z0+Z),'_op3.bmp'],'bmp');
figure;
imagesc(abs(Reconstruction));
colormap('Gray');
title(['Reconstruction Magnitude of ',InputFile,' Z = ',num2str(Z0+Z)]);
axis square

saveas(gcf,[Figure_Path,'\Reconstruction_Amp_',InputFile,'_Z=',num2str(Z0+Z),
'_op3.tiff'],'tiffn');
close(gcf);

imwrite(angle(Reconstruction)./max2,[Figure_Path,'\Reconstruction_Phase_',InputFile,
'_Z=',num2str(Z0+Z),'_op3.bmp'],'bmp');
figure;
imagesc(angle(Reconstruction));
colormap('Gray');
title(['Reconstruction Phase of ',InputFile,' Z = ',num2str(Z0+Z)]);
axis square

saveas(gcf,[Figure_Path,'\Reconstruction_Phase_',InputFile,'_Z=',num2str(Z0+Z),
'_op3.tiff'],'tiffn');
close(gcf);

clear P;
Z = Z+Z_Inc;
end

%Saving parameters
if input('Do you want to save parameters to a file ? (yes=1, no=0) ')

```

```
uisave({'N_Lines','N_Columns','N_Valid','N_Invalid','N_Initial_Invalid','Sampling_Rate','Filter_Bandwidth','Modulation_Frequency'},'Parameters');
    end
    display('COMPLETED!!!');
end
```